

X Window Programming from scratch

Learn X Window programming while building an application using Linux and C



J. Robert Brown



X Window Programming



J. Robert Brown



A Division of Macmillan USA

201 West 103rd Street, Indianapolis, Indiana 46290

X Window Programming from Scratch Copyright © 2000 by Que Corporation

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

International Standard Book Number: 0-7897-2372-7

Library of Congress Catalog Card Number: 00-100691

Printed in the United States of America

First Printing: August 2000

02 01 00 4 3 2 1

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Que Corporation cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The authors and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the CD or programs accompanying it.

Associate Publisher

Tracy Dunkelberger

Acquisitions Editor Katie Purdum

Development Editor Hugh Vandivier

Managing Editor Thomas F. Hayes

Senior Editor Susan Ross Moore

Copy Editor Cynthia Fields

Proofreaders

Harvey Stanbrough Megan Wade

Technical Editor Ed Petron

Team Coordinator Vicki Harding

Media Developer Michael Hunter

Interior Design Sandra Schroeder

Cover Designers Maureen McCarty Anne Jones

Copy Writer Eric Borgert

Production

Darin Crone Steve Geiselman

Contents at a Glance

	Introduction	xvi
Section One: Sta	arting Points	1
Part I: Abs	olute Zero	3
Chapter 1 UNIX for Developers		5
2	Programming Constructs	49
3	A Word on C	83
Part II: The	Pieces of X	111
Chapter 4	Windowing Concepts	113
5	Widget Sets	125
6	Components of an X Window Application	143
7	Xlib Graphic Primitives	179
Part III: Bac	ck to School	195
Chapter 8	Vector Versus Raster Graphics	197
9	Object Bounds Checking	203
10	Trigonometric and Geometric Functions	211
11	Graphic Transformations	225
12	Coordinate Systems	253
Section Two: Gr	aphics Editor Application	257
Part IV: Lay	ving Out the Parts	259
Chapter 13	Application Structure	261
14	Program Flow	299
15	Common Object Definition	305
16	Object Manipulation	319
17	Utilities and Tools	343
18	File Formats	355
19	Save and Restore	363

Part V:	Add	ling Objects to the Editor	377
Chapter 20 Latex Line Object		Latex Line Object	379
	21	Pencil Line Object	405
	22	Object Templates	411
23 Arc Object		Arc Object	421
	24	Vector Text Object	437
Part VI:	Add	ling a Print Driver	477
Chapter	25	Introduction to PostScript	479
	26	Color Versus Black and White	491
	27	Working with XImages and Colormaps	495
Part VII:	Wh	at's Next?	507
Chapter 28 Extending the Graphics Editor		509	
	29	Adding Context-Sensitive Help	513
Part VIII:	App	pendixes	521
	А	Command Shells and Scripting	523
	В	Application Layout Code Listing	539
	С	Additional Vector Font Sets and vector_chars.h	557
		Index	741

Table of Contents

	Introduction	XVİ	
Section One: Starting Points 1			
Part I: Abs	olute Zero	3	
Chapter 1	UNIX for Developers	5	
	The man Command Organization and Navigation Directories Permissions chmod The cd Command Object Files Source Files The vi Editor The vi Editor The make Utility The cc Command Makefile System Tools and Useful Commands grep, Pipes, Redirection, and more The find Command Next Steps		
Chapter 2	Programming Constructs	49	
	Decisions		

	Parameters	67
	Definition	68
	The return Statement	68
	Data	
	Data Types	
	Next Steps	81
Chapter 3	A Word on C	83
	Hello World	
	Comment Tokens	
	The Function main	
	Code Bodies	
	Variable Scope	
	Built-In Functions	
	Memory Management	
	Dynamic Memory Allocation	
	Memory Leaks	
	Definitions and Macros	
	Conclusion	
	Next Steps	
	*	

Part II: The Pieces of X

111

Chapter 4	Windowing Concepts	113
	Origins of the X Window System	113
	The Pieces of X	114
	Client/Server Model	114
	Window Hierarchy	121
	Next Steps	124
Chapter 5	Widget Sets	125
	The Power and Convenience of Using Widget Sets	125
	The Athena Widget Set	129
	The Core Widget	131
	Widgets That Manage Other Widgets	135
	The Motif Widgets	140
	Next Steps	142
Chapter 6	Components of an X Window Application	143
	Connecting to the X Server	144
	Employing Widget Resources Using Variable Argument Lists	144
	Creating the Application Interface	151
	Creating Buttons	159
	Creating Pixmap Icons	162
	Assigning Actions	169

	Managing Windows				
	Processing Events				
	Summary				
	Next Steps				
Chapter 7	Xlib Graphic Primitives	179			
	The Graphics Context				
	The GC Function				
	GCForeground and GCBackground				
	GCLineWidth				
	GCTile				
	Graphic Primitive Functions				
	XDrawPoint				
	XDrawLine				
	XDrawRectangle				
	XDrawArc				
	Filled Graphics				
	Next Steps				
Part III: Back To School 19					
Chapter 8	Vector Versus Raster Graphics	197			
	Vector Graphics				
	Raster Graphics				
	Next Steps	201			
Chapter 9	Object Bounds Checking	203			
	Point-Array-Based Object Bounds				
	Arc Object Bounds				
	Employing Object Bounds				
	Next Steps				
Chapter 10	Trigonometric and Geometric Functions	211			
	Calculating Point and Line Intersections				
	Calculating Slope				
	Calculating Point and Arc Intersections				
	Next Steps				
Chapter 11	Graphic Transformations	225			
	Moving				
	Moving a Line				
	Moving an Arc	231			
	Scaling	232			
	Scaling a Line	234			
	Scaling an Arc				

	Rotating	247
	Rotating a Line	248
	Rotating an Arc	251
	Next Steps	252
Chapter 12	Coordinate Systems	253
	Rectangular Coordinates	254
	Polar Coordinate System	254
	Next Steps	
Section Two: Gra	phics Editor Application	257
Part IV: Lay	ing Out the Parts	259
Chapter 13	Application Structure	261
	Project Structure	
	Laying Out the User Interface	
	Parsing the Command Line	
	The getopt Function	290
	The XtVaGetApplicationResources Function	292
	Setting Up a Canvas	
	Building the Project	297
	Next Steps	
Chapter 14	Program Flow	299
	Processing Events	
	X Event Hooks	302
	Widget Callbacks	
	Event Handlers	
	Widget Translation Tables	
	Next Steps	
Chapter 15	Common Object Definition	305
	Line Object Data Structure	305
	Text Object Data Structure	307
	Understanding Vector Fonts	
	The GXText Data Structure	
	Arc Object Data Structure	
	Common Object Data Structure	
	Next Steps	
Chapter 16	Object Manipulation	319
	Copying an Object	
	Deleting an Object	321

	Refreshing Objects	
	Parsing for an Object	
	Managing Object Handles	
	Managing the Status of an Object	
	Processing User Navigation of Objects	
	Next Steps	
Chapter 17	Utilities and Tools	343
	Common Object Creation	
	Linked List Management	
	Creating a Graphics Context	
	Graphics Context Tiling	
	Using the Cursor as State Indicator	
	Next Steps	
Chapter 18	File Formats	355
	Understanding Files	357
	Binary File Formatting	
	ASCII File Formatting	
	Tagged File Formats	
	Position-Specific File Formats	
	Magic Numbers	
	Next Steps	
Chapter 19	Save and Restore	363
Chapter 19	Save and Restore File Format Strategy	363 363
Chapter 19	Save and Restore File Format Strategy Save and Restore Program Hooks	363 363 366
Chapter 19	Save and Restore File Format Strategy Save and Restore Program Hooks Common-Object Save and Restore	363
Chapter 19	Save and Restore File Format Strategy Save and Restore Program Hooks Common-Object Save and Restore Object-Specific Save and Restore	363
Chapter 19	Save and Restore File Format Strategy Save and Restore Program Hooks Common-Object Save and Restore Object-Specific Save and Restore Next Steps	363
Chapter 19 Part V: Add	Save and Restore File Format Strategy Save and Restore Program Hooks Common-Object Save and Restore Object-Specific Save and Restore Next Steps ing Objects to the Editor	363
Chapter 19 Part V: Add Chapter 20	Save and Restore File Format Strategy Save and Restore Program Hooks Common-Object Save and Restore Object-Specific Save and Restore Next Steps ing Objects to the Editor Latex Line Object	363 363 366 367 373 376 277 379
Chapter 19 Part V: Add Chapter 20	Save and Restore File Format Strategy Save and Restore Program Hooks Common-Object Save and Restore Object-Specific Save and Restore Next Steps ing Objects to the Editor Latex Line Object Creating a Latex Line Object	363 363 366 377 373 376 277 379 380
Chapter 19 Part V: Add Chapter 20	Save and Restore File Format Strategy Save and Restore Program Hooks Common-Object Save and Restore Object-Specific Save and Restore Next Steps ing Objects to the Editor Latex Line Object Creating a Latex Line Object Drawing and Erasing a Line Object	363
Chapter 19 Part V: Add Chapter 20	Save and Restore File Format Strategy Save and Restore Program Hooks Common-Object Save and Restore Object-Specific Save and Restore Next Steps ing Objects to the Editor Latex Line Object Creating a Latex Line Object Drawing and Erasing a Line Object Finding a Line Object	363
Chapter 19 Part V: Add Chapter 20	Save and Restore File Format Strategy Save and Restore Program Hooks Common-Object Save and Restore Object-Specific Save and Restore Next Steps ing Objects to the Editor Latex Line Object Creating a Latex Line Object Drawing and Erasing a Line Object Finding a Line Object Selecting and Deselecting a Line Object	363
Chapter 19 Part V: Add Chapter 20	Save and Restore File Format Strategy Save and Restore Program Hooks Common-Object Save and Restore Object-Specific Save and Restore Next Steps ing Objects to the Editor Latex Line Object Creating a Latex Line Object Drawing and Erasing a Line Object Finding a Line Object Selecting and Deselecting a Line Object Moving a Line Object	363
Chapter 19 Part V: Add Chapter 20	Save and Restore File Format Strategy Save and Restore Program Hooks	363
Chapter 19 Part V: Add Chapter 20	Save and Restore File Format Strategy Save and Restore Program Hooks Common-Object Save and Restore Object-Specific Save and Restore Next Steps ing Objects to the Editor Latex Line Object Creating a Latex Line Object Drawing and Erasing a Line Object Selecting and Deselecting a Line Object Scaling a Line Object Copying a Line Object	363
Chapter 19 Part V: Add Chapter 20	Save and Restore File Format Strategy Save and Restore Program Hooks Common-Object Save and Restore Object-Specific Save and Restore Next Steps ing Objects to the Editor Latex Line Object Creating a Latex Line Object Drawing and Erasing a Line Object Selecting and Deselecting a Line Object Scaling a Line Object Saving and Restoring a Line Object	363
Chapter 19 Part V: Add Chapter 20	Save and Restore File Format Strategy Save and Restore Program Hooks Common-Object Save and Restore Object-Specific Save and Restore Next Steps ing Objects to the Editor Latex Line Object Creating a Latex Line Object Drawing and Erasing a Line Object Finding a Line Object Selecting and Deselecting a Line Object Scaling a Line Object Scaling a Line Object Saving and Restoring a Line Object Saving and Restoring a Line Object Next Steps	363

Chapter 21	er 21 Pencil Line Object	
	Creating a Pencil Object	406
	Pencil Object Management	410
	Next Steps	410
Chapter 22	Object Templates	411
	The Box Object	411
	The Arrow Object	415
	Next Steps	419
Chapter 23	And Object	421
	Creating an Arc Object	
	Drawing and Erasing an Arc Object	427
	Finding an Arc Object	
	Selecting and Deselecting an Arc Object	
	Moving an Arc Object	
	Scaling an Arc Object	
	Copying an Arc Object	
	Saving and Restoring an Arc Object	
	Next Steps	430
Chapter 24	Vector Text Object	437
	Creating a Text Object	438
	Drawing and Erasing a Text Object	461
	Finding a Text Object	
	Selecting and Deselecting a Text Object	
	Moving a Text Object	
	Scaling a Text Object	
	Copying a Text Object	4/2
	Saving and Restoring a Text Object	4/3
	INext Steps	
Part VI: Add	ing a Print Driver	477
Chapter 25	Introduction to PostScript	479
	PostScript	479
	Learning PostScript	480
	Stacks	481
	PostScript Commands	
	Comments	
	PostScript Programming	
	Viewing PostScript Files	
	Comments Understood by Ghostscript	
	Next Steps	490

Determining a Printer's Capability Defining Color Images for Black and White Printers Next Steps	491 492 494
Defining Color Images for Black and White Printers Next Steps	492 494
Next Steps	494
Observer 07. Missish with Vinesees and Oslaman	
Chapter 27 working with Ximages and Colormaps	495
Printing the Canvas	495
Creating an XImage	497
Creating a PostScript Prolog	499
Parsing the X Colormap	500
Writing the PostScript Page Definition File	502
Directing the Output to a Printer or File	504
Port VIII - What's Novt?	507
Part VII. WIIALS NEXL?	507
Chapter 28 Extending the Graphics Editor	509
Attributes	509
Color	509
Line Attributes	510
Arc Angles	511
Kotating Objects	
Chapter 20. Adding Context Sensitive Help	
	513
Processing Help-Kelated Events	
Relating Widgets to Text	518
Next Steps	519
Port VIII Annondivoo	501
Fait viii. Appendixes	JZT
Appendix A: Command Shells and Scripting	523
UNIX Command Shells	523
Command Shell Environment	525
Scripting with the Bourne Shell	531
Shell Variables	532
Writing a Script with Function Calls Debugging Shell Scripts	536
Annendix R: Annlication Layout Code Listing	520
mponiais D. mppinalion Layour oduc Listing	540
Make. der i le File Contents	
gxMain.c File Contents	542

	gxGraphics.c File Contents	544
	gxGx.c File Contents	547
	gxArc.c File Contents	549
	gxLine.c File Contents	
	gxText.c File Contents	
	gxGraphics.h File Contents	
	gxIcons.h File Contents	
	gxBitmaps.h File Contents	
	gxProtos.h File Contents	555
Appendix C:	Additional Vector Font Sets and vector_chars.h	557
	Triplex Bold Italic Vector Font Set	
	The vector_chars.h Header File	
	Index	741

About the Author

J. Robert Brown started his path to a career in software development by earning a college scholarship for Performing Arts in his homeland of central Ohio, where he held the misguided belief that he could be a movie star.

After years of either sleeping in his car or working three jobs concurrently to fund his way through an Electrical Engineering program, he found himself in Europe in the late 1980s working for the Department of Defense.

As a field engineer maintaining the mobile computer systems responsible for collecting and processing intelligence data, he realized that the position required too much manual labor. In 1991, he made his way through a Computer Science program at the European Division of the University of Maryland and although he didn't exactly finish in the top 10% of his class, he believes strongly that he helped those who did to get there.

John was invited to join Los Alamos National Laboratory as a Computer Scientist in 1996 where he remained until only recently. He now works for GTE Data Sources near Tampa, Florida.

Dedication

There are people who exist in the world who, once you've encountered them, change you forever. Through the strength of their character, depth of their spirit, or simply their presence in the world, they leave a lasting impression. I fear that we have one fewer such individual today due to the loss of Shel Silverstein in May, 1999. I hope for everyone there is someone who touches his or her life as Shel's works have touched mine.

Those without whom my life would mean less and this effort would not have been possible are my dear mother, Cindy Baker; my brother and best friend, Scott Brown; the absolute love of my life, Mikeala Elise; and the person who gave her to me, Kinnina McCray.

Acknowledgments

X Window Programming from Scratch is the result of efforts by many people. I am filled with awe and gratitude for the level of professionalism and quality the staff at Que publishing brought to this effort: specifically, Hugh "Red" Vandivier, Susan "I need this back by Monday" Moore, Cynthia "Did this change your meaning?" Fields, and Katie "You'd better meet the deadline" Purdum. And although no one likes to be told they have made a mistake, technical editor Ed "Um, you might want to check this" Petron was able to point out oversights in a way that never came close to wounding my ego. With the help of these and many others behind the scenes, this text is much better than I could ever have made it on my own.

There are others who contributed indirectly to the project by offering their friendship, encouragement, and patience as I tried to keep my head above water: namely, Mike Koscielniak, Cindy Sievers, and Jennifer Brown.

Tell Us What You Think!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

As an Associate Publisher for Que, I welcome your comments. You can fax, email, or write me directly to let me know what you did or didn't like about this book[md]as well as what we can do to make our books stronger.

Please note that I cannot help you with technical problems related to the topic of this book, and that due to the high volume of mail I receive, I might not be able to reply to every message.

When you write, please be sure to include this book's title and author as well as your name and phone or fax number. I will carefully review your comments and share them with the author and editors who worked on the book.

Fax: 317.581.4666
Email: quetechnical@macmillanusa.com
Mail: Tracy Dunkelberger Associate Publisher Que Corporation 201 West 103rd Street Indianapolis, IN 46290 USA

Introduction

Welcome to *X Window Programming from Scratch*. You'll soon discover that there is much more to the text than the X Window System.

Because the X Window System is an environment, it rests upon a programming language as well as on an operating system. Skills using these as well as elements of object-oriented methodology, trigonometric and geometric functions, and the PostScript language will be detailed in this text.

Clearly, this text is not for the faint of heart.

Instead of requiring you to purchase separate manuals for the X Window System, programming language, operating system, and so forth, the intent of this text is to provide an erector set for computer programmers. Within the covers of this text are all the pieces necessary for accomplishing a Graphics Editor project.

Section One, "Starting Points," introduces the many pieces and encourages you to examine the ones with which you are less familiar.

Section Two, "Graphics Editor Application," leads you through the assembly of the pieces into a functional and rewarding project. You will be challenged to further the project and integrate the editor into other applications that you may be responsible for professionally.

Beyond the Title

Window-based user interfaces are the mainstream in professional level software development, and the X Window System holds its share of the community.

The title of this book indicates the attention that this text will pay to learning the X Window System, but there is more in store than just learning X.

The programming language selected for use in the text to employ the X Window System is the C programming language. This decision was made because of the frequency in the professional community of using C when doing X Window System programming.

Other languages can be used, but the approach is less direct and not suitable for introducing the environment.

The X Window System is *non-proprietary*. Not only can it be used with any operating system, but the source code is freely available as well.

Therefore, the next decision to make was what operating system to use. The decision was obvious: The targeted operating system is Linux, although the project has been tested on several operating systems.

According to the January 2000 issue of *ComputerWorld*, of all PC operating systems in use, Linux holds the lead with 38% of the market and growing. A close second is Windows NT with 25% of the market.

Honing skills in the Linux operating system is imperative for continued success in the software development profession.

Software Checklist

The required components for accomplishing the Graphics Editor project in this text include

- Linux Operating System
- C Compiler
- X Window System

Linux Operating System

You have the following choices when acquiring the Linux operating system:

Downloading Linux from the Internet

The Linux operating system is freely available for download from the Internet. However, it is not small, and if being pulled through a modem, the process may take awhile. Further, if you download it, you have to work out the kinks of installing and supporting it.

Generally, when downloading Linux the process of installing it consists of transferring the many files comprising the Linux installation components to floppy disk. Using a utility called rawwrite.exe provided with the download, you must create a *boot* disk and a *root* disk. The images from which you create these disks are selected based on the system you are installing Linux on and the features of the hardware you want to be supported.

For instance, unique images exist for a variety of network cards, modems, video cards, and so forth. Review the README file to understand the differences between available images before creating your startup disks.

After an image is selected and the necessary disks are created, you can put the boot disk into your floppy drive and reboot to begin the first of many installation attempts.

I believe strongly that the following generalization is true:

No one ever installs Linux once.

After several iterations and plenty of research, you'll have a mostly functional operating system.

Alternately, you can purchase a distribution of Linux.

Linux Distributions

Linux is a free operating system protected by the GNU General Public License. However, many vendors offer for sale a Linux *distribution*.

A distribution that requires you to pay more than the cost of the media and postage is generally enhanced in one of many ways.

Either by adding an installation program that automates the selection of the proper Linux kernel, or by adding features to the environment such as utilities to configure your windowing environment or system options, the vendors earn and justify the costs of a distribution.

The vendors advancing Linux include Red Hat, Slackware, Debian, SuSE, and others.

Having experience with all of them, I am unabashedly (and free of charge) going to recommend Red Hat's Linux.

The kernel (core of the operating system) packaged with Red Hat's distribution is not different from the version you could download from the Internet or purchase in another distribution; however, the ease with which the Red Hat distribution installs, configures, and updates is worth the investment.

During the authoring of this text and the development of the Graphics Editor project, I used Red Hat 6.1. This version is packaged for very easy installation and configuration with the XFree86 X Window System and the GNU C Compiler.

The default X Windowing environment provided by the installation of Red Hat 6.1 is the GNOME desktop using the Enlightenment Window Manager. This is reflected in the screen shots used in the text.

To learn more about vendors distributing Linux as well as the availability for downloading the Linux operating system, issue the following command in the search field of your favorite Internet portal:

url: Linux

Because of the popularity of the Linux operating system, you will have to sift through many hundreds of matches.

Optionally, you can go directly to http://www.redhat.com and read about their latest release.

C Compiler

If you opt not to use a version of Linux provided by a vendor, it can be necessary to download and install a C language compiler separately.

Packaged with most distributions of Linux (including Red Hat) is the GNU C Compiler.

Because Red Hat's distribution of Linux used during the writing of the text includes the GNU C Compiler, this is the primary compiler used during the development of the project.

The project has been tested using other compilers and the text addresses compiler differences for managing the project, so feel free to use any C compiler available to you.

If you are using a version of Linux (or UNIX) that does not include a C compiler, it will be necessary to acquire one. The GNU C Compiler is available for free down-load from the Internet.

Refer to the Free Software Foundation Web site (http://www.fsf.com or search for url:gnu using your favorite browser) for information on sites providing the latest version of the compiler and its associated tools.

X Window System

The X Window System is free for downloading and is also provided with every distribution of Linux purchased from the vendors mentioned above.

Visit www.xfree86.com for the latest runtime and development environments included with most Linux distributions.

Optionally, you might be able to ftp (File Transfer Protocol) the X Window System from a variety of sites on the Internet, including gatekeeper.dec.com and uunet.uu.net.

Who Should Read This Book?

The text has several starting points, allowing readers in the range of seasoned professional developer to serious hobbyist to enter at the point they are most comfortable.

Whether the choice is to start at Chapter 1 and learn the basics of the Linux operating system or to delve into Part Two and begin structuring the project, readers with a wide range of abilities will benefit from this text.

This book is intended for those seeking to

- Learn or further C programming skills
- Create Graphical User Interfaces with the X Window System
- Employ the Linux operating system for software development
- Gain a greater knowledge of computer graphics programming
- Challenge their computer problem-solving skills

The learning curve for the text rises very quickly. Therefore readers should have some understanding of structure programming concepts.

Conventions Used in This Book

Some of the unique features in this series include



Geek Speak. An icon in the margin indicates the use of a new term. New terms appear in the paragraph in *italics*.

how too pro nouns' it

How To Pronounce It. You'll see an icon set in the margin next to a box that contains a technical term and how it should be pronounced. For example, "cin is pronounced *see-in*, and cout is pronounced *see-out*."

EXCURSIONS

Excursions. These are short diversions from the main topic being discussed, and they offer an opportunity to flesh out your understanding of a topic.

Concept Web. With a book of this type, a topic can be discussed in multiple places as a result of when and where we add functionality during application development. To help make this all clear, we've included a *Concept Web* that provides a graphical representation of how all the programming concepts relate to one another. You'll find it on the inside front cover of this book.



Notes give you comments and asides about the topic at hand, as well as full explanations of certain concepts.



Tips provide great shortcuts and hints on how to program more effectively.



Warnings warn you against making your life miserable and help you avoid the pitfalls in programming.

Code listings are provided throughout the book. Each code listing has a heading, and these are numbered sequentially within a chapter.

In addition, you'll find various typographic conventions throughout this book:

- Commands, variables, and other code stuff appear in text in a special monospaced font.
- In this book, I build on existing listings as we examine code further. When I add new sections to existing code, you'll spot it in **bold monospace**.
- Commands and such that you type appear in **boldface type**.
- Placeholders in syntax descriptions appear in a *monospaced italic* typeface. This indicates that you will replace the placeholder with the actual filename, parameter, or other element that it represents.
- This symbol → at the start of a line of code means that a single line of code is too long to fit on the printed page. Continue typing all characters after the → as though they were part of the preceding line.

Getting Started

If you're motivated by the many benefits previously outlined and have assembled the necessary software, you are ready to begin.

"Part One: Starting Points" provides the information needed to help you get started.

Starting Points

The complexity of the Graphics Editor can introduce many concepts that are unfamiliar to you.

I address as well the likelihood that I write for an audience gathered from a variety of backgrounds, interests, and experience levels. Therefore, you, the reader, must choose where you enter the text.

You might be comfortable with some of the ideas and disciplines employed by the Graphics Editor project and not with others. However, another reader might have confidence in areas you have not been exposed to during your pursuits.

Therefore, the first portion of this text provides a variety of starting points. Choose the one that best addresses your needs.

Where to Begin

I recommend that you spend sufficient time reviewing the areas that are new or less familiar to you and perhaps give only a cursory review of the subjects in which you are already confident.

What's at the End

Upon completing this section of the text, I expect all readers to have the same foundation, understand the same vernacular, and be prepared for the next section of the text.





Absolute Zero

Chapter 1

In this chapter

- The man Command
- Organization and Navigation
- The C Compiler
- Object Files
- Source Files
- The vi Editor
- The make Utility
- System Tools and Useful Commands
- Next Steps

UNIX for Developers

The development of each phase of the Graphics Editor requires use of a unique skill set. As an application developer, you must become acquainted with many aspects of computer problem solving. Knowledge of the operating system, programming language, windowing environment, and project management is critical to excel in the trade. This chapter introduces the operating system, compiler, editor, and project files in sufficient depth to put you on the path of completing the Graphics Editor project.

Structured application development (writing computer programs using an applied methodology) begins with the operating system. At this level, you arrange, navigate, edit, and compile your source code. If your organizational skills are poor, you won't be able to locate the file you need to advance the functionality of your application. An inability to utilize the capabilities of an editor fully means you will be making changes (assuming you can find the file) one character at a time when you could be manipulating pages or regions. Worse, when the compiler tells you syntax error: line 1162, you have to press the down arrow key 1,162 times to fix the problem because you don't know the go-to command.

Looking into the Linux operating system, you see that many tools (commands) are available to you. Some are sharp and can do damage if used incorrectly; others are obscure and hard to understand, and not all of them are useful for programmers.

In the following sections we'll take a few of the more pertinent tools out of the box and look at them together. If you are already well acquainted with the Linux operating system, feel free to move ahead to the next section.

The man Command

Having something to do with everything, man (short for *manual*) offers help for most all Linux commands. Think of man as a librarian and ask it questions when you want to read the manual for a specific topic.



The output of man is called a *man page*.

Examining the use of man will give insight into the way that most every command in Linux is formed. The basic syntax follows the pattern:

```
command -flag(s) parameter(s)
```

A *flag* is usually a single letter following a hyphen (-) meant to instruct the command to alter its behavior. Further, the command acts upon the parameter(s).

Consider an example where man is the command, there are no flags specified, and *man* is the parameter to be acted on:

bash[1]: man man

Issuing the command man man results in the man command using its default behavior to display the man page for itself. Only a portion of the command output is shown in Figure 1.1; however, it is important that you execute the command on your system and review the entire output.



Note

Commands distinguish between flags and parameters by use of the hyphen (-). A hyphen indicates to the command that what immediately follows is a flag.

You can know what flags or parameters a command accepts by looking at its man page. Anything placed between square braces ([]) in the Synopsis area of a command's man page is optional input. Reading the man page for man, you see the flags acdfFhkKtwW fall within an optional list; however, a topic name does not, and therefore must be specified.

If you use man followed by -k, the output is much different (as shown in Figure 1.2). Instead of getting the man page for a topic, you get a list of pages containing the string specified.



EXCURSION

An Alternative to the man -k Command

A special notice of the -k option to man: On some UNIX systems man -k is a separate command called apropos, which is...well, more appropriate! Although the man command and its -k flag will always exist, apropos can be an abbreviated option to the man -k command on your version of UNIX.

The man command is a suitable first command to discuss because it will assist you in learning other commands to add to your repertoire. Feel free to issue the man command for any commands or keywords that appear in upcoming sections.

1

Organization and Navigation

The organization of a new project must be carefully planned. Consider as an example the placement of files used by the Linux operating system. With great consistency, the directory structure and file locations in one version of UNIX (within a family) will resemble another.



If you want to affect the way your operating system works, you know (based on the flavor of UNIX) where to find the configuration file needed (see Figure 1.4). Logical organization and consistency mean you don't have to search for the correct directory and guess at the name of the file.



How does something with the complexity of an operating system compare to application development?

Professional-level software development implies a life cycle: The software has a starting point, it grows (sometimes painfully), it is changed, and eventually it matures. Even if you are the only programmer to support this cycle, you will have other projects during the life of this one. The organization and structure of the project should enable you to find the directory, file, or function you need by a process of induction.



Using *induction* to find a file or function means that by observing the conventions used in any piece of the application, you can surmise (guess) the name and location of the module you are seeking. Contrast this to a *deductive* process where you would examine every directory and every file one at a time, stopping only when you find what you are looking for.

The example of poor project structure illustrated in Figure 1.5 might be an extreme, but I've witnessed similar structures used in real-world projects. Clearly, moving from one directory or subdirectory to another is arbitrary. There is no clear way to know which place to go to find anything.

9

1



Reuse of directory names at multiple points in the poor project structure illustrated in Figure 1.5 and the seemingly meaningless names chosen make it impossible to induce the location of a file or function within the project. In fairness, even meaningless directory names could contribute to good organization if, for instance, all files and functions named in the directory were prefaced with the directory name.

After you've decided on the basic organization of your project, you are ready to open your Linux toolbox and begin. The structure of the Graphics Editor project is shown in Figure 1.6.



Directories

As George Carlin pointed out, having a place for our stuff is a long-standing concern of mankind. The age of technology offers relief from this dilemma, however, because directories provide locations for storing the files created as part of a project. In fact, directories are the basic building blocks to a project's structure, allowing you to apply the organization that was carefully planned.

The mkdir Command

The Linux command to create a directory is mkdir. It stands for "make directory." Issuing the command with only a name causes the new directory to be placed in the current working directory. If you give an explicit or relative path to the command, the new directory is placed there.



An *explicit* path is one where every component of the path is spelled out:

mkdir /home/users/jbrown/2d-editor

In a *relative* path, shortcuts are used to specify the location. UNIX provides two valid shortcuts for your use. They include a single dot for referring to the current directory (.) and two dots for referring to the current directory's parent directory (..). For instance, by using .../ in a directory specification, the path will be relative to the directory one level up:

mkdir ../2d-editor

In this example, the new directory 2d-editor is placed in the directory above the current one.

Alternately, the *command shell* being used can provide other shortcuts.

EXCURSION

What Is a Command Shell?

A command shell (or simply shell) is assigned to every user of a UNIX system. The shell provides a window into the system, enabling you to issue commands, view the output, and navigate the file system.

Commands available to a user are grouped into two categories: internal and external.

External commands are provided by the UNIX operating system and will be available to all users regardless of the shell assigned because external commands reside as executable files on the UNIX file system.

Internal commands are shell specific, meaning that shells differ in the commands they provide internally. Generally, internal commands offered by a shell determine its suitability for scripting as discussed in Appendix A, "Command Shells and Scripting."

Just as shells differ in the internal commands they understand, they differ in the syntax or conventions they accept. These differences determine, in part, the shortcuts that the shell understands.

Table 1.1 provides a brief description of common UNIX shells. A more detailed discussion of command shells, conventions, and scripting can be found in Appendix A.

Shell Name	Command	Description
Bourne	sh	The Bourne shell is the original shell provided with UNIX. Internal commands understood by sh and its derivative ksh (Korn Shell) include export, for, while, and case. These shells provide no navigational shortcuts.
С	csh	The C shell, and its variation the tcsh (T C Shell), fol- low closely the syntax of the C programming language. Internal commands include setenv, foreach, and while.

Table 1.1 **UNIX Shells**

1

Table 1.1 Continued

		Shortcuts include the tilde (~/) for referencing your home directory or ~userid/ for referencing another user's home.
		For instance, csh and tcsh accept the following relative path: mkdir ~/2d-editor.
		This command creates a directory called 2d-editor in your home directory.
Bourne Again	bash	The Bourne Again Shell is a combination of sh, ksh, and csh conventions.
		As the default shell assigned to user accounts under Linux, bash is a very capable shell. The shortcuts under- stood by csh are also known to bash.

The rmdir command

To remove a directory, use the command rmdir:

```
rmdir /home/users/jbrown/2d-editor
```

The rmdir command, which stands for "remove directory," only works if the directory is empty; this is important to remember. To remove a directory with something in it, issue the rm -r command:

```
rm -r /home/users/jbrown/2d-editor
```

The rm command with the -r flag is one of the sharpest tools in our box! You use the rm command to remove a file, but when told to be recursive (-r), directories and their contents are considered for removal as well, even if the directory contains other directories and those directories contain other directories, and so on. The -r flag sends the rm command to the lowest level of the directory structure, where it begins removing everything it finds as it works its way back up to the starting point. This is illustrated in Figure 1.7.

When using the rm command, you must specify the name of what you want to remove.

I∎ **⊂** Note

The way to specify names for command completion under Linux can be very flexible when employing wildcards. Of course, this flexibility makes the rm command even more dangerous.

1



Unlike other operating systems, Linux has no undo or undelete command for recovering files removed in error. When you unintentionally delete something (notice I said *when* and not *if*), remember two magic words and utter them over and over until you get a response from your system administrator. The words are *backups*. I guess that's only one word. You may have to append *please* to it for it to be magic!



Wildcards for command-line completion use a single character to represent multiple characters.

Commonly understood wildcards and their functions include:

*	Replaces any number of characters (broadest substitution)—for example, bash[2]: rm ca* expands to remove <i>capable, cannibal, canister, cat, cap,</i> <i>cal,</i> and <i>cab.</i>
?	Replaces only a single character—for example, bash[2]: rm ca? expands to remove only cat, cap, cal, and cab.
[a-d]	Replaces only occurrences of a single letter which falls between the specified range for example, $bash[2]: rm ca[a-d]$ removes only <i>cab</i> .
Not to minimize th	a amount of aution that should be used when every signs the pr

Not to minimize the amount of caution that should be used when exercising the rm command, especially if using wildcards, but a safety net, called *permissions*, is built into Linux. A look at permissions follows in the next section on directory listings.
The 1s Command

Requesting a directory listing determines whether anything is in a directory. The listing command is 1s, and it accepts many flags to alter its behavior or output.

The -1 flag passed to 1s tells it to issue a long listing. A long listing includes information about permissions, ownership, modification date, and size. The default output for 1s (no flags specified) is to list only the names of the files. Any file with a period (.) as the first character of its name is considered a *bidden file*, and you must explicitly request that 1s display such a file by passing the -a flag.



When specifying multiple flags to a command, group them together following a single hyphen. Everything after the hyphen and until the next space is interpreted as flags and acted on accordingly. Issuing ls -al requests that the listing command displays all files in a long listing format.

Figure 1.8 shows sample use of the 1s command, including examples of the -1 and -a flags.





Notice that when displaying all (1s -a) files in a directory you see . and . . listed. The . (current directory) and . . (parent directory) are special files Linux uses to maintain information about these directories. When you ask for a listing of your current directory, 1s displays some of the information contained in the . file.

how too pro nouns' it

A . in UNIX is always pronounced as dot.

1

Permissions

Permissions serve several functions within the Linux operating system. They provide security, safety, and control. However, as with any tool, they must be used correctly. You must give thought to the permissions that are applied to the files and directories you create and manage.

You can find permissions for a specific file or directory in the first column of a long listing (1s -1) as you saw in Figure 1.8. Several groupings make up the permission field. Dissecting them into their groupings will make permissions very easy to understand.

```
drwxrwxr-x 9 jrb jrb 4096 Nov 7 05:32 ObjCntrl
```

The first character of permissions is the *designator field*. This character indicates the type of entry in the listing. By the d in the first column of the example, you know that the entry is a directory. Other possible characters found in this position are - (hyphen) when the entry is a file or 1 when the entry is a symbolic link. I haven't discussed links yet but you'll soon see their usefulness to multi-platform application development. For now, remember what it means when you see an 1 in the designator field of permissions.

```
drwxrwxr-x 9 jrb jrb 4096 Nov 7 05:32 ObjCntrl
```

The next grouping to consider in the permissions is the *owner access* field. As the name implies, this determines the accessibility for the owner. Clearly, owning the entry means that as owner you can alter any access field within the permissions. However, access doesn't just limit; it also protects. If used properly, you will have a safety net that prevents accidents.

The first character of the owner access field governs read permission. The presence of the r in the first column of this field indicates that read access has been granted. If access were not granted, there would be a - (hyphen) placeholder. The second character of the owner access field determines write permission, where a w indicates the capability to write to the file, and a - (hyphen) placeholder means that no write permission is granted. Last is execute permission as seen with either an x indicating permission to execute or a - for no permission to execute.

III ⊂ Note

Having execute permission is not always meaningful. For instance, writing a letter to a loved one and assigning execute permission to the file does not cause anything useful to happen. The file does not become a command by simply granting execute permission. (If accomplishing things on a computer were so easy, you probably wouldn't need this book.) However, if the file you create is a series of valid Linux or shell commands, giving execute permission creates a script or shell script for which execute permission is meaningful. (Shell scripting is discussed in Appendix A.)

drwx**rwx**r-x 9 jrb jrb 4096 Nov 7 05:32 ObjCntrl

Following the owner access field is the field governing *group access*. In addition to a unique ID assigned to all Linux user accounts, everyone is placed into at least one group. This grouping is useful for allowing files and directories to be shared.

EXCURSION

What Group Am I In?

You have several ways to determine into which groups you have been placed. For instance, you can execute the id or the groups command. The output of the id command provides a lot of information about your Linux account, including the group assignments. The groups command will simply list the groups to which you are assigned.

The group access field manages sharing within a group. Applying the same read/write/execute pattern as for the owner access field, you can quickly determine that in the example anyone belonging to the same group as the group ownership of this entry may read, write, and execute.

EXCURSION

How to Determine Group Ownership

The SYSV family of UNIX does not show what the group assignment for an entry in a directory listing is unless you explicitly ask. To request that group ownership be displayed in the output of the 1s command, you must pass the -g flag.

drwxrwxr-x 9 jrb jrb 4096 Nov 7 05:32 ObjCntrl

The last field represents permissions that pertain to a collection of users known as the *world*. Any user who does not own the entry and is not assigned to the same group as the group ownership is considered part of the world. Users accessing the entry based on the merit of world permissions include anyone who shares an account on the system (and possibly network). For this reason, world permissions are generally not very giving, as you have no way to know or control those in the world beyond these permissions.

In the example, those in the world may only read and execute the entry. Note that a - placeholder resides in the write column, indicating no write access is granted.

Beyond the Obvious

What exactly can a user do with read, write, and execute permission?

Read Permission

Granting *read permission* enables users not only to view what has been placed into a file or directory, but also to copy it. In fact, you must have read permission to execute the cp (copy) command successfully.

Write Permission

Write permission allows users to modify any entry for which they have permission to write. Most importantly, though, having write permission enables users to remove a file or directory. You must have write permission to execute the rm or mv (move) command successfully. For this reason, if you are not actively changing a file or directory, removing write permission also removes your ability to delete a file in error.

Execute Permission

Being able to execute a file, as discussed earlier, means that each of the commands the script contains is performed as if it were typed at the command line. In the case of a compiled program, execute permission allows the user to run the program. However, if the entry is a directory, execute permission is required to change into that directory. If you do not have execute permission for a directory, it is effectively closed and the cd command will fail.

chmod

With an understanding of the power and utility of permissions, you must know how to change them. The chmod command, which stands for "change mode," allows you to modify the permissions of anything you own.

how too prō nouns' it

The chmod command is pronounced as if it were written change mod.

Syntax of the chmod command is direct.

```
chmod [ugoa][+ or -][rwx] name
```

where:

Item	Means
u	user/owner
g	group
0	world/others
а	all
+	add

-	remove
r	read
W	write
х	execute
name	item to be changed

For instance, the following command changes the permissions of ObjCntrl from the examples above to add write permission for the world:

bash[5]: chmod o+w ObjCntrl

To remove execute permission for both group and world, you could use the following command:

bash[5]: chmod go-x ObjCntrl

Any questions? Just ask man (man chmod).

The cd Command

A discussion on organization and navigation would not be complete if I didn't mention something about navigation. The cd or *change directory* command navigates around the Linux file system. It expects a single parameter to indicate to which directory you wish to change. If you execute the cd command without any parameters, you are returned to your home directory.



Unlike other operating systems, Linux is case sensitive. When trying to change to a directory you know exists but Linux disagrees with, check the case because ./src and ./Src are unique names to Linux.

With an understanding of basic Linux commands, you are ready to advance to commands specific to software development.

The C Compiler

The C Compiler, known as cc, translates *source code* that a programmer has written into *object code* that the machine understands.

Source code can be read and written by you, a programmer. It spells out programming language elements such as keywords, variables, start of body markers, and end of body markers. A computer, however, knows nothing of the letters or characters that are meaningful to programmers. Instead, the computer expects that when you

1

enter the letter A as part of a program solution, it will be translated into the machine code equivalent 0100 0001. Accomplishing this translation is the function of the compiler..

EXCURSION

A Closer Look at Machine Code

Machine code is ultimately a collection of 0s and 1s, which is all any machine can understand. At this level the collection of 0s and 1s is represented in the *binary* numbering system. Unlike the everyday *decimal* numbering system that consists of ten digits (0-9), the binary numbering system has only two digits, 0 and 1. Use of binary numbering to represent machine code can be very tedious to humans, so we also use the hexadecimal (16 digits) and octal (8 digits) numbering systems.

Table 1.2 shows various characters represented using the different numbering systems.

Tab	Table 1.2 Data Representation					
	Character	Decimal	Binary	Hexadecimal	Octal	
	1	49	0011 0001	0x31	\0061	
	2	50	0011 0010	0x32	\0062	
	>	62	0011 1110	0x3E	\0076	
	@	64	0100 0000	0x40	\0100	
	A	65	0100 0001	0x41	\0101	
	В	66	0100 0010	0x42	\0102	
_	а	97	0110 0001	0x61	\0141	

Determining which numbering system to use is largely a matter of convention based on how the data is used. This decision is also influenced by the size of the data being displayed. For instance, representing addresses in computer memory is most often done employing the hexadecimal numbering system.

Data representation is largely for the convenience of programmers; this is important to remember. The computer only understands 0s and 1s.

Object Files

Stating that the compiler translates the letter A from the source file into the machine code equivalent for the object file is an over-simplification.

It is true that A is translated to its machine code equivalent, but only if it is a literal. If it is a keyword, it is translated into an *op code* (code specifying an operation). If A is part of a variable name, it is translated into an offset (distance from the value of an internal control pointer). If part of a function name, A is translated to a *jump point*

(distance from the base address of the application to where the function definition exists). If A is a function parameter, it is translated as an offset of the *stack pointer* (an internal control pointer that manages program execution), and on and on.

The complexity of the compiler is boggling, and I've only given a very high-level overview of its role. Entire books are dedicated to the subject of compiler design. For this reason, compilers are not generally given away for free, except, of course, the GNU C Compiler (gcc), which I will discuss shortly.

To review, an object file consists of only the machine codes (*op codes, jump points, off-sets*, and so forth) translated from what you've written in the source file using a high-level language such as C.

Object files have the suffix .o so that you can distinguish them from files you author. Any attempt to edit or view the contents of an object file will make it instantly clear that there is nothing in it that you can affect.

how too pro nouns' it

Object files, because of their .o extension, are referred to as dot-oh's.

Object files are platform specific: You cannot use an object file generated on an x86 PC platform on a platform of a different architecture. This is important to know.

The object file is the translation of what was written in the source file into a format that the computer understands. Not all computers do things in the same way. If you had an 80286-generation computer and you now own a Pentium, it is clear to you that processors are very different. The most notable difference in this example is speed. Performance, however, is not always a noticeable difference, but when it is, many factors contribute to it, such as the language set available to the processor or the manner in which it groups data into internal representations.

The Pentium processor is boasted to be a 32-bit processor, whereas the 80286 processors were only 16-bit architecture. The different number of bits (0s and 1s) supported by the processor means that the word size (internal bit groupings) will be different for each. Although a Pentium may be able to understand a 16-bit word size (grouping), an 80286 can never understand a 32-bit word.

EXCURSION

Computer Processor Instruction Sets

A processor's instruction set consists of the *op codes* (operational codes) available to it. These instructions are internal to the processor and dictate everything the processor knows how to do. Basic processor instructions include directions for moving data, arithmetic operations, program flow control, and more.

Instruction sets fall into two categories: Reduced Instruction Code Set (RISC) or Complex

Instruction Code Set (CICS)). RISC-based processors know fewer commands than CICS processors but operate more quickly as a result. In other words, the op codes contained within one processor's instruction set may not be the same as those of another processor's of a different architecture (and similar doesn't count). The differences in instruction sets of varying architectures introduce another level of binary incompatibility.

How does all this affect the object files? If you generated an object file on a 32-bit platform, the calculated offsets to memory for representing variables, for instance, would not align to valid memory addresses on the 16-bit system, nor would the jump-to addresses. The distance of the jump would likely be well outside the intended user memory area on the 16-bit platform. This illustrates, of course, one manner in which object files are platform specific.

I've barely mentioned the internal representation of data by a processor. In addition to word size (data grouping), some processors expect the order of the bytes (four-bit grouping) of data to be formatted differently from others.

One method of byte ordering is least significant bit first, called *little-endian*, and another is most significant bit first, called *big-endian*.



Think of little-endian as *little end first*, big-endian as *big end first*, and endian as *end-ed*.

EXCURSION

Waiter, I'd Like To Order Some Bytes.

The determination of which byte order to employ for a given platform's architecture is largely a decision of the design engineers. It has occurred in the history of processor design that the decision to elect one method of byte ordering over another was based solely on a desire to avoid patent infringement.

Source Files

Previously, I spoke at length on the complexity of an object file as generated by the compiler. Figure 1.9 demonstrates the compile process with emphasis on the source files as *plain text*, meaning that they are understandable to programmers and contain no word processor type formatting. The figure further illustrates that the output of the compiler is not in human readable format, but rather in a format that satisfies a computer's requirements.



Knowing the complexity of the compile process that a source file undergoes to be translated into something the computer can understand should make you appreciate that the compiler considers every character of a source file. In effect, as the compiler parses the source file it anticipates the next character or token it expects. If what is anticipated is not what it reads from the file, a syntax error is issued. For this reason, the source files must be plain text to prevent the extra control symbols used in word processing from confusing the very literal compiler.

Even the most basic text formatting performed by a word processor is accomplished by inserting control characters into the document so that the word processing application knows when to turn on and off the formatting. These control characters are, in fact, characters (not usually printable) that would be considered by the compiler as input. The choice of editor is important, as is how you choose to save the file if the editor supports multiple file formats.

An editor is available in all versions of UNIX to ensure that your authored source files contain only text. This is because it is incapable of word processing functions such as bolding, italicizing, and underlining.

The vi Editor

A popular choice among UNIX programmers is the vi editor. The fact that it is readily available on every version of UNIX, it does not require a lot of system resources, and it is fairly easy to master makes it a solid choice.

Certainly, other editors exist that satisfy the requirement of being able to save source files without any extra formatting data. A popular editor called GNU Emacs will enable you to write and save source files as required in addition to telling your fortune and generating whimsical quips. However, we must concern ourselves with availability and system requirements. Emacs is very large and will not always be installed on the system on which you must work.



Even if vi is not your choice for an everyday editor, knowing its basic operations is useful because the day will come when you are sent to a customer site or asked for help by a colleague and your normal editor is not available. Nothing shatters confidence more than seeing someone not know how to edit a file.

To start vi, simply execute the command specifying the name of the file that you want to edit:

bash[2]: vi gxArc.c

If the file didn't exist previously, vi informs you that it was created. When loading an existing file, vi informs you of the number of characters read.

There are three modes in vi: *command mode*, *last-line mode*, and *edit mode*. vi always starts in command mode waiting for input. To move from command mode to edit mode, you must enter an appropriate command. Some commands enabling you to enter edit mode from command mode are shown in Table 1.3:

Table 1.3	vi Commands to Enter Edit Mode		
0	begin inserting below current line		
0	insert above current line		
а	append following current cursor position		
i	insert at current cursor position		

After you have entered edit mode using one of these commands, everything you enter is inserted into the file.

To leave edit mode and return to command mode requires pressing the Esc key.

Other commands available in command mode are listed in Table 1.4.

Table 1.4	Other vi Commands Available in Command Mode
х	deletes character under cursor (when preceded by a number $(\#x)$, which deletes $\#$ of characters
c <spac< td=""><td>e> changes the current character</td></spac<>	e> changes the current character
cE	changes everything from current cursor position until the end of the current word
c\$	changes all characters from current position until the end of line
dd	deletes a line (can be preceded by a number to delete multiple lines)
d\$	deletes from current cursor position until end of line

continues

Table 1.4	Continued	
уу		yanks (cuts) current line (when preceded by a number (#yy) it will yank that many lines)
р		pastes everything in cut buffer (result of yy or #yy) after current line
Р		pastes everything in cut buffer above current line
0		goes to the beginning of the current line
\$		goes to the end of the current line
/str		forward searches for the next occurrence of str
?str		reverse searches for the previous occurrence of str
G		goes to the end of the file
:		enter last-line mode

Last-line mode of vi is identifiable by the presence of a colon at the bottom left corner of the window. This colon serves as an indicator of the last-line mode. As with edit mode, to return to command mode, press the Esc key.

Commands available in last-line mode are listed in Table 1.5.

Die	1.5 vi Last-Line	commanus
: v	V	writes the changes made to the current file
: v	v filename	writes the current file as filename
:0	1	quits this session of vi
:0	11	quits the current session without saving changes
: 6	e file	loads file for editing
: 6	91	reverts to the last saved version of the current file (discards all unsaved changes)
: r	file	inserts a copy of file at the current cursor position
:r	ı	when editing multiple files, tells vi to go to the next file in the list
:#	ŧ	# indicates a line of the file for vi to go to (as in :1162 to go to line 1162)

Table 1.5vi Last-Line Commands

Learning vi simply takes practice, but these basic commands will get you started.

Now able to create a project structure and edit source files, you are ready to learn how to manage building a project for multiple platforms.

The make Utility

The Linux make utility provides a means of managing projects, such as the Graphics Editor, that span multiple files. As illustrated in Figure 1.9, having many source files

requires that each one be passed to the compiler for translation into a corresponding object file. Issuing the C Compiler command (cc) for each source file is a cumbersome effort prone to mistakes and oversights. I have purposely postponed discussion of the syntax of the cc command because of its complexity. This complexity, however, may be masked if viewed as a function of the make utility.

The cc Command

Note

Under Linux a reference to cc is synonymous with the GNU C compiler gcc. If you are using another version of UNIX, gcc and cc might not be synonymous at all but might be two separate commands. Knowing which C compiler will be used is critical because how you construct the command line to invoke them is very different. For instance, gcc understands the flag -W as a means of determining the level of warnings that the compiler issues. Other C compilers have different flags for specifying warning levels. You pass gcc - ansi to instruct it to be ANSI C-compliant, but the Sun WorkShop Pro compiler expects a - Xn (where n is a directive for how ANSI C-compatible you want it to be).

EXCURSION

ANSI C Versus K&R C

The C programming language was written by Dennis Ritchie at AT&T in the early 1970s. With Brian Kernighan, he continued its development under what became known as K&R C. Unfortunately, some of the language details were unspecified, ambiguous, or incomplete. In 1983, the ANSI committee formed, and five years later C was standardized in what is now known as ANSI C.

The understanding to be taken from this discussion is the complexity involved when invoking a C compiler. If you use only Linux, the task is greatly simplified because you only must learn the gcc command and the flags and parameters that it understands. If your goal is to write software that is compatible with any family of UNIX, you must recognize compiler differences and how they affect the configuration of the make utility.

With an appreciation for the fact that there are differences in C compilers, we focus now on the gcc command, identifying what it has in common with other C compilers and contrasting differences where necessary.

qcc -c

The syntax of the gcc command follows most other UNIX commands with a few exceptions. As with other commands you've seen, gcc accepts flags that instruct it to

alter its behavior. Common to all C compilers is the -c flag, which tells the compiler to generate the object file.

With the command bash [3]: gcc -c gxArc.c

gcc stops after generating the gxArc.o file.

Why should you need to instruct the compiler to generate an object file? All along I've told you that the compiler automatically generates object files. The compiler does generate object files, but it also does more. The compiler links these object files together, forming an executable program.

As Figure 1.10 shows, an object file is the translation of a source code file, but it is not necessarily a complete program.



Modules such as gxColr might be invoked in one source file but actually defined in another. Notice in gxArc.c in Figure 1.10 that the variable color is assigned the result of module gxColr; however, the definition of gxColr is contained in gxGraphics.c. After the compiler converts the source code to object files, gxArc.o does not know the actual jump-to address for finding the gxColr module. Another task of the compiler is to link these references together so the program flows without dead ends. In gxArc.o, the jump-to address for gxColr is considered *unresolved*. Objects must go through the linker phase of compiling to determine what the external jump-to points should be.



A source file invoking a function (module) contained in another source file creates an *external dependency*. If, at the end of the link phase, any external dependencies are left unresolved (meaning that invocations from one file could not be matched up with definitions in another), the program will not successfully link. The error reported will be unresolved externals.

EXCURSION

A Program Must Know Where It Is Going

When a program invokes a function that doesn't exist or can't be found, the result is a *bus error*. It is similar to getting on a bus to go somewhere and finding out that the address you were given was incorrect. Often, bus errors are a result of corrupted memory. Rather than the function not existing, which is something the linker must validate, the address given as the jump-to was corrupted during execution and therefore the function couldn't be found.

gcc -o

Knowing that the -c flag passed to gcc instructs the command to stop after creating the object file, another flag common to all C compilers is the -o flag. The -o flag must be followed by a filename because it instructs the command what to name the output.

bash[4]: gcc gxArc.o gxGraphic.o -o 2d-editor

By convention, the name of the object file is the same name as the source file, differing only by the extension. However, if you didn't instruct the compiler to stop after creating the object files and allowed it to continue through the link phase to produce an executable, you would certainly want to give the executable a name. Without -o *filename*, the compiler would name the program a.out instead of something meaningful.

gcc -g

Another interesting flag instructs gcc to provide debugging information in the objects that it creates. The -g flag is known as the *debug flag*. Without it, the compiler streamlines the object data by stripping out information that a debugger would need to enable you to evaluate variables, set breakpoints, and examine program flow.

gcc -W

As mentioned earlier, the -W flag that instructs gcc what level of warnings to issue while compiling the source code is a flag unique to gcc. Invoking gcc with -Wall requests the most stringent level of warnings and aids in the prevention of some programming errors.



Programmers introduce three types of errors when writing source code: *syntax errors, semantic errors,* and *logic errors.*

The first level of errors is *syntax errors*, which the compiler easily finds. Syntax errors are created when you fail to obey the specifications of the language. For instance, in English, failing to place a period at the end of a sentence is a syntax error.

The second type of error is a *semantic error*, which occurs when you don't say what you mean to say. The -Wall flag passed to gcc is a great aid in discovering some semantic errors, because gcc analyzes context and usage as well as syntax.

Unfortunately, the third type of error, called *logic errors*, can only be found by exhaustively testing the program.

gcc -D

The -D flag is common to all compilers. This flag requires that a *directive* immediately follow it. This directive is then defined as a constant throughout the source file. Examples of how to employ compiler directives are provided in the next chapter.

bash[5]: gcc -g -ansi -Wall -c -DUSE_COLOR gxArc.c

As you can see, the examples for invoking the gcc command are becoming progressively more complex. In the preceding example, you are instructing gcc to include debugging information (-g), enforce the ANSI coding standard (-ansi), report all warnings (-Wall), stop after generating the object file (-c), and define the directive USE_COLOR. All of this is for the single source file gxArc.c.

The -I, -1, and -L flags are the last flags to consider before looking at how the make utility simplifies use of the gcc command.

gcc -I

The -I flag tells gcc where to find header files. Immediately after the -I flag you must provide a directory path that gcc will follow to find the header files that the source code has included.

III ⊂ Note

Like the gxColr function shown in Figure 1.10 to illustrate external dependencies, not all functions invoked in a source file will be present in the same file. These external function references must have a declaration before use; otherwise, the compiler makes assumptions about them that may not match with the actual definition. The declaration of a function before its definition is called a *forward declaration* or *prototype* and is generally placed in a header file. A *header file* is a source file that ends with a . h extension.

1

how too pro nouns' it

Following their naming convention, header files are called *dot-h files*.

Because header files can be placed almost anywhere the author deems appropriate, use of the -I flag instructs gcc where to search for them.

Because header files can be used to prototype different functions from different libraries (packages), multiple -I flags may be passed to gcc.

bash[6]: gcc -g -ansi -Wall -c -I ../src/include -I /usr/X11R6/include gxArc.c

This example instructs gcc to look in two places, ../src/include and /usr/X11/R6/ included, for the header files that are included in the source file gxArc.c.

gcc -l

The -1 flag tells gcc what extra libraries to include in the link phase of compiling. Using again the example of gxColr from Figure 1.10, not all functions that you invoke need to be in the source file where you use them. Further, they don't have to be in any source file that you author.

Groupings of functions for a general purpose need only be written once. After they are written, they may be included in a package or library for others to use.

For instance, if you want to find the square root of a number in C, you don't have to write the function to do it. Because it is such a common requirement, the authors of the C language have assembled a library of mathematical functions for your use. You can include in your source file the math.h header file which prototypes the square root function sqrt. Then, so the link phase of the compiling process knows how to resolve the external dependency, you must include the math library (libm.a) by use of the -1 flag passed to the gcc command.

bash[7]: gcc gxArc.o gxGraphics.o -lm -o 2d-editor

EXCURSION

How to Determine Libraries Names for Use by gcc

The gcc command expects that libraries specified with the -1 command follow the naming convention libname.a.

This explains why, in the previous example, gcc understands -1m to be libm.a (the math library) containing the definition of the sqrt function.

As with -I, multiple -I flags may be necessary if you are calling functions from several different libraries.

```
bash[8]: gcc gxArc.o gxGraphics.o -lm -lXaw -lXt -lX11 -o 2d-editor
```

Based on your newly acquired knowledge of the gcc command, you should be able to determine what is happening in the previous example. The command gcc is being invoked with the files gxArc.o and gxGraphics.o. (As these are already object files, you must assume that at some point prior to this command gcc was invoked with the -c flag for each of the related source files gxArc.c and gxGraphics.c.) Further, gcc is being instructed to consider three libraries for resolving external dependencies (libXaw.a, libXt.a, and libX11.a). Lastly, the command specifies that the output should be named 2d-editor (-o).

gcc -L

When instructing gcc to include libraries such as libXt.a or libX11.a, you must also tell gcc where to find them. Informing gcc of the location of libraries is done with the -L flag:

```
gcc gxArc.o gxGraphics.o -L /usr/local/lib -L /usr/X11R6/lib -lm \
-lXt -lX11 -o 2d-editor
```

Having only touched on the more commonly employed flags and parameters used with gcc, it is easy to see how cumbersome the command line could become as directives, header paths, and more and more source files are added to the project. Of course, needing to support multiple platforms where the locations of the libraries and header files differ and where the compiler flags vary would be almost impossible to do by invoking the gcc from the command line.

The make utility enables you to automate the invocation of the compiler (gcc) for every source file in your project. Further, make enables you to create dependencies or conditions that will force object files to be updated only when necessary. It also enables you to assemble the necessary parameter lists for the gcc command based on the operating system and platform being used. Only with the flexibility of make will you efficiently structure support for multi-platform development.

In general, make is a scripting language. It enables you to define variables, make decisions, and execute commands. The most obvious command for make to execute, of course, is the cc (or gcc) command. The uses of decisions within make support the task of multi-platform development. If you want the project to build under Linux *and* Solaris, it will be necessary to tell make the environment differences and let make decide which to employ based on the current platform. It is really not complicated, as you'll see with your first make script.

Makefile

Note

When invoking the make utility, you must provide it with a configuration file. If no configuration file is specified on the command line, make looks for the presence of a default file called Makefile. If you are using the GNU make utility gmake, the default file will be named GNUmakefile.

Just as GNU has its own C compiler, it also has its own make utility called gmake. Under Linux the make and gmake utilities are synonymous; however, other versions of UNIX have a different version of make. In other words, gmake is not part of the standard UNIX environment, but it can be added to any system.

Differences exist between GNU's gmake and the make utility available to other versions of UNIX. For instance, gmake will look for a configuration file called GNUmakefile. If gmake doesn't find a GNUmakefile, it will look for a Makefile. The make under UNIX, however, only considers Makefile. Also, gmake is a much more capable utility than the standard version of make. Either by employing the capabilities of gmake above make, or by choosing a configuration filename of GNUmakefile, you can impose use of gmake as a requirement for building your project. Because gmake is easily available from the Internet and is a superior make utility, I generally do require it.

When creating your make utility configuration file GNUmakefile, several areas must be addressed to construct it properly. Listing 1.1 contains the GNUmakefile that will be used to build the Graphics Editor project.

Examine the listing carefully, identifying the fields and components discussed during the coverage of the gcc command.

Listing 1.1 Graphics Editor Project GNUmakefile

```
1:
    # 2: # GNUmakefile for 2d-editor project
3:
   ##
4:
5: # Read in the system specific environment configuration
6: include ../make.defines
7:
8: PROGRAM = 2d \cdot gx
          = -1Xaw -1Xt -1X11 -1m
9: LIBS
10:
11: OBJS
            = gxMain.o \
12:
              gxGraphics.o \
13:
              gxLine.o \
14:
              gxText.o \
              qxArc.o \
15:
16:
              qxGx.o
```

Listing 1.1 Continued

```
17:18:
19: make-target: $(PROGRAM)
20:
21:
22: $(PROGRAM): $(OBJS)
        @echo "Building $(PROGRAM) for $(TARGET)..."
23:
        @(CC) -g -o $(PROGRAM) $(OBJS) $(X11LIB) $(LIBS)
24:
25:
        @echo "Done"
26:
27:
28: #
29: # end of GNUmakefile
30: #
```

Comments

As with any program that you write, including comments is the courteous thing to do. Whether for your fellow team member who must follow behind you, or for yourself when returning to the code six months later, comments can be critical in effecting something quickly because the clues they provide guide the analysis of what is required.

In the Makefile syntax, a pound sign (#) is the comment token. The make utility ignores anything following a pound sign on a line.

Variables

Variables provide a means of managing the content of your GNUmakefile by enabling you to group and define items that are later assembled into more complex entities. Further, make configuration files grow in size and complexity proportionately to a project. Some elements of the GNUmakefile are repeated for varying conditions and command assembly. The use of variables allows for a single definition of an element that can be used multiple times throughout the file. With the use of variables, modifying an element does not require that you search and replace its every occurrence. Instead, you only have to change the assignment to the appropriate variable.

In Listing 1.1, variables are identified by the equal sign that separates the variable name from the variable value. By convention, variable names within a Makefile are capitalized. Following this convention in your own make configuration files will help others decipher what you write.

Notice that the variables defined in Listing 1.1 can have a value consisting of a single string, as in

8: PROGRAM = 2d-editor

or they can have a value of a list of strings

9: LIBS = -1Xaw -1Xt -1X11 -1m

The make utility expects the carriage return/line feed entered when you press Enter to serve as the end-of-line marker. When assigning values to variables, this end-ofline marker implies end of input. Therefore, values of variables consisting of multiple strings must be viewed as if they were contained on a single line. In the case of the variables LIBS or CFLAGS in the listing, this isn't a problem. However, the value of OBJS is moderately large and would wrap to multiple lines if we allowed it.

Wrapping multiple lines is syntactically correct because you haven't pressed Enter (interpreted as end of input); however, it reduces readability and makes it more difficult to edit the field later. To indicate to the make utility that the items within the value of OBJS are on a single line, delimit the carriage return/line feed with a backslash $(\)$ character.

Delimiting instructs the make utility to ignore what immediately follows the backslash by not interpreting its meaning. The carriage return/line feed is still present, but it is not considered an end-of-line marker when preceded by the delimiter:

11:	OBJS	=	gxMain.o \
12:			gxGraphics.o \
13:			gxLine.o \
14:			gxText.o \
15:			gxArc.o \
16:			gxGx.o

You should now be comfortable with creating and assigning values to variables. To use a variable you've created, a unique syntax is necessary; this unique syntax allows the make utility to distinguish your intention clearly.

Employing a variable requires that it be prepended with the dollar sign (\$). In this way, the make utility knows that you want to reference the value of the variable and not the name of the variable. Also, because the variable may have a value with nested spaces, you must surround the variable using parentheses to indicate the entire value should be treated as an entity. A reference to

\$(LIBS)

is seen by the make utility as the value

-1Xaw -1Xt -X11 -1m

Targets

The use of targets within the make utility syntax enables you to have multiple entry points into the script. The power that this provides enables programmers to support multiple requirements from the same GNUmakefile.

For instance, compiling the project is a key function of the make utility and the reason you construct a GNUmakefile. Once the project is built, however, you may want to create a target within your GNUmakefile to install the project. You may want to move it from the user account level where only you (depending on permissions) can use it to a level where everyone on the system can benefit from it.

EXCURSION

You Know You're a Geek When...

The Towers of Hanoi is a classic programming problem within computer science used to demonstrate computing power and speed.

The legend is that there are monks in a cave somewhere with wooden discs stacked on one of four pegs. The discs decrease in size with the largest on the bottom and the smallest on top (see Figure 1.11). The monks must move the discs from a peg on one side to the peg on the other following these simple rules:

- Only one disc can be moved at a time
- A disc must always come to rest on a peg
- A larger disk can never rest on a smaller one



The telling of the legend ends by stating the number of discs that must be moved and identifying that once the task is complete, the world will end.

I once met someone who accomplished programming a solution to the Towers of Hanoi by using only the make utility syntax.

Two targets are defined in the GNUmakefile of Listing 1.1:

```
19: make-target: $(PROGRAM)
```

and

22: \$(PROGRAM): \$(OBJS)

1

Targets are identifiable by a label followed by a colon (:). The first target encountered by the make utility is considered the default target. When make is invoked, if no target is specified, it will begin at the first one it finds.

Issuing the command

bash[9]: gmake

causes the make utility to look for the file GNUmakefile and when found, to begin executing at the first target defined within the file.

Whereas, the command

bash [10]: gmake 2d-editor

causes the make utility to look for the target 2*d*-*editor* within the GNUmakefile and begin execution there.

Remember, the make utility sees \$(PROGRAM) as its value 2d-editor.

The make utility considers anything on the same line, following the target label, to be a *dependency* of the target. Because a dependency must be resolved first, make evaluates all dependencies before evaluating the body of the target. Based on what the dependency is the evaluation may vary.

For instance, in Listing 1.1, a dependency to \$(PROGRAM) exists for the target make-target.

```
19: make-target: $(PROGRAM)
```

When the make utility begins to evaluate \$(PROGRAM) as a dependency to make-target, it sees that \$(PROGRAM) has dependencies as well.

22: \$(PROGRAM): \$(OBJS)

The dependency declared for \$(PROGRAM) is \$(OBJS). Because \$(OBJS) is not a target defined within this GNUmakefile, the make utility uses built-in rules to evaluate \$(OBJS).

EXCURSION

Generating Object Files from the make Utility's Internal Rule Set

The built-in rule governing object files is in fact a target internal to the make utility. The target uses the base name of the object files as expanded from the **\$(OBJS)** variable. Effectively it says, in order to get a .o (*dot-oh*), look for a .c (*dot-see*) of the same *base name* and run gcc -c against it. If a .o already exists, the time stamp of the associated .c is compared and gcc -c is run only if the .c is newer than the existing .o. Whew, what a mouthful! A target *body* begins on the line immediately following the target label. Each line of the body is prefaced with at least one tab character. The body ends when make finds a line not beginning with a tab.



The use of tabs and spaces can be confusing to a new programmer. The places where make syntax allows a space or expects a tab varies, depending on which part of the make configuration file you are in. For instance, when writing a target body, a character that is not a tab as the first character on a line is seen as terminating the body. However, when assigning a value to a variable and delimiting the end-of-line marker so the value may span multiple lines, consecutive lines may begin either with a space or a tab.

In following the flow of make in Listing 1.1, you see that when make-target is called, the dependency \$(PROGRAM) is immediately evaluated. (As make-target does not have a body, you can assume that its sole purpose is to ensure that \$(PROGRAM) is evaluated.) When the make utility evaluates \$(PROGRAM), it sees that \$(PROGRAM) depends on \$(OBJS). The make utility then invokes its internal rule to create the files specified in the value of \$(OBJS). When the files in \$(OBJS) have been created (or updated), make is ready to evaluate the body of the \$(PROGRAM) target.

The first line of the \$(PROGRAM) target body is the echo command.

```
23: @echo "Building $(PROGRAM) for $(TARGET)..."
24: @(CC) -g -o $(PROGRAM) $(OBJS) $(X11LIB) $(LIBS)
25: @echo "Done"
26:
```

A standard UNIX command, echo simply prints everything that follows on the same line to the screen. Notice that in the \$(PROGRAM) target body, the echo command is preceded by the @ sign.

23: @echo "Building \$(PROGRAM) for \$(TARGET)..."

The @ symbol before a command tells make not to print the command to the screen before executing it. When used, the @ symbol forces make to be silent and simply run the command.

Continuing with the \$(PROGRAM) target body, line 24 instructs make how to assemble the cc command for the last phase of compiling called the link phase.

24: @(CC) -g -o \$(PROGRAM) \$(OBJS) \$(X11LIB) \$(LIBS)

Remember that the \$(OBJS) dependency to \$(PROGRAM) ensured that all the .o files were constructed or updated. The last step is to link all the object files together and resolve any external dependencies. In assembling the cc command, some variables are used that you haven't yet seen (CC, TARGET). To discuss them, we'll look at an internal make utility command called include.

include

With large-scale, professional-level development, source files will almost certainly be grouped by common purpose and located in separate directories from source files of differing purposes. Each directory will have its own make configuration file because the object file list (\$(OBJ)) will differ for each directory, as can the external dependencies or library considerations.

When faced with a project structure of many directories or multiple GNUmakefiles, it is not necessary to repeat variable declarations for each of the make files within your project. Elements common to the entire project can be placed at a central location and then included with the include command into each of the project's make files.



The term make file refers to GNUmakefile or Makefile (or any make configuration file). If you don't use one of the default make filenames (GNUmakefile or Makefile), make must be told the name of the file. This is done by specifying the -f flag:

```
bash[12]: gmake -f SomeMakeFileName
```

Refer to Figure 1.6, Graphics Editor project layout, and notice the file make.defines in the 2d-editor directory. This is the same make.defines file included by the GNUmakefile in Listing 1.1.

```
5: # Read in the system specific environment configuration
6:
  include ../make.defines
```

It is the common definition file for variables and functions global to the entire project.

When you specify a file for inclusion into a make file, the make utility inserts it exactly as if it were typed in at the place of the include command. There is no limit to the number of files that can be included into a make file, nor is there a limit on the number of times a file can be included.

make.defines

Analyzing the make.defines file shown in Listing 1.2 gives you a lot to consider. However, as we break it into parts, you'll see how easy it is to understand. Remembering previous discussions of the gcc command will help identify components of the make.defines file that satisfy the required elements of gcc.

Listing 1.2 The make.defines for the Graphics Editor Project

```
###
1:
2: # make.defines
3: #
4: # Included in each make file used with the 2d Editor Project
5: #
6: # Dependencies on the following environment variables:
7: # TARGET = machine-os
8: #
9: # The syntax of this file is for use
9a: # with 'gmake' (GNU version of make)
10: ###
11: TARGET SPARC SUNOS = sun4u-SunOS
12: TARGET i86 SOLARIS = i86pc-SolarisOS
13: TARGET i86 LINUX
                       = i86-Linux
14:
15: ifdef GxHOME
16:
      GxSRCDIR = ${GxHOME}/src
17: else
18:
      GxSRCDIR = .../src
19: endif
20:
21: vpath %.h ${GxSRCDIR}/include
22: vpath %.c ${GxSRCDIR}
23:
24: #
25: # Configure for Linux running on a PC
26: #
27: ifeq ($(TARGET),$(TARGET i86 LINUX))
28:
       X11INC
               = -I/usr/include/X11
29:
       X11LIB
                 = -L/usr/X11R6/lib
30:
      INCS
                 = -I${GxSRCDIR}/include
31:
       CC
32:
                 = gcc
33:
       OPTS
                 = -ansi -Wall -g
34: endif
35:
36: #
37: # Configure for Solaris running on a Sparc
38: #
39: ifeq ($(TARGET),$(TARGET SPARC SUNOS))
40:
       X11INC
               = -I/usr/openwin/include
41:
       X11LIB
                 = -L/usr/openwin/lib
42:
       INCS
                 = -I${GxSRCDIR}/include
43:
44:
       CC
                 = gcc
45:
       OPTS
                 = -g -Wall -ansi
46: endif
47:
48: #
49: # Configure for Solaris running on a PC
```

1

Listing 1.2 Continued

```
50: #
51: ifeq ($(TARGET),$(TARGET_i86_SOLARIS))
52:
       X11INC = -I/usr/openwin/include
53:
       X11LIB = -L/usr/openwin/lib
       INCS = -I${GxSRCDIR}/include
54:
55:
56:
       CC
               = gcc
       OPTS
               = -ansi -Wall -g
57:
58: endif
59:
60: #
61: # Force all Makefiles using this file to check the configuration
62: # of the environment before building the target
63: #
64: all: make-env-check make-target
65:
66: #
67: # Check to environment variables need to build are set
68: #
69: make-env-check:
70: ifndef TARGET
71:
          @echo
          @echo "TARGET not defined!"
72:
         @echo "Set environment variable TARGET to:"
73:
74:
         @echo " sun4u-Sun0s"
         @echo " i86pc-SunØs"
75:
         @echo " i86-Linux"
76:
77:
          @echo
78:
           @exit 1
79:
       endif
80:
81: clean:
82:
       @rm -f *~ *.o $(PROGRAM)
83: #
84: # end of make.defines
85: #
```

Identifying the pound sign (#) as the comment token in make file syntax enables us to skip down to line 11 of Listing 1.2 to begin the discussion.

11: TARGET_SPARC_SUNOS = sun4u-SunOS
12: TARGET_i86_SOLARIS = i86pc-SolarisOS
13: TARGET_i86_LINUX = i86-Linux

Beyond the comments, the file begins with a series of variable declarations. Because you should now be comfortable with declaring variables and assigning them values, this is review. The variables TARGET_SPARC_SUNOS, TARGET_i86_SOLARIS, and TARGET_i86_LINUX are an anticipation of the platform and operating system combinations supported in this project. As you look through the make.defines file, you see that each of these variables is compared to another variable called TARGET. Consider the following lines:

```
27: ifeq ($(TARGET),$(TARGET_i86_LINUX))
```

```
39: ifeq ($(TARGET),$(TARGET_SPARC_SUNOS))
```

```
51: ifeq ($(TARGET),$(TARGET_i86_SOLARIS))
```

This is consistent with the comment at the beginning of the make.defines file:

```
6: # Dependencies on the following environment variables:
```

```
7: # TARGET = machine-os
```

The environment variable TARGET must be set for the make.defines file to work. Fortunately, make file syntax enables a test of this dependency.

EXCURSION

Promoting a Variable to an Environment Variable

An *environment variable* is much like a variable set within a make file. However, the variables set within a make file are visible only to the make utility. The issue of visibility is known as *scope* and will be discussed at length in the next chapter. In the meantime, any of the variables that you've defined in the GNUmakefile can be set in your *environment*. If removed from the make file and placed in your environment, the variable is available (visible) to every command and utility and is then called an environment variable.

Some environment variables are provided by the operating system. For instance, to see what command shell you are running, echo the environment variable SHELL:

bash[13]: echo \$SHELL

To see all environment variables currently set, use the env command. Notice the environment variable path in the env output. The path variable informs the command shell of all the places to search for external commands.

→ (See Excursion "What Is a Command Shell" under section on "The mkdir Command" for a description of internal verses external commands, page 11.

We don't want just any variable set as an environment variable, so choose the variables carefully in order to not clutter the environment.

Knowing which command shell you are currently using will enable you to use the correct syntax for setting an environment variable called **TARGET**. Table 1.3 is a listing of these shells and syntax.

lan	able 1.3 Environment variable Syntax				
	Shells	Syntax			
	bash, sh, and ksh	export TARGET=i86-Linux			
	csh and tcsh	setenv TARGET i86-Linux			

Table 1.3 Environment Variable Syntax

As discussed previously, the first target (a label followed by a colon) which the make utility encounters is the default; it is executed if no target is specified on the command line.

Notice from Listing 1.1 that the include command is the first line (beyond comments) of the GNUmakefile. Therefore, any target found in make.defines precedes any target found in the GNUmakefile. As you look through make.defines, notice the target called all.

64: all: make-env-check make-target

This target has two dependencies that are also targets. The first to be evaluated is the make-env-check target, which ensures that the TARGET environment variable is set. It does this with the keyword ifndef, which stands for *if not defined*. The ifndef must be followed by a variable name so that the make utility knows what to test for definition. Because the ifndef implies a conditional body (what to do if not defined), the body must have an end. The ifndef end-of-body marker is the keyword endif.



To create conditional bodies in the case that a variable *is* defined, use the ifdef keyword. It is used exactly like the ifndef keyword, except that the body is acted on when the variable exists in the environment instead of when it doesn't.

If the make-env-check completes successfully (meaning the TARGET variable is defined), the make utility continues on to evaluate the make-target dependency of the all: target. The make-target, as seen in Listing 1.1, is contained in the GNUmakefile. The target's sole purpose is to ensure that the \$(PROGRAM) target executes.

Surprisingly, we are already through a good portion of the make.defines file. Having discussed the initial variables, the target all:, and the dependencies make-env-check and make-target, we are ready to look at the test of GxHOME.

```
15: ifdef GxHOME
16: GxSRCDIR = ${GxHOME}/src
17: else
18: GxSRCDIR = ../src
19: endif
```

The GxHOME variable ensures that the GxSRCDIR variable is set correctly. The GxSRCDIR variable enables us to store the source files and object files in different places. This flexibility is crucial in multi-platform development because every platform will need to generate private copies of the object and executable files.



If necessary, review the section "Object Files" for a description of object file incompatibility between platforms, page 19.

Refer to Figure 1.6 where the i86-Linux directory for storing object files is separate from the src directory. Similarly, you could have an independent directory for every platform or operating system you intend to support, as illustrated in Figure 1.12.



EXCURSION

Building a Project for Multiple Architecture

The make file structure we've evaluated for the Graphics Editor project will support the platforms shown in Figure 1.12. Through the use of the TARGET environment variable and its successful comparison to one of the environments defined in the make.defines file

11:	TARGET_SPARC_SUNOS	= sun4u-SunOS
12:	TARGET_i86_SOLARIS	= i86pc-SunOS
13:	TARGET_i86_LINUX	= i86-Linux

the project could be built successfully for any of these.

To add platform support for other environments, an additional TARGET_MACHINE_OS variable could be added to make.defines with a corresponding ifeq body to set the necessary variables to the values correct for the new platform.

The separation of object files from source files is possible through the use of the GxSRCDIR variable. After the location of the source files is determined, the variable vpath can be set for the .h and .c files of the project.

The vpath variable in the make.defines works much like the path variable in your environment. Just as the path variable instructs the command shell where to search

for commands, the vpath variable tells the make utility where to search for source files.

→ See Excursion "Promoting a Variable to an Environment Variable" for a description of the path variable, page 40.

The vpath variable informs the make utility where to look when it needs to find a .c (represented as %.c) or a .h (%.h) file.

21: vpath %.h \${GxSRCDIR}/include 22: vpath %.c \${GxSRCDIR}

This way, the .c and .h files that make needs do not have to be present in the same directory where the make command is executed.

By creating a directory of the same name as the TARGET value, you can issue the make command from within this directory having the results stored in this platformspecific directory separated from the source code.

One last detail necessary to make this separation of object files and source files work is the fact that the GNUmakefile must be visible in each of the object directories. It is the GNUmakefile and its inclusion of make.defines that establishes the necessary targets, defines the \$(OBJS) file list, and assigns the pertinent vpath values.

To accomplish having the GNUmakefile visible to each of the object directories, you could use the cp (copy) command and place a copy of the GNUmakefile in each directory needing it. But what happens when you need to modify the GNUmakefile? You would either have to repeat the modification in every place where you have a copy of the file (supposing you could remember them all) or you'd have to re-copy the file to every directory.

In UNIX there is a way to make a single file visible in multiple places. Changing the file at any place where it is visible is the same as changing it at its actual location. The mechanism for making this possible is called a *symbolic link*.

The In -s Command

A symbolic link can be a pointer to a file or a directory. The power of symbolic links is that they are treated exactly like what they point to. In the case of files, you can edit them with the changes getting stored in the actual file.

To create a symbolic link, use the following command:

```
bash[14]: ln -s /home/jbrown/2d-editor/src/GNUmakefile \
/home/jbrown/2d-editor/i86-Linux/
```

This places a pointer to GNUmakefile in the i86-Linux directory. You can then change (cd) to the i86-Linux directory and execute the gmake command. The command will find the GNUmakefile as if it were located there.



You can use the link command (1n) against directories as well as files. The syntax for the command supports relative or explicit paths. The basic syntax for creating a symbolic link follows the form

```
ln -s what where
```

Specifying what you want to link (*what*) is always necessary. However, if the where is the current directory maintaining the same name as *what*, you can omit the where designator. As always, see the man page for a complete description of the ln command.

You've now completed a close look at the elements required to create, edit, and build a project under the Linux operating system. However, other commands and utilities are available under Linux that are not required but will give you an advantage when accomplishing your development tasks.

System Tools and Useful Commands

Linux is a complex operating system that puts literally hundreds of commands at your disposal. Learning them all will take time, investigation, and practice. Just as a musician always feels someone is more accomplished, you'll never feel like you've learned everything. If ever you feel you know it all, you are probably consulting.

The following section discusses some of the more common Linux commands used in software development.

grep, Pipes, Redirection, and more

The grep command searches files for occurrences of regular expressions. Following this basic syntax forms the command

bash[15]: grep [flags] string files(s)

When a match is found, grep returns (prints to the screen) the entire line that contained the match. Use of the -n flag helps in reading the output because it instructs grep to include the line numbers where the matches were found.

As mentioned earlier, UNIX is case sensitive. This sensitivity extends to the grep command as well. However, unlike UNIX you may instruct grep to suppress this and perform caseless searches. Passing a -i flag to grep instructs that case should be ignored.

A common command used in a development environment is

bash[16]: grep -i something *.[ch]

that requests grep to perform a caseless search (-i) on all .c and .h files for something.

This command is also useful when using grep is filtering out what you don't want to see. This can be done with an *inverse search* (-v flag) in which you tell grep what not to print, or in other words, what to ignore.

An example of when an inverse search would be useful is in searching for the string man where grep returns multiple lines from the input that includes the string manual. To instruct grep not to return the lines containing manual, you would issue the following command:

bash[17]: grep man myfile.txt | grep -v manual

Very important to this example is the symbol linking the output of the first grep command to the input of the second. This symbol is called a *pipe* (|), and it is an extremely powerful feature of UNIX because it provides a way of chaining commands together.

The result of the preceding example is that grep searches *myfile.txt* for every occurrence of the string *man.* As I pointed out, the output also contained multiple lines with the string *manual*, which was undesirable. So the output of the first grep is made the input of the second grep by use of the pipe, allowing an inverse search of the string *manual* on the output of the first command. The inverse search removes any line from output of the first grep containing the string *manual*. What you are left with (hopefully) following the second grep are only occurrences of the string *man.*

Now that you're comfortable with the use of the grep command, let's focus a moment longer on the pipe feature of UNIX.

The capability to take the output of one command and make it the input of another is called *redirection*. Through redirection, the pipe symbol (|) can chain together two commands, as seen in the preceding example.

A very useful application of redirection is the more command. When the output of a command or file is too large for the window, you can ask UNIX to break it into window-sized pieces.

```
bash[19]: ls -l | more
or
bash[20]: more myfile.c
```

It is also possible to redirect the output of a command to a file. The > (greater than) symbol works much like a pipe, but instead of sending the output of one command to another, it sends the output to a file. If the file doesn't exist, it will be created.

bash[18]: grep man myfile.txt > grep.output

If the file already exists, its contents will be overwritten with the results of this command. To ensure that nothing is lost, you could redirect the output of grep so that it appends the file by using >> (two greater than signs):

bash[19]: grep man myfile.txt >> grep.output

A common way that developers use the power of redirection is to save the output of the gmake command so that reported errors are recorded in a file:

bash[20]: gmake > make.out

The find Command

The find command allows for searching a directory structure for just about anything. With the find command, you can search for a file or directory of a specific name, a file of a certain extension, a file or directory of specific permissions, or a given modification date.

The more powerful a command, the more complex its assembly and execution, as is true with the find command.

The basic structure of the find command follows the syntax

find starting point condition value action

From *starting point*, the find command searches through the entire directory tree for a *condition* of some *value*, and when satisfied it executes the *action*.

The find command requires that you provide a *starting point* that is a valid directory in your file system.

The conditions which find understands are numerous. The simplest is -name. As given in the command syntax, a condition must have a value. The -name needs the name of a file or directory to search for:

```
find /home -name readme.txt ...
```

Wildcards can be used to specify the *value* for the name condition; however, when using wildcards the *value* clause must be contained in double quotes ("). Otherwise, the command shell will interpret the wildcards, and the find command will never see them.

find /home -name "*read*" ...

Knowing that find, like all of UNIX, is case sensitive, you must explicitly assign the value string to the characters you want find:

find /home -name "*[Rr][Ee][Aa][Dd]*" ...

This would find *README* as well as *readme* or any combination of the two (such as *ReaDme*).

Having properly formed the *starting point* for the find command, the *condition* and *value* to search for, the only remaining element is instructing find what to do when the search is successful.

Current UNIX versions will automatically perform the *-print* action if no other action is specified.

bash[20]: find /home -name "*[Rr][Ee][Aa][Dd]*" -print

The find command understands several other actions. One very useful for software developers is the -exec command. This instructs find to execute a command against the matches found. Clearly, in conjunction with -exec, you must tell find what command to execute.

find /home -name "*.[ch]" -exec grep MyFunc {} \;

Looking closely at this example, you see everything discussed.

This illustrates the find command with a *starting point* of /home, a *condition* of -name, a condition *value* of "*.[ch]" (all .c and .h files), an *action* of -exec (execute a command), and the command of choice being grep. Finally, grep is searching the lines returned by find for any occurrence of the string *MyFunc*.

To represent the matches returned by find to the grep command, use {} (open and close curly braces). It is also necessary to end the command portion of -exec, which is done with a semicolon (;). However, you must hide the semicolon from the command shell by using the delimiter I spoke of earlier. This is necessary because the semicolon has meaning to the shell as well as the find command but is meant to inform find that the command portion of -exec is finished.

Next Steps

With a few Linux commands under your belt, an understanding of the compiling and linking processes, and an introduction to the vi editor for creating source files, you are ready to begin looking at programming constructs. Chapter 2, "Programming Constructs," will lead you through structuring solutions to software problems by applying common programming constructs and logic processes.

Chapter 2

In this chapter

- Decisions
- Loops
- Functions
- Data
- Next Steps

Programming Constructs

Learning to properly define the problem to be solved in a software application is an unavoidable first step of computer programming. The second step, applying the correct construct to effect the solution, lies at the heart of software engineering.

Identifying and solving a problem at this level is different from the role of a systems analyst in the software development process. Systems analysts are concerned with the design method, feature list, and milestones of an application.

As a software engineer, you are continually challenged with accomplishing the next portion of the application. Perhaps the task is to read a configuration file and initialize a variable set accordingly. Instantly, you should begin thinking about loops, decisions, functions, and data.

This chapter focuses on modeling problems by the programming construct most often employed to solve them. Learning syntax and convention is required to add a new computer language to your skill set; however, programming constructs are constant throughout computer programming. As the boss likes to say, "No one is inventing a new bubble sort." This means, of course, that there is an appropriate way (construct) for solving specific problems when writing software. Let us review those now.
EXCURSION

Do People Really Sort Bubbles?

A *bubble sort* is a method of ordering data with neighboring data elements in a list compared as the list is traversed. If the element closer to the beginning of the list is greater than its neighbor, the elements are swapped. After multiple iteration, the smaller values bubble to the top (beginning of the list). The list is sorted when nothing is swapped during a traversal of the list.

A *programming construct* is the basic element used in computer problem solving. It is a method, practice, or means of accomplishing a task or representing data.

Knowing when (and how) to apply a construct properly is the challenge of learning computer programming. You add new languages to your abilities by determining how to accomplish decisions, looping, or function calls using this new language.

The following sections focus on creating a foundation of basic programming constructs. If you are already comfortable with programming constructs, feel free to continue to the next chapter.

Decisions

A decision tree begins with a question where the answer may present further questions or an immediate end to the tree.

EXCURSION

Do You Have Apple Pie?

"Great, I'd like apple pie a la mode with vanilla ice cream, and I'd like the pie heated. If you don't have vanilla, I'd like whipped cream but only if it is fresh. If it is not fresh, then nothing. Only the pie, but then not heated."

("When Harry Met Sally," Metro Goldwyn Mayer, 1989)

Figure 2.1 is an example of how to approach decisions, as each part of the process has significant impact on what happens next. Failure on the part of the programmer to account for a choice or condition required by the user can result in the application having a short life span.

Decisions create execution branches, control loop conditions, and dictate overall program flow. With the advent of object-oriented and event-driven programming methods, the model of purely sequential program execution was replaced. New methodologies complicate decision trees, because applications must be resilient enough to account for any circumstance at any moment.



Certainly, there is more to applying methodologies than simply making a decision. However, the decision construct controls the state of an application, which in turn determines when actions are valid.

EXCURSION

Creating a State Machine from a Decision Tree

Referring again to Figure 2.1, each *yes* branch could be labeled with a number to indicate the tree's state. Clearly, the action of heating the pie is only valid if we reach the state where it is allowed. This is a function of the decision construct.

Because the Graphics Editor project is X Window-based, event-driven application development will be looked at closely in Chapter 4, "Windowing Concepts." The decision construct is crucial to managing the state of an application, as it controls how a program unfolds and determines when actions are allowed and, often, what action is required.

As illustrated in Figure 2.1, a basic decision consists of a true or false condition. Does the apple pie exist? Is the ice cream vanilla? Is the whipped cream fresh? Each condition is tested and program execution follows one of two branches representing a true action or a false action.

No matter the complexity of the decision tree or means of representing the decision process within the programming language, it can be reduced to this level of simplicity.

The if Statement

The if statement represents a basic decision. The syntax of if follows the pattern

```
if condition action
```

The if statement always performs the *action* based on an evaluation of the terms of the *condition* resulting in true.



The evaluation of a condition in the C programming language results in true for any non-zero value. A condition is considered false only if it equals zero.

You could be more precise by explicitly testing for the true.

```
if condition equals true action
```

Notice that when the condition being tested is negated, if still takes the action based on successful evaluation of the terms of the *condition*:

```
if condition equals false action
```

In other words, when it is true that the condition is false, if performs the action.

EXCURSION

A Stylistic Note...

Programmers commonly put the negative condition before the positive, directing to test for the negated or *condition equals false* before testing for the true. Although semantically and syntactically correct, it is less readable for anyone who might follow behind.

The else Statement

As demonstrated, if executes the *action* based on a true evaluation of the terms of a *condition*. To branch to an action based on an implied false evaluation, use the else statement.

The else statement must be preceded by an if and immediately follow the *action* of the if:

```
if condition
action
else
another action
```

Types of Conditions

Let us focus for a moment on what is referred to as the *condition* being evaluated by the if statement. Because conditions depend upon the data being evaluated, it will be necessary to look at each type separately.

Numbers

When forming test conditions for decisions, a variable representing a number can be evaluated in many ways. For instance, evaluation can determine if the number is equal to zero:

if (number == 0) action



Notice that I've grouped the condition using parentheses. This representation is more readable and helps you to start thinking in terms of the C language syntax. Remember, the focus here is on the constructs; representing them using C syntax will begin the introduction of Chapter 3, "A Word on C," where the focus is the C programming language.

A variable representing a number can also be tested for being less than or greater than zero:

```
if( number < 0 )
action
```

or

```
if( number > 0 )
action
```

Further, it can be tested for being within a range:

```
if( number < 20 && number > 10 )
    action
```

This example tests to see whether number is less than 20 and greater than 10.

Table 2.1 shows the symbols understood by the C language for forming test conditions on numbers. Again, C syntax is not the current focus but is useful because a means of representing the conditions is needed for our current discussion.

	Symbol	Meaning	Example			
	>	greater than	if(variable > value)			
	<	less than	if(variable < value)			
	>=	greater than or equal to	if(variable >= value)			
	<=	less than or equal to	if(variable <= value)			
	==	equal to	if(variable == value)			
	!=	not equal to	if(variable != value			

Table 2.1 Symbols for Numeric Conditions

Т

able 2.1 Continued							
	Symbol	Meaning	Example				
	&&	and	if(condition && condition)				
	II	or	if(condition condition)				
_	!	not	if(!condition) equivalent to if(condition == false)				

Characters

Conditions created using variables representing characters follow the same rules and employ the same symbols as numbers. This is true because, as we discussed in Chapter 1 (see the section "The C Compiler," page 18), the computer does not understand the letter *A*. The representation of a letter in a manner that the computer can understand is as a number.

The letter A, as seen in Table 1.2, is the same as the number 65 when it is compiled into machine language, allowing the same conventions used with numbers to be applied to characters when forming test conditions.

To illustrate this, consider the following conditions:

```
if( character > 'A' )
or
if( character < 'Z' )
or
if( character > 'A' && character < 'Z' )</pre>
```



Treating characters like numbers requires you to know the proper order in which the language places them. This ensures that the semantics of a condition are as intended.

Because the letter *A* is a decimal 65, *Z* is 65+25 or 90 because the letter *Z* is 25 places away from the letter *A*. But where do *a* through *z* fit in? Is *a* less than or greater than *A*? Also, what about other printable characters that are not letters, such as punctuation?

A standard known as *ASCII* defines the representation of characters as numbers for manipulation by computers.

To see the entire ASCII table of characters and where all printable (and nonprintable) characters fit in to it, use the man command:

bash[20]: man ascii

Figure 2.2 shows a portion of the man page for the ASCII table.

EXCURSION

ASCII (American Standard Code for Information Interchange)

ASCII is the most common format for text files in computers and on the Internet.

In an ASCII file, each alphabetic, numeric, or special character is represented with a 7-bit binary number (a string of seven zeros or ones). ASCII defines 128 possible characters.

UNIX- and DOS-based operating systems (except for Windows NT) use ASCII for text files. Windows NT uses a newer code, *Unicode*. IBM's System 390 servers use a proprietary 8-bit code called EBCDIC.

Conversion programs enable different operating systems to change a file from one code to another. The American National Standards Institute (ANSI) developed ASCII.



Strings

Strings differ from characters in that they are *groupings* of characters. Although this might seem too obvious to point out, manipulation of strings requires a completely new heuristic.

The ASCII table does not apply to groups of characters. If the test condition considers the individual characters comprising a string, it is okay to apply what you've learned. However, when evaluating the group of characters as an entity, you must use other methods.

I∎ **⊂** Note

Strings do not employ the same symbols for comparison as numbers or characters. To do so requires that a decimal value be acquired for a string; however, very different strings could result in the same numeric value. For instance, when adding the ASCII values for each of the letters in the following strings, the same value results:

FIG = 70 + 73 + 71 = 214and SAB = 83 + 65 + 66 = 214

The means or mechanisms used to evaluate strings are specific to a computer language. In the C programming language, an entire library is dedicated to string manipulation and testing. The C string library is covered in Chapter 3; however, here we need an understanding of test conditions for strings and specifically the concept of *lexical analysis*.

The lexical analysis of strings results in one string being evaluated as less than, greater than, equal to, or not equal to another.

Everyone, at some point, has used a dictionary to look up a word he didn't know how to spell; we are all comfortable with the idea that strings are ordered. The lexical analysis of two strings seeks to determine whether one string comes before or after another in the dictionary. If one string precedes another, it is said to be less than the second string. If a string appears after another in the dictionary, it is considered greater than. The concepts of strings being equal or not equal should be immediately apparent.

As mentioned, string manipulation is language specific; therefore, examples of how to form test conditions for strings are detailed in Chapter 3.

With a clearer understanding of test conditions for different data types, you are ready to form the construct to support multiple test conditions.

The if else Statement

So far, we've only considered a condition of true (if) and an implied false (else).

What if the condition you are testing has one of many possible values? For instance, what if the user has selected an option from a menu of several choices, as shown in Figure 2.3?



Testing the user response requires an equal number of tests as choices.

To test for all the possible inputs, you could use a series of if statements:

```
if( input == '1' )
  open action
if( input == '2' )
  close action
if( input == '3' )
  delete action
if( input == '4' )
  rename action
if( input == '5' )
  print action
```

The shortcoming of this approach, however, speaks not only to performance but also to the lack of a default condition in the case where the user misses the number 5 key and presses 6.

Simply adding an else to the list as in

```
if( input == '1' )
    open action
if( input == '2' )
    close action
if( input == '3' )
    delete action
if( input == '4' )
    rename action
```

```
if( input == '5' )
  print action
else
  error action
```

is not valid because the *error action* is taken not only when the user enters an invalid entry, but also when he enters any valid response other than 5. This, of course, ignores the fact that every *if* statement is evaluated regardless of the input or when a match is found.

Use of the else if statement resolves both problems:

```
if( input == '1' )
   open action
else if( input == '2' )
   close action
else if( input == '3' )
   delete action
else if( input == '4' )
   rename action
else if( input == '5' )
   print action
else
   error action
```

After an evaluation results in true, no further if statements are considered. Also, when no match to input is found, then and only then is the *error action* performed.

Notice in the if else example, placement of the if else statement follows rules similar to use of the else statement. The first use of an if else must be preceded by an if statement and must immediately follow the action associated with the if. Following the first use of an if else, further if else statements may follow in the chain immediately after the action of a preceding if else.

The if statement always starts a new chain and must always be first. The else statement always ends the chain; therefore, it must be the last statement in the series (chain) of if, if else statements.

The if else statement is well suited for evaluating conditions with potentially many values. However, continuing with the discussion of decision constructs, the case statement provides an alternate solution to this problem.

The case Statement

Another way to solve the problem of evaluating the input shown in Figure 2.3 is to employ a case statement. The case statement provides a graceful and flexible alternative to the if else for this type of problem.

A case statement operates by switching on the value of a numeric or character variable for each of the conditions you want to account for:

```
switch( input ) {
  case '1':
    open action
   break
  case '2':
    close action
   break
  case '3':
    delete action
   break
  case '4':
    rename action
   break
  case '5':
    print action
   break
  default:
    error action
}
```

For each of the conditions in the switch, the case statement equates to an entry point for the condition and the break defines where to exit the structure. The flexibility this provides is that case statements can be grouped to share actions.



Consider again the if else solution to the menu problem. If you decided to accept either a 1, 0, or o as valid input to invoke the *open action* you would have to expand the test of if (*input* == '1') to a significantly more complex condition.

The flexibility of the case statement requires only that we add a condition (entry point) for the new input:

```
case '0':
case '0':
case 1:
open action
break
```

The case statement provides a decision construct that remains clear, succinct, and easily maintained.

Before leaving the case statement, notice the default condition in the previous examples. As expected, this is the condition executed when no other match is found.

Loops

Whether you are trying to find a match to an item in a list, count the number of customers in a database, or endlessly display a menu soliciting user input, doing things multiple times is often necessary in a computer program. A construct to make repetition possible is the *loop*.

Four ways to perform looping are available, and which one to choose does, of course, depend on the problem being solved.

The for Loop

Faced with the problem of having to look through a fixed length *array* for an empty slot, you should instantly consider the for loop in the solution.

EXCURSION

An Array Does Not Live in the Ocean

An *array* is a storage mechanism best thought of as a filing cabinet. A filing cabinet has a fixed number of drawers after it is created. Each *drawer* of an array is a storage location called an *element*, and the number of elements make up the array's *length*.

Like the number of drawers in a cabinet, the number of elements in an array cannot be changed after creation. Indexing into an array always begins with zero: array[0] is the first element of any array. Therefore, if an array has four elements, array[0] is the first and array[3] is the fourth and last element.

how too pro nouns' it

Indexing into array[0] is spoken as if written *array sub zero*. Following this pattern, array[1] is *array sub one*, and so on.

Although there are several ways to construct a for loop, it generally expects that things will be done a finite number of times. The layout of the for loop looks like the following:

```
for ( initialize - test - increment ) {
   action
}
```

EXCURSION

How to Define Bodies Using C Syntax

As seen with the switch statement, C syntax uses the curly braces ({}) as a means of grouping lines of code into a body. Syntactically, they are not required with the for loop as they were with the switch statement. However, without the start and end body markers, the for action could be only a single line of code.

Consider again the if statement where an *action* was associated with the true and false conditions. That was an easy way to conceptualize that the successful evaluation of the test results in something being done. The same could be said for looping, as the following shows:

```
for( initialize -- test -- increment )
   take action
```

However, the *action* taken may not be a single line of code but rather a series of lines.

If the *action* spans multiple lines, you must identify that they are *all* conditional, either conditional on a test associated with an *if* or conditional on the number of times you want to perform a loop.

Placing the conditional lines within a code body allows for the necessary grouping to identify them as a single conditional action.

To review, the C programming language uses the { as the start of a body of code (begin body marker) and the opposing } as the end of the body of code.

Consider again the for loop with the correct C syntax:

```
for( initialize; test; increment ) {
   some action
}
```

Notice the correct syntax is to have the initialize, test, and increment fields of the for loop separated by a semicolon (;).

Having some variable, you set its initial value in the initialize field of the for loop. In the next field, you test it to see whether it has reached some limit or satisfied a condition that will stop the looping. Finally, you increment the variable so that it approaches the value needed to satisfy the test:

```
for( i = 0; i < 10; i++ ) {
    some action
}</pre>
```



In the previous example, if you omit the increment field of the for loop, you never move the value of i closer to the condition of i equal to 10, which ends the loop. If this happens, i always equals zero, which is *forever* less than 10. This is called an *endless loop* because the loop never ends.

Optionally, you can place the increment field in the body of the for loop if you leave a placeholder:

```
for( i = 0; i < 10; ) {
    i++;
some action
}</pre>
```

Assuming the length of an array to be searched is 10 (meaning it has 10 elements), the for loop in the following example is ideal for the task:

```
for( i = 0; i < 10; i++ ) {
    if( item == array[i] ) {
        item found action
    }
}</pre>
```

Of course, the test field of the for loop could be adjusted according to the length of any array.

The while Loop

A while loop is generally employed when you don't know the number of times the loop will need to execute. An instance when a while loop is most likely to be employed is finding the end of a list of unknown length, as in the following:

```
while( condition ) some action
```

or more specifically

```
while( !found ) {
    ...
}
```

how too pro nouns' it

The statement reads while not found.



Referring to Table 2.1, the *not symbol* (!) literally inverts the value of the variable. In order for the loop to execute, the value of found must initially be set to false as not false is true. As long as the condition of the while is true, the loop continues to execute.

The condition to end the loop must be created within the body of the loop to avoid the endless looping discussed earlier. For example,

```
while( !found ) {
   found = (item == list_entry );
   ...
}
```

In the previous example, the value of the variable found is assigned the result of the test ensuring that when item is found, the loop ends.

This would be equivalent to

```
while( !found ) {
    if(item == list_entry ) {
      found = True;
    }
    ...
}
```

Note the ..., signifying that some critical elements of this fragment are missing to make this example fully functional. Specifically, the code necessary for traversing the list of unknown length is not shown. However, as this type of list management is used heavily to parse the objects within the Graphics Editor, I will visit it again in more detail in Chapter 6, "Components of an X Window Application."

The do while Loop

Very similar to the while loop is the do while loop. It differs in that the test for determining whether you continue executing the loop comes at the end of the loop instead of the beginning. This ensures that the loop is performed at least one time.

Borrowing from the menu example from Figure 2.3, consider that when displaying a menu for user input the menu must be displayed at least once. Further, it continually displays until the user enters some input to signify his intention to exit the program. This is exactly the problem for which the do while loop should be used.

```
do {
    print menu action
    get input action
} while( input != EXIT );
```

While the input from the user is not equal to the exit condition, continue displaying the menu.

Certainly the body of the do while wants to employ the case statement for performing the actions consistent with the user's selection.

Having thus far only generalized about the actions associated with decisions and loops, constructs are required to support branching program execution for valid actions. These constructs, which make branching possible, are called *functions*.

Functions

For the same reasons that you don't store all your files in a single directory, you don't put all your source code into a single routine. This amounts to poor organization, which makes program maintenance, advancement, and support extremely difficult. Further, without grouping code into functions, sharing functionality within a project requires typing it in again and again.

Functions, also known as *modules*, are the constructs that enable you to organize and group your source code based on the tasks it performs. The more precise the task, the more concise the code, the easier it is to maintain, and the greater chance you'll be able to reuse it elsewhere in your project.

Within a project, performing a task more than once should indicate the need to write a function. For instance, the requirement to open files is common to many programs. Instead of placing the code necessary to accomplish this in multiple places within your project, it is appropriate to add a function that accepts a filename, determines whether the file exists, opens the file, performs the necessary test for success, and, finally, returns the handle acquired by the open.

Many aspects are critical to understanding and using functions: declarations, return types, parameters, and definitions. After we've discussed each of these, you'll be prepared to look more closely at the C programming language.

Declarations

To use a function, whether it's one that you've written or one that is provided by the language or environment in which you are working, it must be prefaced with a declaration.

Declaring a function enables the compiler to validate it by looking for obvious violations of syntax, and perhaps warning about semantic errors.

Syntax violations checked by the compiler include ensuring that the proper numbers of arguments (parameters) are passed at invocation. Examples of parameter usage are provided in the section "Parameters."

If a function expects two arguments and you've only specified one, serious and potentially fatal results can occur. This is not fatal to the programmer, luckily, but it could very well cause the program to crash.



The incorrect number of parameters specified when calling a function is a syntax error that causes the compiler to stop and insist that you fix it. Detection of this syntax error, however, is only possible if the compiler knows the number of parameters to expect for a function. This information is provided by the function declaration.

Semantic errors considered by the compiler include comparing the data type of items passed to those required. The type of the data implies an associated size, as discussed in the section "Data Types," later in this chapter. A mismatch in the type of data passed and expected can cause data to be lost or corrupted because size differences force the data to be expanded or truncated to satisfy the expected size.



Type checking compares what the function expects to what is passed. For instance, if the function requires a number (integer), the compiler ensures that you passed a number.

More detail on data types is provided later in this chapter. However, it is important to know that a function's declaration ensures that the function is used properly. Without the function declaration, the compiler must make assumptions about the function, and the compiler always assumes anything unknown is an integer because this is the default type in the C programming language.

Two ways to declare a function for use in your source code are by *prototyping* the function and by *defining* the function.

When a function is defined, it satisfies as a declaration because the compiler learns everything it needs to by the function's definition. If a function definition appears before its first invocation, there need not be a separate declaration.



A function's *definition* is, in fact, the function, and *a prototype* is the declaration of a function before its definition in order to provide the compiler with the following necessary information:

<return data type> Function Name (parameter data types);

Although I've not discussed the various data types at length, you should be comfortable with the idea that variables must have a type associated with them. Some types available in C are listed in Table 2.2.

Data Types		
Description	Example	
character	a, b, c, and so forth	
integer	1, 2, 3, and so forth	
number with a decimal point	1.0, 2.02, 3.5, and so forth	
non-type	an undefined or non-type	
	Data Types Description character integer number with a decimal point non-type	Data TypesDescriptionExamplecharactera, b, c, and so forthinteger1, 2, 3, and so forthnumber with a decimal point1.0, 2.02, 3.5, and so forthnon-typean undefined or non-type

There is much more to know about data types, but we'll postpone the discussion until we visit them more fully in the section, "Data Types."

Return Type

The first thing to decide when writing a function, or to determine when employing an existing function, is the function's return type.

The function can return an integer specifying some success or failure. Perhaps there is data formed and returned by the function. Determining the return type will enable you to satisfy the first field of your function definition.

The return type of a function will be one of the data types from the first column of Table 2.2.

III Note

You usually don't need to prototype an existing function because this is the responsibility of the author of the function. This is true in the C language because the functions provided for use have a corresponding prototype in a header file provided with the library. A challenge of learning C is becoming aware of the functions provided by the language and the correct header file to include when employing them.

Function Name

Following the return type of the function, you must include the function's name in the function's prototype. No matter what you've learned about computer programming, choosing names for variables and functions is the hardest part.

When choosing a function name, try to be as clear as possible without overdoing it. A function name too terse is as bad as a name that isn't sufficiently succinct.

EXCURSION

Naming Functions and Variables Is More Difficult Than Naming a Pet

Valid function and variable names must start with either a character or an underscore (_) followed by more characters and/or numbers. A function name cannot contain spaces or punctuation and should be descriptive of the task it performs.

For instance, *addTwoNum* is an appropriate name for a function that adds two numbers. Because function and variable names are allowed to contain numbers, *add2Nums* is also acceptable.

Parameters

Continuing to form the function prototype, following the function's name, the prototype is completed by providing a comma-separated list of the data types expected as arguments (parameters).



A parameter list is the data provided to a function.

Because the return type associated with a function provides the caller with a means of getting data out of a function, the parameter list is a way to pass data into a function.

Consider the example where you have a function that adds two numbers together.

The prototype for this example would be the following:

int addTwoNum(int, int);

Although I have yet to write the function (that is to define it), you know already what it looks like.

The function adds two numbers together so that we must pass the two numbers to be added into the function. As the purpose of the function is to sum two numbers, the return type should be the result of the addition. The only thing not immediately evident is what to call it (the hardest part). Choose a name as descriptive as possible without needing punctuation.

Notice that in specifying the data types required by the function (the return type and parameter list) you were not concerned with the variable names associated with the types. This is true for a function's declaration: It cares nothing for variable names. However, including them is not a violation of syntax, as in

int addTwoNum(int num1, int num2);

A function's prototype provides a wealth of information to the compiler. Once it is declared, you are free to use the function no matter where it resides in your project structure.

Even if the compiler does not know the contents of the function (its definition), you are still able to use it by merit of its prototype.

Definition

To define a function is to write it. The syntax for writing a function is similar to declaring it with two notable exceptions. Although optional in our prototype, we must include variable names in the parameter list, and we must specify the startand end-of-body markers.

```
int addTwoNum( int num1, int num2 )
{
    function stuff goes here
}
```

The parameter list is the method of passing data into the function. In this example, you are passing two integers *num1* and *num2* into the function.

Presumably (based on the function name), the two numbers will be added together and the sum returned to you by the return statement. The data that the function returns dictates its return type.

The return Statement

The return statement is not present in the function body of the example, but it is required. Without it, the compiler reports the following error message:

```
Control reaches end of non-void function
```

From this you can infer that if your function does not return anything, its return type must be void.

The return statement for the function *addTwoNum* has the value of the sum of the numbers passed:

```
return sum;
```

Similarly, if the function does not receive anything, meaning that it accepts no data in the parameter list, the parameter list is void.

```
void SomeFunction( void )
```

The necessity of a return statement within the body of the function is important to remember. It must return data consistent with the return type of the function.

If the function has a void return type, no data is returned and the return statement is *optional*. If an optional return statement is present in a function, it cannot return data, as in the following:

```
void SomeFunction( void )
{
  return;
}
```



All C programs must have a function called main because this serves as the entry point into the program. When the program is executed, the function main is called. Examples of defining main with the required parameter list are covered in Chapter 3.

Having alluded to different data types and their associated size to satisfy elements of our discussion on functions, we are now ready to turn the focus to data.

Data

Perhaps the most important aspect of computer problem-solving is the understanding of data and its representation within the computer. Knowing differences between a byte, an integer, and a float is imperative for a computer programmer.

The understanding of data is even more crucial with the C programming language because C allows the representation of all data by reference.



Accessing data *by reference* means referring to its address (location) in memory. A data element used by reference is also called a *pointer* because you are pointing to the data instead of pointing to the value of the data.

EXCURSION

Declaring Variables in the C Language

In the C programming language a variable is declared as follows:

int sum;

which follows the form

datatype variable name;

Any reference to the variable sum after this declaration is a reference to its value.

```
{
    int sum = 0;
    sum = sum + 1;
}
```

In the previous example, you are adding 0 (the current value of sum) and the constant 1 together, storing the result in sum, which is an integer (int).

The C programming language permits referencing data (variables) by their addresses. This is used extensively throughout the Graphics Editor project. Employing variables by reference is a very useful and powerful feature of the C language. A clear benefit of variable pointers (variables by reference) is realized when passing data to a function's parameter list. When data is passed as a parameter to a function, it must be copied to a temporary storage location called the *stack* in order for the function to see it. As will be evident when we define the data structures used in the Graphics Editor project, the data being copied to this temporary location can be very large. The larger the data being copied, the larger the stack must be, and the longer it takes for the computer to copy the data.

However, if the data passed to a function is a pointer to where the data already resides, only the pointer is copied to the stack. Relatively speaking, pointers are small and quickly copied. Further, passing the address of where the function can find the data enables the function to modify the data with the results visible throughout the program.

Declaration of the variable sum as a pointer would appear as the following:

int * sum;

Notice the asterisk (*), which wasn't in the declaration of sum in the previous example. In this declaration, sum is declared as a pointer to an int and not itself an int.

A pointer refers to an address in memory.

Demonstrating the danger of pointer manipulation, consider that the variable sum is a pointer (reference to an address of where an int can be stored), and therefore, an assignment such as

sum = 0;

sets the *address* where an int would be stored to 0x0000, which is invalid. The next *correct* assignment to sum would result in data being placed at address 0x0000 in memory and a crash (segmentation violation) would soon follow because the assignment was beyond the bounds (segment) allowed.

The correct method of accessing and assigning values of pointer variables is discussed later in this chapter; however, it is important to appreciate the flexibility of the C programming language in its representation of data.

Pointer manipulation is one of the most powerful (as well as one of the most dangerous) features of the C language. The authors of Java thought so ill of it that the Java programming language does not allow pointer references.

The goal of this section is to gain an understanding of the data types available in the C programming language. After discussing data types, focus will shift to understanding data by reference, as our Graphics Editor project uses this feature of C extensively.

Data Types

The first consideration when deciding what data type to assign when declaring a variable is the information to be stored by the variable. A second consideration is the precision (or space) required for the anticipated maximum and minimum values.



Deciding the type of information stored by a variable is an easy first step because whether the variable will be used for whole numbers, characters, or floating point digits should be clear from the context the variable is used.

Refer back to Table 2.2 for the data types available for representing information held in a variable.

When selecting a data type from Table 2.2 based on the criteria of the type of information being represented (stored), you must appreciate that the data type has an associated size.

The size of a variable determines the limit of information that can be stored in it. Trying to fit more data into the storage space allotted by the implicit size of the data type will either result in corrupted data, or worse, a segmentation violation. (See Chapter 3, Figure 3.2 for an explanation of a segmentation violation.)

The char Data Type

The character data type (char) is represented in computer memory as one byte of data. As you might recall, a byte consists of eight bits, as shown in Figure 2.4.

Figure 2.4

Size of char.



In an earlier discussion, I represented the capital letter A as a decimal 65 or binary 0100 0001. Because it is the computer representation that matters and not the form that we are most comfortable viewing it in, let us consider the eight bits (byte) that the machine needs for representing a character. The binary representation of the letter A (0100 0001) has eight digits, each corresponding to one of the eight bits allowed for a character in computer memory.

Remember that these eight bits are bound by the binary numbering system in which there are only two digits, 0 and 1.



Although there are eight bits within a byte, the most significant bit (furthest left in our representation) is used as a *sign bit* (+/-) and does not count in the value. Instead, the sign bit determines whether the data value represented is negative (bit set to 1) or positive (bit set to 0).

To determine the maximum value able to be represented by a variable of type char, set all bits available for representing data to 1 and add up the value.

With all available data bits in a char set to their maximum value, you get s111 1111 where the s reminds you that there is a sign bit. Calculating the value requires a review of the binary numbering system.

As in the decimal numbering system, each column is weighted. When you balance your checking account you are comfortable with the fact that the weight of each column is the number of digits available within the numbering system raised by the column number.

In Figure 2.5, column weights for the decimal system are demonstrated (the numbering system used to balance your checkbook and pay your speeding tickets).

н		
\mathbf{U}		
Ν		
D		
R	Т	0
\mathbf{E}	\mathbf{E}	Ν
D	Ν	Е
S	S	S
10^{2}	10^1	10 ⁰
\$ 1	9	8
\$	H U N E D S 10 ²	H U N E E D N S S 10 ² 10 ¹

Reading from right to left, number the columns starting with zero. For each column, raise the number of digits available for the numbering system by the column number.

Notice above the number 8 in Figure 2.5 that 10 is raised to a power of 0. There are 10 digits available to the decimal numbering system and 0 is the column number, making the weight of the first column 1 (any number to the power of zero is 1).

After you've determined the weight of each column, multiply the number in that column by the weight of the column and add it to its neighbor:

 $(1 \times 100) + (9 \times 10) + (8 \times 1) = 198.$

We are so used to the decimal numbering system that we might have forgotten how it works.

Apply the example of the decimal numbering system to the binary column weights shown in Figure 2.6.

Figure 2.6	127								
Binary column weights.	S I G N	64 ·	+ 32 ·	+ 16	+	8 -	+4-	+ 2	+ 1
	27	26	25	24		23	22	21	20
		1	1	1		1	1	1	1

The maximum value a variable of type char can represent is 127.

Referring again to the ASCII table (see Note in previous section "Characters," page 54), you see that a value of 127 is sufficient for representing any character found in the ASCII table.

To determine the minimum value that can be assigned to a variable of type char, set all bits available for data to 1 and set the sign bit as well.



Setting the *sign bit* simply makes the entire value negative. Therefore, a variable of type char can represent any value from -127 to 127. However, what happens if you assign a value greater than 127 to a variable of type char?

If you've done the math, you've discovered 2^7 (the column weight of the sign bit) is 128. Therefore, representing any value greater than 127 using the data type char requires setting the sign bit. Although you think you are assigning, for instance, 129 to a variable of type char, the value it holds after the assignment is really -127 because you've exceeded its maximum value and data has *overflowed* to the sign bit column.



The overflow condition is evidence of the necessity for type checking as discussed previously. If a character (char) variable is expected as a function parameter, but a data type with a greater maximum value is passed, the likelihood of exceeding the maximum value of the char is high.

How do you store a value greater than the 127 allowed by a char data type?

One way to exceed the maximum value of the char data type is to modify the char with the C keyword unsigned.



The *unsigned modifier* preceding a data type instructs the compiler to use the most significant bit for data instead of the sign bit. Clearly, a variable declared as unsigned is not capable of representing negative numbers. However, depending on the use of the variable this might not be an issue.

An unsigned char has a maximum value of 255 (128+64+32+16+8+4+2+1) and a minimum value of 0.

A second way to store values greater than the maximum of a char is to declare the variable as data type int.

The int Data Type

The data type int (integer) employs four bytes in computer memory for storing data, as indicated by Figure 2.7.

Figure 2.7

Size of int.



Figure 2.7 gives instant appreciation that the maximum value able to be represented by an int is significantly larger than that of a char.



The size difference between a char and an int is not linear because the column weights in the binary numbering system increase exponentially.

How many data bits are available within the four bytes of the int?

```
4 (bytes) \times 8 (bits per byte) – 1 (sign bit) = 31
```

There are 31 data bits in the data type int (integer). Table 2.3 shows the column weights of each of the 31 bits.

Table	2.3	Maximum	Value	of	an	int	

Column Weight	Value
2 ³⁰	1,073,741,824
2 ²⁹	536,870,912
2 ²⁸	268,435,456
227	134,217,728
2 ²⁶	67,108,864
2 ²⁵	33,554,432
224	16,777,216
2 ²³	8,388,608
222	4,194,304
2 ²¹	2,097,152

220	1,048,576
2 ¹⁹	524,288
218	262,144
217	131,072
2 ¹⁶	65,535
215	32,768
214	16,384
213	8,192
2 ¹²	4,096
211	2,048
2 ¹⁰	1,024
2 ⁹	512
2 ⁸	256
27	128
2 ⁶	64
2⁵	32
24	16
2 ³	8
2 ²	4
2 ¹	2
2°	1
Total	2,147,483,647

2



A variable of type int is capable of representing any number between -2,147,483,647 and 2,147,483,647.

As with the char data type, the modifier unsigned can be applied to the int. An unsigned int has a range of values from 0 through 4,294,967,295.

Because the data type int is able to represent much larger values than the char, why select the char?

Just as the size of the data type enables it to represent larger data values, it also requires more space in memory. Based on the data requirements, a programmer must decide the appropriate type to assign a variable. It is poor design to always use the largest data size available.

A programmer must give careful thought to how the data will be employed, correctly anticipate its maximum and minimum values, and choose a data type to represent it best.

Having covered the data types int and char and the modifier unsigned, let's continue by looking at other data types for representing numbers in the C programming language.

The short Data Type

Like the int, the short data type is used to represent numeric values. However, the short data type only employs two bytes for its data representation, making it smaller than the int and therefore capable of representing a smaller range of values.



The 2 bytes (16 bits) available in a short allow it to represent values in the range -32,767 to 32,767. Applying the unsigned modifier to the short (unsigned short) changes the range to between 0 and 65,535.

The long Data Type

Another data type for numeric representation is the long. With 8 bytes (64 bits) of storage capacity, the long is the largest numeric data type with values between -9,223,372,036,854,775,807 and 9,223,372,036,854,775,807. As an unsigned, the long can represent between 0 and 18,446,744,073,709,551,615.

EXCURSION

A Caveat to the long Data Type...

The 64 bits required by the long must be supported by the platform. As reviewed in Chapter 1, "UNIX for Developers," PC platforms (Intel 80x86 architecture) are 32-bit machines. When only 32 bits are available to the long, it mirrors the int data type.

Other architectures exist that support the full 64 bits (for example, DEC Alpha).

A third data type provided by the C language allows for representation of real numbers.

The float Data Type

The float data type is used to represent numbers containing a decimal point or fractional value. Less important than size when choosing the float is the precision.

A larger data type, and therefore one significantly more precise, is the double. The double maintains up to 8 bytes of data for representing floating point numbers.

Consistent with other data types, choosing between the float and double depends on data requirements of the application and what is being represented.

In an application that, for instance, maintains data representing currency, a float can be adequate for representing fractions of dollars. However, if the task is to calculate world or geospatial coordinates, a double can be necessary for pinpoint accuracy.

Having covered the basic data types of the C programming language, we are ready to create our own.

Defining Structures

Structures are groupings of data types used to create a new entity.

The syntax for defining a structure requires use of the keyword struct followed by the name to be assigned to the new entity. Begin ({) and end (}) body markers enclose the fields of the structure.

```
struct mystruct {
    int count;
    char flag;
};
```

The example structure has two fields: an integer named *count* and a character *flag*. Separating the name of the structure and the name of a field with a period is one way to access fields internal to the structure.

```
mystruct.count = 1;
```

Defining mystruct this way makes it available for use in your program as well as for declaring other occurrences of it.

Declaring another occurrence of mystruct is accomplished with the syntax

struct mystruct newStruct;

where struct mystruct is treated as a data type and *newStruct* as the variable name of the new occurrence.



Remember that struct mystruct must be defined before a declaration that employs it.

In the following example, newStruct is declared to be a struct mystruct and its fields are referenced as with mystruct:

```
newStruct.flag = 'a';
```

Optionally, C provides the keyword typedef for defining new data types.

Defining Data Types

To define the structure mystruct as a new data type requires the typedef keyword as a modifier to struct:

```
typedef struct {
    int count;
    char field;
} mystruct;
```

The modifier typedef instructing the compiler to define a new data type is different for this definition of mystruct. Also, the name of the structure follows the structure definition.

Following the typedef of mystruct, variables are declared in the following form:

```
mystruct newStruct;
```

Notice the keyword struct is no longer necessary in the declaration of *newStruct*. Further, following the typedef of mystruct, mystruct is not available for use in any other capacity except as a data type.

The last thing to consider before focusing on the syntax and convention of the C programming language is the correct method of referring to pointer variables.

Data by Reference

As discussed in the introduction to data types, the C language has the capability of referring to variables by their location in memory.

A variable, which refers to an address where the data is stored, is called a *pointer*. A variable of any data type can be declared a pointer.

Syntax for declaring pointers uses the asterisk (*) to indicate that the variable is an address.

int *intPtr;

In the previous example, intPtr is a pointer to where an int can be stored.

Consider a declaration of integer in the following manner:

int i;

At the moment of this declaration the value of *i* is unknown and considered *garbage*; therefore, the variable must be initialized.

The same is true for the value of intPtr. The address where an int may be stored is, at the moment of intPtr's declaration, unknown and intPtr must be given a known value.

EXCURSION

Automatic Variables Employ Volatile Memory

Variables declared within a function are called *automatic* variables. The memory an automatic variable uses for storing its value is provided from the stack.

The stack was said earlier to be temporary storage used by the program. In addition to being temporary, the memory associated with the stack is very volatile: It is changing constantly.

Every time a function is called, a new *frame* is placed on the stack. A frame is a structure containing fields the program needs to store information about the function call. This information includes the function's parameter list, its automatic variables, and the point to return to when the function completes.

When a function finishes, the frame is removed from the stack and replaced with a frame for the next function called.

The contents of the memory where a stack frame is placed to support a function call is uninitialized except for the values assigned by the system to the fields that it controls. These fields include the return point and parameter data (assuming that the correct number of arguments was passed). Although memory is reserved for automatic variables, the content of this memory is entirely unknown and referred to as garbage.

An assignment into intPtr before its initialization would be a segmentation violation and a fatal error.



An assignment can be made to any variable, but only pointers can be assigned into, or have a value placed in the location (address) pointed to by the variable. Before an assignment can be made into intPtr, a valid address must be assigned to it.

Consider the following syntax for correctly initializing the intPtr:

```
1: {
2: int i;
3: int *intPtr;
4:
5: i = 0;
6: intPtr = &i;
7: }
```

In this example, a variable of type int (i) and a pointer to an int (*intPtr) are declared. The variable i is initialized to 0 and the variable intPtr is assigned the address of i. Notice the use of the ampersand (&) for obtaining the address of i.

Following this example, intPtr is assigned a valid address for storing values (the address of i). It is now possible to store values into intPtr. To do so requires the following syntax:

*intPtr = 1;



intPtr points to a memory address for storing an integer. *De-referencing* the variable with an asterisk (*) indicates the *contents* of the storage location or, literally, the value stored there.

Finally, consider how pointer manipulation applies to structures.

Modeling an example similar to intPtr and borrowing the structure definition mystruct results in the following code fragment:

```
1:
      {
2:
        typedef struct {
3:
          int count:
4:
          char flag;
5:
        } mystruct, *mystructPtr;
6:
7:
        mystruct newStruct;
8:
        mystructPtr newStructPtr;
٩·
10:
        newStruct.count = 0;
        newStruct.flag = 'a';
11.
        newStructPtr = &newStruct;
12:
13:
      }
```

Varying from the previous example of typedef, this example declares two new data types:

5: } mystruct, *mystructPtr;

Applying what has been learned, you should realize that the variable newStruct is an occurrence of the structure mystruct. However, newStructPtr is a pointer to where a structure can be stored.

Just as the contents of newStruct are unknown at its declaration, meaning the value of the fields newStruct.count and newStruct.flag are considered garbage, the address pointed to by newStructPtr is also unknown at its declaration:

```
7: mystruct newStruct;8: mystructPtr newStructPtr;
```

Therefore, initializations must be performed:

```
10: newStruct.count = 0;
11: newStruct.flag = 'a';
12: newStructPtr = &newStruct;
```

Because newStruct is an occurrence of the structure, it has memory associated to it. The address of the memory is obtained by using the ampersand (&) before the variable name. This address can be used in an assignment to newStructPtr:

```
12: newStructPtr = &newStruct;
```

To reference the fields of a pointer to a structure, such as newStructPtr, requires syntax different from referencing fields of an occurrence of a structure. With intPtr it was necessary to de-reference the address the variable pointed to. The same must be done to newStructPtr; however, the symbol to de-reference the address is different:

```
newStructPtr->count = '1';
newStructPtr->flag = 'b';
```

The dash followed by an arrow (->) is the correct syntax for referencing the fields pointed to by newStructPtr.

Next Steps

The discussion of programming conventions and data types throughout this chapter employed many elements of the C language syntax; however, we have not seen many subtleties and nuances of the language. Further, it is important that the requirements of the language be specified. The next chapter, "A Word on C," provides a thorough look at the syntax and conventions of C.

Chapter 3

In this chapter

- Hello World
- Conclusion
- Next Steps

A Word on C

Having been exposed to the C language syntax in Chapter 2, "Programming Constructs," we now focus full attention on the structure and convention of the language.

Every year the C community runs a contest to see which participant can write the most obfuscated C code.



obfuscate: -cated, -cating, -cates. 1. a. To render obscure. b. To darken. 2. To confuse.

The point of the contest is to show the importance of programming style in an ironic way and to emphasize subtleties of the C programming language. Although other structured programming languages have intricacies of their own, C inherently lends itself to the contest by offering features that are often confusing.

This chapter will lead you through a review of the syntax seen in the examples of Chapter 1, "UNIX for Developers," and Chapter 2, "Programming Constructs," and expand your exposure to the C language.

If you are already comfortable with your understanding of the C programming language, feel free to proceed to the discussion in Chapter 4, "Windowing Concepts."

The C programming language has a significant evolutionary history in the world of computer science. It was derived from the B language written by Ken Thompson in the late 1960s. (The predecessor of B was BCPL written by Martin Richards.)

The importance of this history is to recognize that C did not appear as a new language but was adapted from an existing language.

C's predecessor, B, was a language without types wherein it was up to the programmer to ensure that variables were used in a valid context.

The previous chapter discussed data types and the challenge they bring to computer programming. On the heels of this, you have to wonder how confusing the B language could have been.

Dennis Ritchie wrote C in the early 1970s, keeping most of B's syntax. Two elements that are characteristic of C, and often cause the most confusion with the language, are the relationship between pointers and arrays and the similarities between declaration syntax and expression syntax.

Developing the necessary awareness of C pitfalls while learning correct programming structure and C syntax is a goal of this chapter.

Hello World

A constant when teaching C is that at the end of the exercise the student is able to print *Hello World*. Using this as our structured example and bringing together lessons from previous discussions, let's begin with a file to hold the C source code.

A file can be created using the vi command (refer to "The vi Editor" in Chapter 1, page 22 for a review of the command):

```
bash[1]: vi first.c
```

Because every C program must define a function called main, begin by inserting the following code into the file:

```
0: /* the start of the program */
1: int main( int argc, char *argv[] )
2: {
3: printf("Hello World");
4:
5: return( 0 );
6: }
```

Comment Tokens

Use of the comment token recognized by the C language starts our example:

```
0: /* the start of the program */
```

The slash asterisk (/*) combination begins the comment and an asterisk slash (*/) ends it. Anything contained within the start and end comment tokens is ignored by the compiler and does not get placed in the object file. The GNU C Compiler also accepts the token slash slash (//) for support of a comment contained on a single line:

// everything after this comment token is ignored

The Function main

Continuing the discussion of the code sample, look closely at the definition of the function main:

1: int main(int argc, char *argv[])

Because the compiler requires the function main, it does not need a prototype (declaration) before its definition. The compiler expects the function and therefore dictates its parameter list and return type.

EXCURSION

A Syntax for Representing Arrays of Strings

The declaration of the variable argv uses a syntax that you've only seen in part.

char *argv[];

Use of the char data type as a pointer (char *) is for storing a string or group of characters.

char *str = "abcdefghij";

Further, the syntax of following a variable name with square brackets indicates that the variable is an array (argv[]) (see the section "Data Types" in Chapter 2, page 70 for a review of arrays).

Defining a variable as

char str[10];

requests the compiler create an array with enough space in memory to hold 10 characters.

```
str[0] = 'a';
str[1] = 'b';
.
.
.
.
str[9] = 'j';
```

Combining the two in a single declaration as with argv from the example, we see

```
char *strArray[3];
```

which indicates that strArray is an array of strings as in

strArray[0] = "abc"; strArray[1] = "def"; strArray[2] = "ghi";
86 Part I *Absolute Zero*

Reviewing the definition of main, you see that many elements have already been discussed.

Identifying the data types in the function definition

1: int main(int argc, char *argv[])

reveals the return type of the function main is the data type int. Further, main expects two parameters, an int named argc and an array of char * (character pointer) called argv.

EXCURSION

Compiler Differences Affecting the Declaration of the Function main

The C compiler used imposes the return type and parameter list of the function main. The arguments argc and argv are consistent with all C compilers. These parameters enable the program to access the command-line parameters the user passes.

For instance, in Chapter 1 we used man to illustrate passing flags and parameters to UNIX commands:

bash[1]: man ascii

With the execution of the man command, the operating system responds by calling the function main defined by the program's author. When main is entered, the first parameter, argc, specifies the *number* of arguments placed on the command line when the user invokes the program and argv holds the *value* of each of them.

Invoking man ascii results in the value of argc being set to 2 where argv[0] contains man and argv[1] ascii. A way of representing this is

char *argv[] = { "man", "ascii" };

Notice that in this example, argv contains two elements, which correspond to the value of argc.

The return type expected from main is not consistent between compilers. Unlike Linux, most SYSV versions of UNIX allow a return type of void when defining main.

A return type is beneficial because it gives you the ability to test for the program's success or failure. By convention, a program exits with a value of **o** to indicate that no errors occurred. If the program fails to complete, a value greater than 0 is generally returned.



Think of argc as the *argument count* and argv as the *argument values*.

Code Bodies

Now that we've evaluated the definition of the parameter list and return type for main, let us consider the function's body:

```
2: {
3: printf("Hello World");
4:
5: return( 0 );
6: }
```

With any body of code, you must inform the compiler where the body begins ({) and ends (}).

In the previous example, no automatic variables are defined in the body of main; however, when automatic variables are defined, they must be placed immediately after the open brace marking the beginning of the code body.

EXCURSION

Where to Define Code Bodies in the C Language

Code bodies can be placed anywhere within a function and they can be nested. Code bodies define functions and associate a body of code with a loop or decision; however, code bodies can also be unconditional within a function. For instance, it is valid to have code bodies within bodies:

1:	int main(int argc, char *argv[])
2:	{ // start of function body
3:	int done = 0;
4:	<pre>printf("Hello World");</pre>
5:	{ // start of unconditional body
6:	while(!done) { // conditional body
7:	/* do something */
8:	} // end while
9:	} // end of unconditional body
10:	return(0);
11:	} //end function body

Notice the automatic variable **done** in this example. The declaration of **done** occurs immediately after the start of a code body, specifically, the function's body; however, variables can be declared after any start-of-body marker.

EXCURSION

A Stylistic Note...

When code bodies are added to functions (as in the preceding code), it is a courtesy for those who follow behind you, and it also increases the readability of the code and adds a level of indentation to the code contained in a body. For instance, the code in the first body would be tabbed once, the code in the second body twice, and so on, with the associated begin and end body markers always in the same column. Consider the previous example without using any indentation and without the benefit of comments:

```
1:
    int main( int argc, char *argv[] )
2:
    {
3:
    int done = 0;
4:
    printf( "Hello World" );
5:
    {
6:
    while( !done ) {
7:
    /* do something */
8:
    }
9:
    }
10: return( 0 );
11: \}
```

Only by carefully studying the sample can you determine the relationships between the various code bodies illustrating the utility of proper indentation.

When automatic variables are declared within a new body of code, they are governed by rules of visibility known as *scope*.

Variable Scope

Automatic variables declared in a body of code are visible from the moment of the declaration until program execution reaches the end of the body.

```
int main( int argc, char *argv[] )
1:
2:
   {
3:
        int cnt;
4:
        /* sum is not visible here as it has not been declared */
5:
        {
6:
            int sum;
7:
            /* cnt is still visible here as
8:
               execution has not reached the body
9:
               in which it was declared */
10:
        }
        /* sum is no longer visible as the body
11:
           in which it was declared has ended */
12:
13: }
```

A variable is said to be *in scope* when it is visible and *out of scope* when it is not.



Variables defined outside a function are called *global variables*.

If a variable declaration occurs outside a function, its scope is extended to all functions and code bodies that are defined after the point of the declaration.

```
7: }
8:
9: int add2Nums( int num1, int num2 )
10: {
11: return( num1 + num2 );
12: }
```

This example takes into account nearly everything discussed so far. As required by the GNU C compiler, the function main is defined as returning int and expecting the parameters of argc and argv. When main is called, it assigns the value of the global variable sum the result returned by the function add2Nums.

The function add2Nums expects two arguments of type int, which main provides by passing 1 and 2.

Notice that add2Nums does not have to return anything but could make the assignment directly to sum.

```
9: void add2Nums( int num1, int num2 )
10: {
11:    sum = num1 + num2;
12: }
```

As the variable sum is defined before the function main, it is visible to main as well as to add2Nums. Notice that as add2Nums now has a void return type the return statement may be omitted. Further, main would simply invoke add2Nums as

```
6: add2Nums( 1, 2 );
```

A significant detail has been overlooked in this example; there is no declaration of add2Nums prior to its invocation by main.

Figure 3.1 shows the error issued when an attempt is made to compile this sample.



The GNU C compiler is relatively clear on what is wrong with the sample code.

The compiler makes up an *implicit declaration* when no actual declaration is found before a function is called. Implicit declarations are avoided by explicitly defining functions.

Notice the portion of the error output in Figure 3.1 that reads

sample.c:11: warning: 'add2Nums' was previously implicitly declared to return 'int'

This is consistent with the description in Chapter 2, section "Forward Declarations," page 64 of the compiler's behavior. Anything the compiler doesn't recognize is assumed to be an int, which is the default data type in C.

At line 6, according to the error output, the compiler found an implicit declaration. This line corresponds to the first use of the function. Later, at line 11, it found the actual definition of the function that wasn't consistent with what was implied (the compiler's assumption).

To resolve the error, simply add a new line at the beginning of the file declaring the function prior to its use:

```
0: void add2Nums( int, int );
```

Knowing how to define, declare, and employ functions and variables correctly is critical in advancing our knowledge of the C programming language. Building on this knowledge, it is important to become familiar with the functions that C provides us.

Built-In Functions

You will not have to author all the functions that your application employs because the C programming language provides many functions for you. When you become aware of functions inherent to a language, you increase your proficiency in it.

The availability of functions provided by an environment such as X Window or a third-party package adds to this challenge.



Third-party packages are ones that you add to your system and are generally not part of the standard installation process. These packages are usually purchased separately and selected for the functionality they add to your development or runtime environment. The OSF/Motif Widget Set is an example of this; it compliments the X Window System because it provides elegant three-dimensional features to X Window-based application development. Focusing on some of the functions provided by the C programming language, look again at the code sample from the Hello World example that started the chapter:

3: printf("Hello World");

The printf Command

The printf command is a function the C language provides for printing formatted output to the terminal. The use of printf is extremely flexible because it supports every data type recognized by the C language.

EXCURSION

Terminals Aren't Just Where Planes Depart

A terminal may be your screen or a terminal emulator (window) such as the xterm depending on whether your system is running in graphics or text mode.

Generally, printf places its output on the command line following where the program was executed. This destination is known as *standard out* and represented in C as stdout. In addition to standard out, the C language also provides the destination standard error (stderr) for output.

Standard out and standard error start as the same destination; however, they can be altered through the use of redirection as discussed in the section "grep, Pipes, Redirection, and more" in Chapter 1, page 44.

Chapter 4 covers windowing concepts and terminal emulators in detail.

As with the function add2Nums defined earlier, there must be a forward declaration (prototype) for the function printf. Because the C language provides the function, it also provides the prototype.

The printf function's declaration (and many other functions performing input and output) is found in the file stdio.h, which is part of the standard C header files.

EXCURSION

Another Look at the Use of Header Files

Header files, as discussed in the Note found in the section "Definition" in Chapter 2, page 68, are source files often containing prototypes and other declarations shared by multiple files within a project.

You include a header file by using the include *compiler directive*. As the name implies, a compiler directive directs the compiler to perform a task or make a decision.

In C, all compiler directives are prefaced with the pound sign (#).

#include <stdio.h>

Used in the same manner as the include statement from Makefile syntax as discussed in Chapter 1, section "Include" on page 37, the compiler directive to include a header file

performs the inclusion at the point of the directive. The effect is that the compiler sees the contents of the file as if replicated by the directive.

The syntax when employing the include compiler directive offers a hint to where the file can reside. Specifically, when the header is part of the standard C library the < and > are used to enclose the filename. However, when the header file is part of a third-party package or one that you have authored as part of the project, the filename is enclosed by double quotes.

#include "gxGraphics.h".

The C compiler first looks in a standard directory such as /usr/include for files enclosed with the <> symbols and then it considers the paths specified by the -I flag passed when the compiler was invoked.

To review the gcc command and use of the -I flag, refer to Chapter 1, section "gcc -I", page 28.

To properly declare the printf function prior to its invocation, add the following line to the "Hello World" code sample:

0a: #include <stdio.h>

The syntax of the printf function follows the form

```
printf( "format string" [, arg][, arg][, ...] );
```

To accomplish the formatting recognized by the printf function, the *format string* ranges from a constant such as "Hello World" to accepting a variety of *formatting tokens*.



Notice in the syntax of the printf that the args are enclosed in square braces ([]), indicating that they are optional. An argument (arg) is only required if a format token is nested in the format string.

Table 3.1 shows some common formatting tokens recognized by printf that can be embedded in the format string.

able 0.1 Frinti Formatting Tokens					
	Token	Туре	Sample	Output	
	%S	char *	printf("String: %s", "Hello");	String: Hello	
	%C	char, unsigned char	printf(" Character: %c", 'A');	Character: A	

Table 3.1 Printf Formatting Tokens

Table 3.1	Continued			
%d	int, long, short, unsigned int, unsigned long, unsigned short	printf("Number: %d", 198);	Number: 198	
%f	float, double	printf("Real Number: %f", 3.14);	Real Number: 3.14	

All formatting tokens are prefaced with the percent sign (%), indicating to printf that what follows is for argument substitution. Further, for each token in the format string there must be a corresponding argument to satisfy the substitution.

Multiple arguments are comma separated and substituted in the order they are placed.

```
printf( "String %s and Char: %c", "Hello", 'A' );
```



As demonstrated in the previous example, C uses double quotes ("") to represent multiple characters (strings) and single quotes to represent a single character ('').



printf's capability to perform type checking is limited. A token nested in the format string expects a complimenting argument of a specific type. If an argument of a differing type is placed in the argument list, the results cannot be predicted. The printf statement will attempt to *cast* the argument, but because this is done at run-time (while the program executes) there is no recovery if the types are not compatible.

EXCURSION

Promoting Variable Data Types to Satisfy Type Checking

Casting from one data type to another is a way to promote variables to satisfy type requirements and avoid compiler warnings. For instance, a character (char) is easily promoted to an integer (int), which has a larger storage capacity.

The code fragment

char chr = 'A'; int num = (int)chr; assigns the letter A to the variable chr. Because the letter A has a decimal equivalence of 65, casting chr to an int would assign the value 65 to the variable num.

Caution must be used when casting from a larger data size to a smaller, however, because data could be lost. Consider the fragment

```
int bigNum = 999;
char chr = (char)bigNum;
```

Because the maximum value of a character is 256, the assignment of **bigNum** to **chr**, although made legal by the cast, results in the new value of **chr** being -25. Clearly, this is not the expected result.

See the section "Data Types" in Chapter 2, page 70 for a review of valid value ranges and the implicit size of recognized data types.

Looking again at the format string accepted by the printf function, it should be clear why printf is said to perform formatted output. Additional formatting tokens and token modifiers are available to printf for outputting data types in varying forms as well as controlling field widths, alignment, and more.

Review the printf man page for a full description of its capabilities.

The printf is one of several functions that C provides for performing formatted output. Similar to printf are the functions fprintf and sprintf.

Both fprintf and sprintf enable the output to be directed some place other than standard out. For instance, fprintf may be used to send the formatted output to standard error.

fprintf(stderr, "%s %s", "hello", "world");

Differing from the syntax of printf, fprintf requires as its first parameter the destination designator for the output. This destination can be one that C provides, such as stderr used in the example, or it can be one created by using the fopen (file open) function.

Before looking at the use of fopen, consider sprintf as it relates to the printf and fprintf functions. Use of sprintf enables a programmer to format output for placement in a *buffer*.



A *buffer* is a character array used for intermediate storage during input or output operations. For instance, in

char message[25];

message is declared as an array of 25 characters and could be used to satisfy the first parameter required by sprintf.

sprintf(message, "Error occurred at line %d", lineno);

EXCURSION

Apply Great Caution when Determining the Correct Size of an Array

In the sprintf example, message was declared with a length of 25. Count the characters in the format string passed to the function:

```
"Error occurred at line %d"
```

Including spaces, there are 23 characters in the format string, excluding the value of the formatting token %d.

The C function **sprintf** null terminates the output that it formats: It inserts a null character (**0**) at the end of the string. This termination is important for other C functions that can act on the string and it consumes one place in the buffer.

With the 23 characters in the format string and one character for null termination, a total of 24 characters are placed in the buffer message before the argument substitution for the token %d. If the value of lineno substituted in the message is only one digit (0-9), you have exactly filled the 25 character spaces available to message. However, if the value of lineno is greater than 9 (two or more digits), you will exceed the length of message because the null termination will be placed outside the valid memory associated with the buffer.

Figure 3.2 illustrates the effect on memory when the boundary of an array is exceeded, a condition known as a *segmentation violation*.

The value 69 placed in the buffer exceeds the allowed space for message. The owner of the memory that the 0 (null) overwrites is unknown. The memory can be unused or it can be a critical part of the program. Depending on the importance of the unknown space, the program might crash instantly, or it might only corrupt the value of the neighboring space, leading to a crash much later in program execution.

I∎ **⊂** Note

Improper use of arrays is one of the most common causes of program bugs.



A *program bug* is anything that causes a program to behave unexpectedly, often resulting in a crash.

Now that we've reviewed the built-in functions of printf, fprintf, and sprintf for formatting and outputting data, we can return to the use of the fopen function for creating destination designators to be passed to fprintf.

The fopen Function

The fopen function opens a file named in the parameter list returning a handle to the file known as a *file pointer*.

Incorporating everything discussed thus far, consider the following code sample illustrating the use of the fopen function:

```
1:
    #include <stdio.h> // for printf and fopen function prototypes
1a:
                        // and FILE structure definition
2: FILE *openFile( char * filename )
3:
   {
4:
         FILE *fp;
5:
         fp = fopen( filename, "w+" );
6:
7:
8:
         if( fp == NULL ) {
9:
             fprintf( stderr, "Unable to open file %s", filename);
10:
11:
        }
12:
        return( fp );
13: }
```

The code sample defines a function named openFile that accepts a single parameter filename and a character pointer, and returns a pointer to FILE.

Figure 3.2

An illustration of the error that results in a segmentation violation.







The FILE data type is a structure that C provides for referencing files opened with the fopen function. The structure is considered *opaque*, meaning that the fields defined within the structure are not to be accessed. The structure exists only to serve as a handle for manipulating files.

The standard out (stdout) and standard error (stderr) references provided by C are pointers to the FILE structure (FILE *).

In the code sample demonstrating use of the fopen function, notice the test on the value returned by the fopen.

8: if(fp == NULL) {

If fopen is unable to open the file specified, it returns NULL.



C provides NULL as a means of representing *nothing*. Literally defined as 0 (zero) and cast to a void pointer ((void *)0), it is returned in cases of failure to create a valid reference to a return value by functions such as fopen.

The fopen command expects two parameters. The first, evident by the example, is the name of the file to open:

fp = fopen(filename, "w+"); 6:

The second parameter is the *mode* in which fopen should open the file.

Table 3.2 shows the valid modes that can be passed to the fopen function and describes their effect.

la	able 3.2 File Modes Understood by fopen		
	Mode	Description	
	r	Open the file for reading	
	r+	Open the file for reading or writing	
	w, w+	Truncate to zero length or create; file is opened for writing	
	a, a+	Append, open, or create file for update, writing at the end of the file	

Table 0.0 Ella Madaa Understand bu

Upon successfully opening a file, fopen returns a file pointer (FILE *) reference which can be passed to functions requiring a destination designator such as fprintf:

```
fprintf( fp, "Line entered into file referenced by fp" );
```

Now that you're comfortable with the code sample illustrating the fprintf command, the use of NULL, function bodies, rules of scope, and variable and function declarations, we will consider another family of functions provided by C.

The C String Library

The lexical analysis of strings as discussed in Chapter 2, the section "Types of Conditions," page 52, was said to require a different method from integers and characters for forming test conditions. Because string manipulation is language specific, the C string library satisfies this requirement.

C provides many built-in functions for comparing, copying, and creating strings. The first to consider is the function for comparing two strings called strcmp (string compare). Passing the two strings for comparison satisfies the parameters required by the strcmp function:

```
1: #include <strings.h>
2: {
3:
        char *str1 = "tag",
             *str2 = "day";
4:
5:
6:
        if( strcmp( str1, str2 ) == 0 ) {
7:
            // the strings match
8:
        } else if( strcmp( str1, str2 ) < 0 ) {
```

```
9: // str1 appears before str2 in a dictionary
10: } else if( strcmp( str1, str2 ) > 0 ) {
11: // str1 comes after str2 in a dictionary
12: }
13: }
```

Notice that this code sample uses the include compiler directive to include the file strings.h:

```
1: #include <strings.h>
```

This is the header file that C provides to satisfy the built-in string functions' forward declarations.

EXCURSION

Combining Declarations and Assignments Using C Syntax

A variation on variable declaration appears in this code sample as the declaration of str1 and str2 are combined sharing the data type char.

3:	char	*str1	=	"tag",
4:		*str2	=	"day";

Syntactically correct, multiple variables of the same type can be comma separated at their declaration.

Seen previously but not explicitly noted is the combination of a variable's declaration and its initialization as shown in the previous example.

A sound programming habit is to initialize a variable prior to the variable's use. It saves you from having to scour code later looking for the obscure bug that lack of initialization might cause if the variable is employed in an expression before being assigned an initial value.

As implied by using strcmp in the previous code sample, the comparison of the two strings returns 0 if the strings are equal. The strcmp function returns a value greater than (or less than) 0, depending on the lexical analysis of the two strings.

EXCURSION

```
The Lexical Analysis of Two Strings of Varying Length
```

If the two strings being compared by strcmp are not the same length, as in

```
char *s1 = "act", *s2 = "abra";
```

```
if( strcmp( s1, s2 ) == 0 ) {
```

then the function compares the strings up to the point of finding the NULL termination or end-of-string marker. (See the Excursion in the section about sprintf for a description of string termination performed by C). It is possible to inform C how many characters of the two strings to compare by use of the strncmp function.

The strncmp function accepts one more parameter than its sister function strcmp. Specifically, the third parameter informs strncmp of the number of characters to consider in the comparison.

```
char *s1 = "act", *s2 = "abra";
if( strncmp( s1, s2, 3 ) == 0 ) {
```

In this example, you've explicitly said to stop comparing after three characters.

how too pro nouns' it

The function strncmp is read like it is written *stir-n-compare*.

Table 3.3 shows several string functions that C provides; they are common in programs employing the language.

Table 3.3 C String Functions

•		
Function	Example	Description
strcmp, strncmp	<pre>if(strcmp(str1, str2) == 0) { or if(strncmp(str1, str2, 4) == 0){</pre>	Compare two strings (optionally specifying number of characters to compare)
strcpy, strncpy	<pre>char str1[10]; strcpy(str1, "Initialize"); strncpy(str1, "Initialize" , 10);</pre>	Copy one string into another (optionally specifying number copy); Caution: the destination string, str1, must have space available for the number of characters being copied to avoid a segmentation violation
strcat, strncat	<pre>char str1[10]; strcat(str1, "Warn"); strcat(str1, "ing "); strncat(str1, "at line", 2);</pre>	Concatenate two strings; the second string specified is added to the end of str1 (optionally specify number of characters to add to the end of str1); Caution: the destination string,str1, must have space available to hold the contents of the second string or a segmen- tation violation will occur

3

continues

DIE 3.3 COIILIIIUEU		
Function	Example	Description
strchr	char *token, *buf = "No error"; token = strchr (buf, 'e');	Find the first occurrence of a character within a string; function returns either a pointer to the position of the character within the first parameter, or NULL if no match was found
strstr	char *token, *buf = "No error"; token = strstr (buf, "err");	Find the first occurrence of a sub-string within a string; function returns a pointer to the start of the sub-string within the first parameter or NULL if no match was found
strdup	char *newStr = strdup(" error_log");	Duplicate a string, returns a new copy of the string specified to the function
strlen	int len = strlen (newStr);	Return the length (number of characters) comprising the string

Table 3	8.3 co	ntinue
---------	--------	--------

Table 3.3 shows that dangers are associated with string manipulation because the space available when copying strings must be sufficient to hold the new contents. The idea of space in a computer program always translates to memory.

The caution extended in the introduction to pointer manipulation, variables by reference, (see the section "Data Types" in Chapter 2, page 70) must be applied to strings as well. In fact, it is the same warning described in the Excursion in the section "sprintf" earlier in this chapter. The common thread is a necessity for proper memory management.

Memory Management

Proper memory management is critical in application development. Careful attention has been paid in previous code samples to ensure that the compiler always *implicitly* provided the memory space necessary.

Memory management can be implicit or explicit. C provides built-in functions enabling a programmer to allocate and free memory explicitly. It is also possible that the necessary memory is implied by the manner in which a variable is declared.

Samples of associating memory implicitly with a variable include

```
char str1[10];
```

where the variable array str1, upon declaration, has sufficient space for storing 10 characters.

Consider also

char *buf = "No error";

where the variable buf is declared with space enough to hold 8 characters as implied by the initialization combined with the declaration.

Memory associated with variables can be taken from one of several areas available to an application. Variables declared after a start-of-body marker reside on the stack. (See Chapter 1 for a review of automatic variables and the stack.) The stack is volatile memory changing constantly during program execution.



A programmer affects the stack through the manner in which variables are declared. In other words, the programmer can never explicitly allocate or free stack memory.

The memory associated with the variable buf in the previous code fragment resides on the stack until it is out of scope. After it is no longer visible, the stack frame holding the reference to buf is removed from the stack and a new frame uses the memory.

Contrast the declaration of the two character pointers in the following example:

The variables buf and token both point to characters with the contents of buf being initialized at the moment of its declaration. Based on buf's initialization, you know both its size (length) and contents. Because no memory is associated with token, it has neither a length nor valid contents. Token is a character pointer, but as of yet points to nothing (garbage).

The cumbersome task of *memory management* thus begins.

A valid statement is to assign the memory associated with buf to token.

token = buf;

Because the variables are compatible (both character pointers) the assignment is legal and logical; now, token refers to valid memory (the implicit memory given to buf from the stack). Employing token follows the same rule as using buf, namely, the size of the memory provided cannot be exceeded (eight characters). 3

Furthermore, when the variables go out of scope, their contents are no longer valid because a new stack frame will begin using the memory that was once reserved for them.

Consider this incorrect code sample in which the memory associated with an automatic variable is returned to the calling function:

```
1: // forward declaration
2: char *someFunc( void );
3:
4: // entry point of the program
5: int main( int argc, char *argv[] )
6: {
7:
       char *badString = someFunc();
8:
        printf("The value of str is: " );
       sleep( 1 ); // new stack frame
8a:
       printf( "'%s'\n", badString );
9:
       return( 0 );
10:
11: }
12:
13: char *someFunc( void )
14: {
15:
       char buf[15];
16:
        strcpy( buf, "Error ahead" );
17:
        // return the memory provided from the
18.
        // the stack for the variable buf
19:
        return( buf );
20: }
```

As described in the previous example, when the function main calls someFunc, a new frame is placed on the stack to manage all aspects of the function call. Specifics such as the function arguments, the return address of the calling function, the return value, and the local (automatic) variables, as well as temporary storage needed during function evaluation, are part of the stack frame.

Therefore, the memory used to store the value of buf in the function someFunc resides in the stack frame managing the function call. When the function returns, the frame is removed and all associated memory is made available for subsequent function calls.



Warning

Returning and employing memory from the stack creates memory errors that will likely result in a program crash when the stack frame is no longer active.

In the previous example, if main did not exit immediately but instead invoked another function, a new frame would be placed on the stack and the memory previously associated with buf (still pointed to by badString) would be reused. The new function owning the memory location would store a new value, possibly a value that is not character data. This would ensure that the value of badString is not as expected and a program crash could soon occur.

To avoid the problem of returning memory contained on the stack, you must explicitly allocate memory associated with pointer variables or reserve them in non-stack memory if their use extends beyond a single function.

The static Keyword

The keyword static serves several purposes in the C programming language. One of its uses is to force the memory associated with a variable to come from non-stack memory. This is beneficial because the memory will be persistent through the life of the program and not just the life of the function, as we saw with the previous example.

Use of the static keyword follows the form

static data type variable name;

Applying this to the previous example to correct the imminent program crash would require that the declaration of the variable buf change from

```
15: char buf[15];
```

15: **static** char buf[15];

As stated previously, the static keyword informs the compiler that the memory associated with the variable must not be taken from the stack. Instead, persistent memory is used and therefore the contents of buf are maintained during the life of the program. In addition to the memory address being safely returned to the calling function with the line

19: return(buf);

the contents of the variable will be maintained in consecutive calls to the function. More examples of static automatic variables will be seen in the Graphics Editor project.



Memory associated with static variables can never be returned or unreserved by a program. Therefore, memory reserved as static permanently increases the size of a program.



You can also use the static keyword beyond making a variable's memory persistent to limit the variable's scope.

- \rightarrow As described earlier in this chapter, in the section "Variable Scope" on
- page 88, variables declared outside of a function body are global variables.

Global variables are visible (in scope) from the moment of their declaration until the end of the file. Global variables can be made visible to other files through use of the keyword extern.

The extern Keyword

The keyword extern can preface any variable declaration as a method of informing the compiler that the variable's *actual* declaration is *external* to the current file.

The declaration

extern int sum; // references a previous declaration of sum

tells the compiler that use of the variable sum is by merit of a previous declaration in another file. By using extern, the variable is global (visible) to multiple files.

However, if the original (actual) declaration employed the static keyword, visibility would be limited to the file in which it was declared. In other words, when you use the keyword static on a global variable, the variable cannot be declared externally through use of the extern keyword in another file.

Similarly, when functions are prefaced with the static keyword, their visibility, too, is limited to the file in which they are declared even if a prototype for the function exists elsewhere.

```
A function defined as
```

```
static int add2Nums( int num1, int num2 )
{
    return( num1 + num2);
}
```

can only be invoked by functions in the same file, which contains this definition because the static keyword limits its visibility.

Using the static keyword in the context of ensuring that a variable's memory is persistent during program execution is one method of managing memory that doesn't employ the stack. Another method of managing memory is to dynamically request it from an area known as the *heap*.

Dynamic Memory Allocation

The area of memory known as the heap that is available to applications is managed by using built-in functions provided by C for dynamic memory allocation.



The *heap* is an area of memory, separate from the stack, which is available for program use. Nothing automatically happens to heap memory as it does to the stack with frames added and removed constantly; this makes the heap a preferred memory location for data and structures with a life span greater than a single function.

Only heap memory can be dynamically allocated and freed using the built-in functions for memory management.

To dynamically allocate memory, C provides several functions. The first to consider is the function malloc.

malloc (Memory Allocation)

The allocation function malloc reserves a block of uninitialized memory from the heap in the size specified.

```
char *buf;
buf = malloc( sizeof(char) * 10 ); // space for 10 characters
```

If the function malloc is not able to reserve the requested memory, it returns NULL. Testing the return of the function ensures that a valid memory block exists and diverts a program crash.

```
if( buf == NULL ) {
    printf( "Fatal Error: Failed to malloc memory" );
    exit( 1 );
}
```

The allocation of buf in the previous example is equivalent to

```
char buf[10];
```

except that the memory returned from the malloc is heap memory and will exist after the function containing the allocation returns. This complicates memory management slightly because any memory allocated from the heap must be explicitly returned to the heap when the program is finished with it.

To return memory to the heap, making it available for subsequent allocations, C provides the function free. 3

free (Returning Allocated Memory)

Every block of memory dynamically allocated should have a corresponding free.



When you fail to free memory that is dynamically allocated within a program, you have caused a *memory leak*. Memory leaks cause programs to suffer in performance as well as face a potential failure.



The memory associated with a program grows in direct proportion to the size of the program's heap.

A program that never returns memory to the heap continues to grow in size until no further growth is possible; either the system runs out of memory or a maximum program size is reached, at which time further calls to malloc fail. Also, the greater the amount of memory consumed by a program the slower its execution.

The syntax for performing a free follows the form

free((char *)mem);

where *mem* is the memory gained by a dynamic allocation function such as malloc.



The function free expects a parameter of the char * type to be passed to it. Because the function is used to return the allocation of any type of data to the heap, it is often necessary to cast the memory being returned to a pointer of the proper type (char *).

Table 3.4 shows several functions provided by C for performing dynamic memory allocation.

Function	Example	Description
malloc	<pre>memPtr = malloc(size);</pre>	Returns a pointer to an unitialized block of memory; where <i>size</i> is the size of the block, for example, (sizeof(<i>data</i> <i>type</i>) * <i>count</i>)
calloc	arrayPtr = calloc(size, num_ele);	Returns a pointer to an array initialized to 0, with num_ele (number of elements) of the specified size
realloc	<pre>memPtr = realloc(memPtr, new_size);</pre>	Expands an existing block of memory to a new_size; current contents of memPtr are transferred to the expanded block

Table 3.4 Memory Al	location	Functions
---------------------	----------	-----------

Function	Example	Description
strdup	char *newStr = strdup	Duplicates a string; constitutes a mem-
	("Abcde");	ory allocation function because the
		results of strdup must be explicitly freed

Let me give you a final word of caution for developing proper memory management skills: Care for the relationship of automatic variables placed on the stack and references to dynamically allocated memory, which at some point must be returned to the heap.

Memory Leaks

Consider the following code fragment:

```
void someFunc( void )
{
    char *buf;
    buf = malloc( sizeof(char) * 256 );
    // do something
    return;
    // returning from the function results
    // in the frame being removed from the stack
}
```

The variable buf is local to someFunc and its value is lost when the function returns. The value, however, is a reference to memory allocated from the heap. If the function returns without an appropriate call to free, the memory can never be returned to the heap and is an illustration of a memory leak.

Finally, in our discussion of C syntax, style and convention are mechanisms for defining constants and macros within a program.

Definitions and Macros

As discussed previously, compiler directives are a means of directing the compiler to make decisions or perform actions.

In addition to the include directive seen earlier (refer to the Excursion in the section "The printf Command," on page 91, for an introduction to compiler directives), directives exist for defining constants and macros within a program.

The define Directive

Use the define compiler directive to instruct the compiler to perform a Search and Replace for the item being defined.

For instance, defining a constant GLOBAL in a header file included by multiple files is one method of managing global variables.

```
100: #ifndef GLOBAL
101: #define GLOBAL
102: #else
103: #define GLOBAL extern
104: #endif
105: GLOBAL int lineno;
```

Recognizing that variables declared external to extend their scope to multiple files must have an actual definition, the macro GLOBAL is a graceful way to accomplish it.

The first time the compiler sees this code fragment, GLOBAL is not defined, thus the compiler directive ifndef results in true and the compiler defines GLOBAL—albeit with an empty value.

101: #define GLOBAL

After you've defined GLOBAL to an initial empty value, line 105 appears as the *actual* definition for lineno.

105: int lineno;

Subsequent inclusions of this code fragment by other source files result in the test for GLOBAL as *not defined* (ifndef), resulting in false because GLOBAL was defined with the initial inclusion as the empty value. Therefore, the compiler directive else is performed, setting GLOBAL to extern. This time, when line 105 is reached, the variable lineno is defined as external.

```
105: extern int lineno;
```

The actual definition of the variable lineno is effectively in the first source file, which included the header containing this fragment. Subsequent inclusions result in lineno being referenced as an external variable.

More examples of macros and constants are seen in the Graphics Editor project.

Conclusion

The C programming language is a vastly complex language, and it literally takes years to master.

This chapter attempted to give you an introduction to the language syntax and style through example. You should now have sufficient exposure to face the more advanced issues that follow in the text, namely, the functions made available through the X Window System environment, and ultimately the challenge of creating the Graphics Editor.

Next Steps

At this point, you should have a level of confidence in the topics discussed so far, either through experimenting with the code samples provided or through independent study.

In the next chapter, focus shifts quickly from the introductory material consisting of use of the Linux operating system, understanding programming constructs, and employing the C programming language, to applying these concepts at an advanced level as we begin looking at the pieces that comprise the X Window System.



The Pieces of X

Chapter 4

In this chapter

- Origin of the X Window System
- The Pieces of X
- Next Steps

Windowing Concepts

Origins of the X Window System

Students involved in an effort known as *Project Athena* began the X Window System at MIT in the mid 1980s. Since the initial release for commercial use, the X Window System has seen many changes.

how too pro nouns' it

The X Window System is called X for short.

Despite the similarity in names between *X Window* and *Microsoft Windows*, there is little consistency. The most notable difference is the separation that X maintains between the windowing environment and the operating system. Microsoft Windows, on the other hand, is a proprietary environment closely tied to the DOS operating system.

Many proponents of the Linux community want to adopt a naming convention that clearly illustrates the separation of the two systems. However, the course has been set, and there is little left to do beyond educating users.

Until recently, driven by a consortium of businesses such as Digital Equipment Corporation, Hewlett-Packard, Sun, IBM, and AT&T, the X Window System was given much focus to further its development and guide its evolution. With the release of revision 6, X has reached a level of maturity that will see it through the next few decades. Although the products produced at MIT form the core of the X Window System, others in the industry have emulated, ported, and furthered it. Examples include companies such as Metro X, Hummingbird, Xi Graphics, and those involved with the efforts of XFree86, which is the X Window System port used in the Linux operating system.

The Pieces of X

This chapter focuses on gaining an understanding of windowing concepts necessary to program in the X Window System environment. If you are already acquainted with client/server models, immediate graphics, window hierarchy, window clipping, and event propagation and queuing, feel free to proceed to the next chapter.

Client/Server Model

The X Window System follows a client/server model that is unique to any windowing system. The model meets one of the goals set by the authors of the X environment in that it is fully extensible.



The X Window Server is responsible for managing all resources available to a *display*.



The term *display* in X Window vernacular does not refer only to the monitor associated with a workstation. A display is everything having to do with input and output for a specific system.

A typical display is a single monitor, keyboard, and mouse. However, it can be simply a touch-type plasma screen, or, it can consist of several monitors, a graphics tablet, and a keyboard.

Whatever the hardware configuration for your workstation, the X Server is responsible for its management.

Due to the relationship of the X Server and the specific hardware comprising a workstation, X Servers are not very portable. To manage the resources available to a particular video driver or monitor properly, the X Server must be very finely tuned at the hardware level.

Clearly, generic servers exist for many devices, but they will not maximize features or capabilities as a device-specific server would.



For major platform support, manufacturers of the hardware generally author the X Server because they hold the details of the devices needed to do it properly. This is certainly the case with product lines produced by Sun Microsystems, Digital Equipment Corporation (Compaq), and Silicon Graphics.

The PC market has been the exception to this rule, because the vendors were not as interested as consumers in having X Servers for the myriad of device configurations available to this class of machine.

The manner in which you start the X Server varies slightly from system to system. Under the SYSV family of UNIX, the X Server is generally added to an *initlevel*.



Initlevel stands for initialization level as defined in the system file /etc/inittab (initialization table).

Effectively, the system is set to initialize to a specific level. A corresponding entry in the initialization table instructs the system, which proceeds to start to satisfy the desired level. If configured to start automatically, the X process will be added to an initlevel.

Under the BSD family of UNIX, an entry can be made in the /etc/rc.local file to start the server.

When the system is configured to automatically run X, the command for starting the server is xdm.

Executing the xdm command starts the X Display Manager, which enforces the user login process. After a user has successfully entered a login ID and password, xdm launches the X Server.

If the system is not configured to run X automatically, you can, after satisfying the normal UNIX login requirements, issue either the startx or xinit command to start X manually.

Once started, the X Server for the display hardware is responsible for servicing all requests made by an X client. For instance, a request by a client to draw a line or render a window is communicated to the server, which manages the hardware to satisfy the request. Also, any *event* generated by the hardware is queued for the client.

EXCURSION

The Flow of Events in the X Client/Server Model

The X Window System is entirely *event driven*. The X Server communicates events generated in the display hardware to the X application (client). Events acted on by a client in turn generate requests of the server. Because the client may need time to process an event sent by the server, event queues are maintained by the server for all clients. The client is responsible for removing the events from the queue and processing them.

Figure 4.1 illustrates the flow of events and requests in the X client/server model.

Figure 4.1

The exchange of event generation and client requests.



As the mouse cursor (pointer) is moved across the screen in Figure 4.1, the X Server communicates the motion event (PointerMotion) to the X Client. Based on the client's function, the event can be acted on or ignored. In this illustration, the X Client is drawing a line to connect the previous location of the pointer (implied by the variables prev_x and prev_y) to the current pointer location. (This is the behavior of the pencil object in the Graphics Editor.)

Notice that the X Server in Figure 4.1 entirely separates the X Client from the display hardware. An X Client that attempts to communicate directly with the display sacrifices portability.

Events exist to represent all actions possible in the display hardware managed by the X Server. Table 4.1 shows some events understood by the X Server that will be processed by the Graphics Editor.

ble 4.1 X Events Needed by the Graphics Editor			
X Event	Is Generated When		
PointerMotion	The mouse moves.		
ButtonPress	The user presses a mouse button. Fields in the event structure indi- cate which button was pressed.		
ButtonRelease	The user releases a mouse button being pressed.		
KeyPress	The user presses a button on the keyboard.		
EnterNotify	The mouse enters a window.		
LeaveNotify	The mouse leaves a window.		
Expose	A window has been obstructed in such a way that the contents of the window need to be redrawn.		

Tab

Understanding that an X Server must exist for the display in which the X Window System will run, we now introduce the X Client in more detail.

X Clients

A critical step when structuring an X Client application is establishing a connection to the X Server. (The X library call to perform this task is discussed in Chapter 6, "Components of an X Window Application.") The communication between an X Client and the X Server occurs using a standard network protocol such as TCP/IP.



An X Client is any X-based application that communicates display requests through an X Server.

I■ **Note**

Understanding the details of the network over which the client and server communicate is not necessary for creating X clients.

What is important is that a network layer exists on the system on which you are running the server and clients. Because TCP/IP is inherent to UNIX, and specifically Linux, this should never be a problem. If you feel it is, consult the Linux setup to ensure that network services are started by the system.



When running an X Client, the server and the client processes typically execute on the same system. However, because the communication between the two processes is network based, it is possible to run a client on one machine and have it displayed on another. Examples of this will be covered in Chapter 6 where we demonstrate how clients choose to which server to connect.

Table 4.2 shows several X Clients provided by the Linux operating system and a brief description of their functions.

Га	ble 4.2 Linux X Clients	
	Command	Description
	xterm	Terminal emulator for X windows
	xset	Enables user to set preferences for the display
	xmessage	Displays a window containing a message
	xman	Manual page browser
	xload	Displays a periodically updated histogram of the system load average
	xfontsel	Application to display fonts known to the X Server
	xeyes	Watches what you do and reports to the Boss
	xev	Creates a window and displays all event information sent by the server
	xedit	Simple text editor
	xdpyinfo	Utility for displaying information about an X Server
	xdm	Manages an X display by prompting for login name and password, authenticating the user, and running a "session"
	xconsole	Displays messages that are usually sent to /dev/console
	xclock	Displays time in analog or digital
	xcalc	Scientific calculator desktop accessory

4

Many X Clients exist to perform a variety of tasks ranging from image manipulation to integrated application development. Further, entire environments are provided by packages such as CDE, KDE, and GNOME. These packages determine the look and feel of a tailored X session and provide embedded clients and utilities specific to the environment. Included in these environments are X Clients to manage files and printers, access the Internet, or manage your Internet service provider dial-up.

EXCURSION

The Critical Role of the Window Manager

An X Client known as a *window manager* is key to any desktop environment. A window manager is a special purpose application that provides the capability of X applications to be moved, resized, minimized, and restored dynamically by the user.

The window manager applies *decorations* to an application that enable the user to access these features. Figure 4.2 shows the window decorations and their purpose.



By using these decorations, a window can be dynamically altered in a number of ways.

Many window managers exist for use with the X Window System. The Enlightenment window manager is shown in Figure 4.2 and is provided as part of the GNOME environment included in Red Hat's distribution of Linux. The Tab Window Manager has been released as part of the standard distribution of the X Window System.

Window managers generally provide similar functionality with common management features applied to windows. The differences lie in the *look and feel* imposed, or the style of the decorations applied to windows.



Many resources exist for becoming acquainted with the clients available under the Linux operating system. With the current focus on the structure of the X Window System and not on specific clients, realize that many clients exist to serve a variety of needs. Any need recognized as yet unanswered is a call for your contribution.

To review, the X Server manages all display resources and relays events that occur in the hardware to the clients it serves. Further, the clients process the events and generate server requests.

Requests instruct the X Server to perform tasks such as creating, positioning, coloring, destroying, rendering, drawing, and clearing windows.

Windows

The task list that the X Server understands is quite extensive. Every request made to the X Server is relative to a window within the application.



In terms of application development, invoking functions contained in the X Library libX.a generates an X Server request. (Review the section "gcc -1" in Chapter 1, page 28 for a description of function libraries.) The contents of the X Library will be discussed at length in Chapter 6.

A window is the basic element of an X Window application. By arranging multiple windows within the application, the program forms its purpose or usefulness.

Consider the xcalc application, as seen in Figure 4.3.

Every component of the xcalc application consists of a window. Every one of the buttons forming the keypad of the calculator is a window with another window acting as the display panel.

As the cursor moves through the windows of the application, it generates events and places them in a queue that the application processes. For instance, as EnterNotify events occur for the different windows, the application processes it by highlighting (increasing the border width) of the window that was entered. When the LeaveNotify event is processed, the window border returns to its original width.

The ButtonPress event is important to the xcalc application. Many things must occur within the program when a ButtonPress event is seen.



1. A request is made to the server for a change in the window colors. The application requests that the foreground color be used for the background and the background color of the window be used in the foreground. The inverted colors give the appearance of a button being pushed, as seen in Figure 4.4.





2. Another request is then made to the X Server to display in the status window the value represented by the window in which the ButtonPress occurred, as seen in Figure 4.5.



When the button is released and the ButtonRelease event is generated, the colors must be returned to their original values, as seen in Figure 4.6.

Figure 4.6 ButtonRelease event.



As demonstrated in the previous example, processing the many events that occur in the various windows of an application, and subsequently generating requests from them, is the nature of event-driven programming.

In addition to understanding the purpose of processing events communicated by the X Server, it is important to recognize the relationship between the multiple windows of an application.

Window Hierarchy

The xcalc application, as with all X Window applications, consists of a hierarchy of windows. The more sophisticated the application the greater the complexity of the hierarchy.

Window hierarchy is used to group windows based on common function. This is particularly useful when a window grouping is conditionally mapped to the screen. For instance, a file selection dialog box only appears when the user prompts to save or load a file.



The term mapping a window refers to a request for the X Server to display it or make it visible on the screen. To unmap a window means to remove it from view.

The programmer, by the parent specified to the window creation function, determines the window hierarchy of an application.

EXCURSION

The Effect of Window Parenting on Window Visibility

Window creation is covered in Chapter 6. However, every window has a parent and the relationship between a window and its parent imposes a behavior that must be understood to ensure proper management.
Specifically, if a parent window is not visible on the screen, its children (descendents) will not be visible either.

Further, any portion of a window extending beyond the dimensions of its parent will be clipped by the parent and not appear onscreen.

Figure 4.7 illustrates the concept of window clipping, as the portion of the child not fully contained within the parent window is not visible.



You can further understand the relationship between a parent and a child window by describing *window origin*.

The origin of any window is its upper-left corner. When placing a window within its parent, the position (location) specified is always relative to the parent's origin.

Examples of how to manage the placement of windows are provided in Chapter 6.

Figure 4.8 shows conceptually the relationship between the multiple windows of the xcalc application.

The toplevel window in Figure 4.8 is the window decorated by the window manager as a means of providing the application with capabilities such as resizing, minimizing, and positioning.

The toplevel window also has the responsibility of managing all the application's child windows. This responsibility includes ensuring that the windows are visible and correctly placed and that the events sent to the program are dispatched to the appropriate window.

The usefulness of several of the events dispatched was illustrated with discussions of ButtonPress, ButtonRelease, EnterNotify, and LeaveNotify. We will now consider the Expose event as shown in Table 4.1.

Expose

Processing the Expose event is crucial to X-based applications due to the *immediate graphic nature* of X.



The immediate graphic nature of X refers to the absence of an inherent mechanism for redrawing graphics contained in windows.

A primary reason for creating windows in an application is to communicate something to the user. Whether to provide the keypad associated with the xcalc application or a canvas for drawing graphical objects as in the Graphics Editor, windows communicate either textually or by employing graphics.

The contents of the windows associated with the keypad of xcalc were explicitly placed to provide the functionality of the application. Because of the immediate nature of X, no mechanism is within the environment to replace the contents of these windows if they are erased or destroyed.



Figure 4.8

archy.

The X Server does not retain a window's contents, but places them in the window when the program makes an explicit request. After this they are forgotten. If a window is unmapped or obscured by another window, its contents are lost.

Rather than attempting to retain and replace the contents of all of the windows in an application, the X Server will send an Expose event to notify the program that a redraw is necessary. Details within the Expose event structure will indicate which window requires updating, the position and dimensions of the region affected, and the number of Expose events pending in the event queue.

The immediate graphic nature of X adheres to a sound principle of software development: 90 percent of the effort should not be extended to benefit 10% of the users.

The task of maintaining and replacing the contents of all windows serviced by the X Server is daunting. Each X Client could conceivably have dozens of windows associated with the application. Further, the X Server is servicing dozens of applications, amounting to many windows. The computing power and memory required to accomplish this task by the server could make the environment unusable.

However, the application that owns the window has already assigned the initial window contents. The task of the application updating the window as necessary is trivial in comparison. For that reason, X sends the Expose event notifying the program when an update is necessary.

Next Steps

With the understanding of windowing concepts gained in this chapter, you are ready to learn how aspects requiring constant management in an X-based application can be automated through the use of objects known as *widgets*.

Chapter 5, "Widget Sets," will lead you through a discussion of the X Intrinsic Toolkit and the widget objects it manages.

Chapter 5

In this chapter

- The Power and Convenience of Using Widget Sets
- The Athena Widget Set
- Next Steps

Widget Sets

Authoring an X Client requires constant attention to many details. The relationship between using windows within the application's hierarchy, processing events in the queue (ensuring critical events are processed quickly), and anticipating and responding to user input are a few of the considerations when writing an application for the X Window environment.

Much of the management necessary within an X-based application is greatly simplified through use of the X Toolkit Intrinsics, known as *Xt*.

As Chapter 4, "Windowing Concepts," led a discussion of X at the lowest level, this chapter demonstrates the relationship between widgets and windows, and the toolkit that manages them.

If you are already familiar with the X Toolkit Intrinsics and the Athena widget set, feel free to continue to the next chapter where you will build your first X Window application.

The Power and Convenience of Using Widget Sets

All aspects of creating an X Window application can be accomplished using only the X library known as Xlib, which is the lowest layer of X library functions. Figure 5.1 shows the architecture of an application employing only Xlib.



As demonstrated in Figure 5.1, the application employs the X library to form the requests made of the server.

Specifically, the application invokes functions from the X library, which formats the request into a network call for communicating them to the server. The X Server, in turn, is responsible for translating the request to the specific devices it is responsible for managing.



By ensuring that all communication to devices targeted by an application is accomplished solely through the X Server, the application maintains maximum *portability*.

Many details accounted for in the application in Figure 5.1 are left to the imagination. As discussed in Chapter 4, elements of event management, refreshing graphics, and responding to user input add complexity to the program.

However, through use of a toolkit and a set of *widgets* the tasks common to most application are simplified. Figure 5.2 shows the relationship of an X-based application to the X libraries and toolkit.



A *widget* adheres to the principles of object-oriented methodology and is, in fact, an object.



In Figure 5.2, side relationships illustrate the association between each component of the architecture. The application in this figure is employing the X library (Xlib) as well as the X Toolkit Intrinsics (Xt) and a *widget set*.



In studying the side relationships of the components in Figure 5.2, notice Intrinsics supports the widget set and Xlib supports Intrinsics. These relationships are critical in the X environment because a widget set would be of little use when writing an X-based program without the use of Xt.

Because Xt is a toolkit, its purpose is to simplify the use of Xlib and to manage a widget set. The relationship between Xlib, Xt, and a widget set are intricate.

Consistent with Figure 5.1, however, is that all communication with the devices displaying the application are managed by the X Server in order to maintain portability.

Chapter 4 discussed X Window concepts demonstrating the hierarchy of windows within the xcalc application. Windows are the basic building block of an X application and are the primary management function of the X library.

Adding a toolkit to the X library, as X Toolkit (Xt) does, maintains the window as the basic component of an application. However, as stated earlier, the toolkit simplifies use of the X library (Xlib) and has as its primary management function *widgets*.



A *widget set* is a group of components that manage different aspects of a graphical user interface. Elements such as menus, buttons, scrollbars, and text fields are entities provided by a widget set.

As demonstrated with the xcalc application in Chapter 4, section "Windows," page 119, processing all the events pertinent to a window within an application is the responsibility of the programmer. An example is acting upon EnterNotify and LeaveNotify for modifying the border width of the window as the cursor moves through the application. Another example is managing ButtonPress and ButtonRelease for inverting the colors of the window to give the appearance of a button being pushed. A final example of an area that an Xlib programmer must address is invoking the necessary procedure for responding to the user's selection of a value or function associated with a window.

A *button* component that monitors and processes all the events pertinent to changing colors and border widths or invoking functions is provided by a widget set.

As a programmer, placing the *button* widget provided by the widget set on the interface of the application frees you from the mechanics of making a window look like a button. Because a window is still the basic component of any X application, it should be understood that the widget creates and manages a window. The actual components provided within a widget set vary based on the set being used. Two common widget sets are *Athena* and *Motif*. Widget sets, in general, provide the programmer with a variety of objects to employ for accomplishing a graphical user interface.



Because the X Toolkit supports any widget set used within an application, migrating from one widget set to another is largely an exercise in editing. In other words, the naming conventions differ between widget sets but the underlying mechanics are similar. To move from one widget set to another can be as simple as editing the names of the widgets and their *resources* to those understood by the new widget set.

Choosing between one set of widgets and another is largely a matter of determining the desired *look and feel* for the application.

Another deciding factor is the sophistication of the components provided by the widget set. Dictated by the purpose of the application, components of certain function or capability may be required within the application, forcing the selection of one widget set over another.



A *look and feel* is the appearance and layout of the application. Consistency in the appearance and position of components in an application is critical for ensuring that users develop a comfort level for using and navigating the product.



Widget *resources* are the mechanism by which the widgets remain extensible by the user.

Resources are the attributes of an application at the component (widget) level. For instance, it *may be possible* through the resources available to a widget for the user to select the font, color, and placement of a button within an application.

The caveat *may be possible* is applied because the user cannot alter any resource specified by the programmer (hard coded into the application).

Examples of specifying resources are provided in the next chapter, "Components of an X Window Application."

The Athena widget set is available as part of the standard release of X and provides basic components for addressing nearly every requirement of a graphical user interface. It lacks the sophistication of other widget sets, but this simplicity makes it an easy first widget set to learn.

Because the Athena widget set is used in the Graphics Editor project, it will be the primary focus of the remainder of this chapter.

The Athena Widget Set

As described, a *widget* is a component used to implement some aspect of a graphical user interface. Widgets account for menu components, scrollbars, text entry, push buttons, frames, and more.

Widgets are objects in the sense that they have internal structures and methods that are opaque to the programmer. Influencing these objects is done exclusively through the X Toolkit Intrinsics, or convenience routines provided by the widget set. Further, following object-oriented methodology, widgets obey a class hierarchy as imposed by their authors. Because every widget belongs to a *class*, all widgets are represented in the hierarchy for the widget set in which they belong.



A widget *class* groups widgets based on common appearance and behavior. As a hierarchy implies, there is inheritance from one class in the hierarchy to its descendents.

Widgets inherit resources (attributes) and functionality from their ancestors. Knowing a widget's position in the class hierarchy aids in determining the capabilities of the widget and the appropriate time to employ it.

Figure 5.3 shows the class hierarchy of the Athena widgets.



III ⊂ Note

The widgets within the class hierarchy shown in Figure 5.3 provided by Intrinsics are underlined, but those included as part of the Athena widget set are plain text.

Intrinsics provides several basic widgets to serve as *super-classes* for vendor specific widgets. Also, some widgets are either so general or so special in purpose that it shouldn't be necessary for them to be repeated by every vendor who wants to create a unique widget set.

Every widget set, despite the vendor, will fold into the widgets provided by Intrinsics.



A *super-class* is any widget designated as the direct ancestor for subsequent widgets. For instance, in Figure 5.3, the super-class of the Command widget is the Label widget and the super-class of the Simple widget and Composite widget is the Core widget.

Consider again the idea of inheritance as we review the Athena class hierarchy shown in Figure 5.3.

Any resource available to a widget is available to its descendants. For this reason, resources are not repeated in the structure of the descendants. Similarly, the functions or methods available in the ancestor are available for inheritance as well.

Some widgets within the hierarchy exist for no other purpose than to provide a common set of resources or functionality to their descendants. The Core widget that heads the hierarchy is an example of a widget not meant to be *instantiated* directly but instead provides features to its descendants (which you will note extends to all other widgets).



The term *instantiation* refers to a widget being created within an application. Think of it as creating an *instance* of the widget.

Resources made available to all widgets by merit of inheritance from the Core widget include colors, fonts, coordinates, width, and height. Because these attributes are necessary for all widgets, the Core class ensures that they are included at the lowest level of the hierarchy.

Functionality made available by a widget's ancestor is largely the determining factor when selecting the super-class for a new widget. Consider, as an example, the Command widget and its Label super-class. Everything the Label widget can do (namely continuously display a text string or image) is required by the Command widget. However, the Command widget extends the capabilities of the Label widget. The ButtonPress and ButtonRelease events are ignored by the Label widget, whereas the Command widget acts upon them and toggles the widget's window colors and invokes a callback function.

By inheriting all methods from its ancestors, widgets in the descending layers of the hierarchy grow in complexity.

The following sections provide a description of the widgets used in the Graphics Editor project and the resources available to each. Although there are many more widgets available from the Athena widget set, for the sake of brevity only those needed by the Graphics Editor project are discussed in this text.

The Core Widget

Not explicitly created within the Graphics Editor application, but as the root of the class hierarchy seen in Figure 5.3, the Core widget provides many characteristics important to widgets that are instantiated.



The Core widget is sometimes used as a drawing area, but otherwise it is rarely instantiated.

Table 5.1 shows the resources defined by the Core widget class.

Name	Class	Туре	Default Value
accelerators	Accelerators	AcceleratorTable	NULL
ancestorSensitive	AncestorSensitive	Boolean	True
background	Background	Pixel	XtDefaultBackground
backgroundPixmap	Pixmap	Pixmap	XtUnspecifiedPixmap
borderColor	BorderColor	Pixel	XtDefaultForeground
borderPixmap	Pixmap	Pixmap	XtUnspecifiedPixmap
borderWidth	BorderWidth	Dimension	1
colormap	Colormap	Colormap	Parent's Colormap
depth	Depth	int	Parent's Depth
destroyCallback	Callback	XtCallbackList	NULL
height	Height	Dimension	widget dependent
mappedWhenManaged	MappedWhenManaged	Boolean	True
screen	Screen	Screen	Parent's Screen
sensitive	Sensitive	Boolean	True
translations	Translations	TranslationTable	widget dependent
width	Width	Dimension	widget dependent
x	Position	Position	0
у	Position	Position	0

Table 5.1 Core Widget Resources

Table 5.2 shows the descriptions of each of the items in Table 5.1.

0.C	Boooniphono or	
Name	,	Description
accel	erators	List of <i>event to action</i> bindings to be executed by this widget when events occur in other widgets—allows for application <i>shortcuts</i>
ances	torSensitive	The state of sensitivity for a widget's ancestors—a widget is insensitive if either it or any of its ancestors are insensitive. An insensitive widget will appear as grayed-out and will not process events
backg	round	A value used to index into the widget's <i>colormap</i> to determine the back- ground color of the window associated with the widget
backg	roundPixmap	The background <i>pixmap</i> applied to the widget's window—if this resource is set, the pixmap specified is used instead of the widget's background color.
borde	rColor	A pixel value used to index the widget's colormap to determine the border color of the widget's window
borde	rPixmap	The border pixmap applied to this widget's window—if this resource is set, the pixmap specified is used instead of the border color.
borde	rWidth	The width of this widget's window border
color	map	The colormap used by the widget
depth		The depth used for this widget's window
destr	oyCallback	A function list called when the widget is destroyed
heigh	t	Specifies the height of the widget
width		Specifies the width of the widget
mappe	dWhenManaged	If True, the widget's window will automatically be mapped by the Toolkit when it is managed—otherwise, an explicit request to map the window will be necessary.

Table 5.2 Descriptions of core Widget Resources

EXCURSION

The Vernacular of an X Window Programmer

A number of terms and types referenced in the **Core** widget's resource list have not been previously introduced.

The following list is a mini-glossary to explain these new concepts.

Boolean Data type used to represent a True (defined as 1) or False (defined as 0) value

Callback A function registered with the widget by the programmer after instantiation callbacks are specific to a reason such as ButtonPress, EnterNotify, CreateNotify, DestroyNotify, and more. The function is invoked by the widget when an event corresponding to the callback reason is received. **Colormap** An array of elements defining color values—the colormap associated with a window is used to display the contents of a window; each pixel value is an index into the colormap to produce a red, green, and blue component that drives the guns of a monitor. Depending on hardware limitations, one or more colormaps can be installed at any one time.

Depth The number of bits per pixel of the window or drawable

Dimension The Dimension data type is defined within the X library for specifying width and height values for widgets.

Drawable Either a window or pixmap used as either the source or destination of a graphic operation

Pixel A value used to index into the colormap of a window for determining the actual color to be displayed

Pixmap A two-dimensional array of pixels used for displaying icons or applying texture in an X application

Screen An X display server can provide several screens, which are typically independent physical monitors. A **Screen** structure is the data type containing the information about the screens controlled by the server.

Translations A *translation* maps an event into an action name understood by the widget. Translations and actions can be added to widgets based on user needs. After a translation is installed in a widget, the named action will be invoked when the specified event occurs in the widget.

XtCallbackList A list of Callbacks registered with a widget for a given reason—all functions in the list are invoked when the corresponding event (reason) occurs in the widget, although the order in which they are invoked cannot be predicted.

EXCURSION

The ApplicationShell Widget

Returned by the Intrinsics function call that opens a connection to the X display server, the ApplicationShell widget serves as the root widget for the *instance hier-archy* of an application.



Differing from the class hierarchy shown in Figure 5.3, an application's *instance hierarchy* is dependent on the order in which the program's author creates the widgets.

An instance hierarchy indicates the relationship of the widgets created in the application and, specifically, the manner of parenting specified by the programmer. Table 5.3 shows the resources introduced by the ApplicationShell widget.

Та	ble 5.3	e 5.3 ApplicationShell Widget Resources				
	Name	Class	Туре	Default Value	Description	
	argc	Argc	int	0	Stores the value of argc passed to the function main	
	argv	argv	StringArray	NULL	Stores the values of argv passed to the function main	



The utility of X knowing the parameters made available to the program (argv) will become clear in Chapter 6 when we take a detailed look at the function used to connect to the X Server.

In brief, a standard set of parameters is accepted by all Xt-based applications. The presence of the accepted parameters is made known by providing the variables argc and argv to the server initialization call.

Table 5.4 shows the resources inherited by the ApplicationShell as a descendant of the ToplevelShell widget.

Tab	le 5.4 ToplevelShe	e11 Widget F	Resources		
	Name	Class	Туре	Default Value	Description
	iconName	IconName	String	NULL	Specifies the name to be applied to the icon when the application is minimized.
	iconNameEncoding	IconName Encoding	Atom	NULL	As X supports internationalization, it is necessary to specify how the name is encoded.
	iconic	Iconic	Boolean	False	Determines whether the program is started in the iconified or minimized state—the default value is that it is not minimized at startup.

Table 5.5 shows a few of the resources inherited by the ApplicationShell as a descendant of the Shell widget class.

ibic 0.0 Sherr Widge	71 1103001003			
Name	Class	Туре	Default Value	Description
allowShellResize	AllowShellResize	Boolean	False	Specifies whether the user will be able to use the decorations applied by the window manager to dynamically change the width and height of the application.
geometry	Geometry	String	NULL	An alternate way to specify the initial place- ment and dimensions of the application. The geometry string follows the form widthxheight+/ -xoffset+/-yoffset and is interpreted rela- tive the root window (desktop).

Table 5.5 Shell Widget Resources

Referring again to the Athena class hierarchy shown in Figure 5.3, the ApplicationShell is a descendant of the ToplevelShell widget class. As is true for all shell widgets, it is only able to parent a single child. For this reason, a class of widgets known as Composite widgets exists in order to serve as *managers* of other widgets.

Widgets That Manage Other Widgets

Two forms of manager widgets are employed in the Graphics Editor project—the Box and Form widget. Each widget has a unique approach for managing the widgets that it parents.

The Box Widget

The Box widget, the simpler of the two, provides management of an arbitrary number of widgets by containing them in a box of specified dimensions.

The children of the Box widget are rearranged when the parent is resized or when children contained by the box are managed, unmanaged, or resized. The Box widget always attempts to pack its children as tightly as possible within the geometry allowed by the parent. Box widgets are most commonly used to manage a related set of buttons, but their children are not strictly limited to being only buttons.

Table 5.6 shows the resources introduced by the Box widget.

Tabl	е 5.6 вох Wi	dget Resources	5		
	Name	Class	Туре	Default Value	Description
	hSpace	HSpace	Dimension	4	The amount of horizontal space left between each child of the Box widget
	vSpace	VSpace	Dimension	4	The amount of vertical space left between each child of the Box widget
	orientation	Orientation	Orientation	XtOrientVertical	Determines whether the children are oriented verti- cally (XtorientVertical) or horizontally (XtorientHorizontal)

The Form Widget

Like the Box widget, the Form widget can contain an arbitrary number of children; however, the Form widget provides strict geometry management of its children.

The Form's management style is to control the position of each of the children by asking the programmer to specify the relative placement of all widgets managed by the Form. When a Form widget is resized, it computes new positions and sizes of its children while attempting to maintain the relative positions specified when each child was added to the form.

The default width of the Form widget is the minimum required for holding all its children after computing their initial layout. If a width and height are assigned to the Form widget that are insufficient to house all its children, window clipping can occur.

Table 5.7 shows the resources introduced by the Form widget.

Tabl	e 5.7 Form Widg	get Resourc	es		
	Name	Class	Туре	Default Value	Description
	defaultDistance	Thickness	int	4	The default distance to use as spacing between the Form widget's children

Table 5.8 shows the resources inherited by the Form widget as a descendent of the Composite widget. These resources enable children of the Form to specify individual layout requirements.

Name	Class	Туре	Default Value	Description
bottom, left, right, top	Edge	XawEdgeType	XawRubber	Where to place the corresponding edge of the widget when the Form widget is resized
fromHoriz, fromVert	Widget	Widget	NULL	Specifies the widget this child should be placed underneath or to the right of respectively—the default value of NULL indicates that the widget should be posi- tioned relative to the corre- sponding edge of the parent
horizDist vertDistance	Thickness	int	defaultDistance (Form resource)	Sets the amount of space between this child and its horizontal or vertical neighbor respectively
resizeable	Boolean	Boolean	False	Determines whether the Form widget will ignore geometry requests made by the child—note that this doesn't prevent the Form from resizing the child.

 Table 5.8
 Composite Widget Resources

Allowable values for the bottom, left, top, and right resources include

XawChainBottom	Ensures the corresponding edge of the widget remains a fixed distance from the bottom of the Form widget
XawChainLeft	Ensures the edge of the widget remains a fixed distance from the left edge of the Form widget
XawChainRight	Ensures that the edge of the widget remains a fixed dis- tance from the right edge of the Form widget
XawChainTop	Ensures the edge remains a fixed distance from the top of the Form widget
XawRubber	Enables the corresponding edge to move a proportional distance relative to the new size of the Form widget

5

Examples of the semantics for employing the constraint resources of the Form widget are provided in Chapter 13, "Application Structure."

The final two Athena widgets to consider are the Label widget and the Command widget.

The Label Widget

The Label widget is responsible for displaying a text string or Pixmap within a rectangular region of the screen.

The label can contain multiple lines of text or a single Pixmap image. When displaying a string, the Label widget supports text left, right, or center justification. The Label widget cannot be directly selected or edited and is provided by Athena for output only.

Table 5.9 shows the resources introduced by the Label widget.

au	Ine 5.9 Laber winder uesonices					
	Name	Class	Туре	Default Value		
	bitmap	Bitmap	Pixmap	None		
	encoding	Encoding	unsigned char	XawTextEncoding8bit		
	font	Font	XFontStruct *	XtDefaultFont		
	foreground	Foreground	Pixel	XtDefaultForeground		
	internalHeight	Height	Dimension	2		
	internalWidth	Width	Dimension	4		
	justify	Justify	Justify	XtJustifyCenter		
	label	Label	String	Widget instance name		
	leftBitmap	LeftBitmap	Bitmap	None		
	resize	Resize	Boolean	True		

Table 5.9 Label Widget Resources

Table 5.10 shows descriptions of the resources introduced by the Label widget.

Table 5.10	Descriptions of Labe1 Widget Resources
Name	Description
bitmap	The Pixmap to be displayed instead of the character string—this enables the creation of icons that invoke actions.
encodir	The encoding method enables internationalization and is used to interpret the value of the label resource. The allowed values are
font	The font to use when displaying the text string specified by the label resource

foreground	The pixel value used to index into the widget's colormap to determine the foreground color of the widget's window
internalHeight	The minimum space left between the label or bitmap resource value and the vertical edges of the window associated with the widget
internalWidth	The minimum space left between the label or bitmap resource value and the horizontal edges of the window associated with the widget
justify	Specifies the justification of text displayed in the Label widget—allowable values are XtJustifyLeft, XtJustifyCenter, and XtJustifyRight.
label	Sets the character string to be displayed in the window associated with the widget
resize	Specifies whether the widget should attempt to resize to its preferred dimensions anytime its resources are modified

Last to consider is the Command widget. As described earlier, the Command widget extends the capabilities of its super-class, the Label widget.

The Command Widget

The Command widget is a rectangular area containing a character string or Pixmap image much like the Label widget; however, it accepts input for selection.

how too pro nouns' it

Although spelled *c-o-m-m-a-n-d*, it is pronounced as if it were written *button widget*.

When the mouse cursor enters a Command widget, the widget highlights by increasing the width of its window border. This highlighting indicates that the widget is ready for selection. When the left mouse button is pressed, the Command widget responds by reversing its foreground and background colors. Upon the release of the left mouse button, the colors are returned to the normal values and a notify action is invoked that will call all functions registered on the widget's callback list. If the mouse cursor leaves the widget's window before the button is released, the button reverts to its normal colors and does not invoke the notify action.

Table 5.11 shows the resources introduced by the Command widget. Remember that all resources associated with the Command widget's super-class are available as well.

u		Jot mosouroos			
	Name	Class	Туре	Default Value	
	callback	Callback	XtCallbackList	NULL	
	cornerRoundPercent	CornerRoundPercent	Dimension	25	
	highlightThickness	Thickness	Dimension	2	
	shapeStyle	ShapeStyle	ShapeStyle	Rectangle	

Table 5.11 command Widget Resources

Table 5.12 shows descriptions of the resources introduced by the Command widget.

	Similaria Wraget Hesouries
Name	Description
callback	List of routines called when the notify action is invoked
cornerRoundPercent	When a shapeStyle of roundedRectangle is specified, cornerRoundPercent determines the radius of the rounded corner. This radius is a percentage of the length of the shortest side of the Command widget.
highlightThickness	Thickness of the highlighted rectangle used to alert when the widget is ready for selection
shapeStyle	Used to create non-rectangular widgets

Table 5.12	Descriptions of	Command	Widget	Resources
------------	------------------------	---------	--------	-----------

Examples using the Athena widgets will be provided in subsequent chapters. Understand for now the functionality introduced by each of the widgets and the resources associated with them.

Other widget sets exist that are significantly more complex than the Athena. The following section introduces the Motif widget set.

The Motif Widgets

The Motif widget set is the product of the Open Software Foundation (OSF) and despite implications of the name, the widget set is not free.

Motif is an industry standard widget set that has nearly become synonymous with the X Window System. Its use is widespread in the professional development community because it is considered a serious widget set for commercial software development.

Some in the Linux community hope to eventually have available a free Motif-*like* widget set by merit of the efforts of those working on a Motif clone known as LessTif (as opposed to *Mo*-tif).

The Motif widget set is significantly more complex than Athena and is therefore more difficult to learn. Some parallels exist between the widget sets; however, Motif offers many more widgets with greater capabilities. Further complicating the Motif widget set is that many of Motif's features are controlled through *indirect* resources.



An *indirect resource* refers to a resource of one widget affecting the behavior of another.

Mastering Motif resources requires sifting through volumes of information.

To complete this brief introduction of Motif, the following section shows the similarities and differences between Motif and what you now know about the Athena widget set.

The Core and ApplicationShell Widgets

Refer to the class hierarchy shown in Figure 5.3 and notice that the Core and ApplicationShell widgets are provided by the X Toolkit Intrinsics. Because the Athena widget set class hierarchy folds into those widgets provided by Xt, the Core and ApplicationShell widgets also appear in the Motif widget set.

The other widgets introduced so far, however, are provided by Athena and will require a corresponding widget in the Motif widget set.

The XmRowColumn Widget

The XmRowColumn widget provided by Motif is closely compared to the Athena Box widget. The management styles of the two widgets are similar except that the XmRowColumn enables the programmer to specify preferences not available to the Box widget.

The XmRowColumn is used widely as a super-class within the Motif widget set, providing for menu bars, menu panes, and radio or check box buttons. Some indirect resources specific to the XmRowColumn for supporting these rolls include

- adjustLast and adjustMargin override the margin resources of Label and Button widget children.
- isAligned and entryAlignment override the alignment resources of any Label or Button widget children.

Like the Athena Box widget, the Motif XmRowColumn widget provides direct resources for controlling its layout, orientation, borders, and spacing. However, making it more extensible than the Box widget, the XmRowColumn enables the programmer to specify the number of columns for vertical orientation or number of rows for horizontal orientation in which to arrange its children. As was mentioned of the Box widget, it always packs its children as tightly as possible; the Motif XmRowColumn widget, however, allows the packing style to be set.

Consider now the Athena Form widget and the Motif equivalent.

The XmForm Widget

Both the Athena and Motif widget sets provide a Form widget for strict management of widget children. However, as was true with the Motif XmRowColumn widget, the Motif XmForm widget is significantly more complex than the Athena equivalent. The XmForm widget provided by Motif, like the Athena Form widget, enables the children to specify attachments for the edges of its widgets, but it also enables edges to be positioned to occupy a percentage of the parent window.

The XmLabel Widget

The XmLabel widget provided by Motif is equivalent to the Athena Label widget, differing only in the naming convention of its resources.

The XmPushButton Widget

The Motif XmPushButton widget, though closely related to the Athena Command widget, demonstrates the sophistication of the Motif widget set above Athena.

For instance, like the Command widget, the XmPushButton widget can contain a Pixmap image in lieu of a string; however, it can contain as many as three Pixmap images that are displayed automatically by the widget based on the appearance of certain events.

The XmPushButton understands that when the right mouse button is clicked, the widget is *armed* and therefore will display the armedPixmap if specified by the programmer. This means that a different image will display when the mouse button is clicked and when the mouse button is released. Further, a third image called the insensitivePixmap will be displayed when the widget resource sensitive is set to False.

This discussion is meant only to serve as an introduction to the existence of the Motif widget set and illustrate that it is significantly more sophisticated and therefore more complex than its neighbor Athena.

Because Motif is not freely available, and the Motif clone alternatives are not at a sufficient state to build even novel applications around them, the Graphics Editor will employ the Athena widget set for use with the X Toolkit Intrinsics.

Next Steps

With an understanding of widget concepts, resources, and the Athena class hierarchy, the discussion will advance to a step-by-step approach for creating the necessary elements of an X Window application.

Applying everything discussed thus far in the text, Chapter 6 will demonstrate the required elements for connecting to an X Server, creating widgets, and modifying resources.

Chapter 6

In this chapter

- Connecting to the X Server
- Creating the Application Interface
- Managing Windows
- Processing Events
- Summary
- Next Steps

Components of an X Window Application

Having established a sound foundation in preceding chapters, the aim now is to apply everything discussed concerning programming constructs, C language syntax, windowing concepts, X Toolkit Intrinsics, and widget sets to create an X-based application.

This chapter demonstrates the required elements of an X Window application as it leads you through connecting to the X Server and creating and modifying widgets to form a graphical user interface. If you are already acquainted with the components required to program an X Window application, feel free to proceed to the discussion of "Xlib Graphic Primitives" in Chapter 7.

The steps to create an X Window-based application read like a recipe. The following sections describe the code fragments that accomplish each step involved, with final assembly of these fragments at the end of the chapter to create a functional X Window application.

The necessary components in the creation of an X-based application include

- 1. Connecting to the X Server
- 2. Creating the application interface

Creating buttons Creating pixmap icons Assigning actions

- 3. Managing windows
- 4. Processing events

Starting at the beginning, you must establish a connection to the X Server.

Connecting to the X Server

As demonstrated in Figures 5.2 and 5.3 from Chapter 5, "Widget Sets," an X application communicates requests to the X Server through a network connection. This provides the client with absolute separation from the X Server, ensuring its portability and enabling processes to be distributed over a network.

A preliminary step in writing an X application is to establish the lines of communication with the server that will respond to the application's requests.

Several functions are available in the Xt library for establishing a connection to the X Server.



Of course, anything the X Toolkit Intrinsics (Xt) provides functions for can be accomplished by using only the Xlib or basic X library.

However, employing the Intrinsics layer of X for the conveniences it provides is common to software development. For this reason, focus throughout this chapter is on using the X Toolkit and the Athena widget set.

Chapter 7, "Xlib Graphic Primitives," demonstrates when it is unavoidable to employ only the lowest layers of X.

Two of the most common functions used to establish a connection with an X Server are XtAppInitialize and XtVaAppInitialize.

The functions are closely related except that the first requires that a preformed *resource list* is passed as a parameter and the second enables a *variable argument list*, accounting for any number of resources.

Employing Widget Resources Using Variable Argument Lists



A *resource list* is a list of widget resources and their corresponding values. The Arg data type is defined in the X Toolkit to support specifying resources and values.

```
typedef struct {
    String name;
    XtArgVal value;
} Arg;
```

The structure of the Arg data type contains two fields: one for specifying the resource name and the second for specifying the resource value.

The data type XtArgVal differs depending on the architecture where X is installed. It makes the specifications of the value field as generic as possible for portability between platforms of differing architectures.

Generally, an array of Arg elements is defined along with a counter for use with the Xt-provided macro XtSetArg. This macro is used for filling elements of the array to form the predefined resource list required by functions in the Xt library.

Listing 6.1 illustrates use of the Arg structure and the macro XtSetArg.

Listing 6.1 Arg and XtSetArg Usage

```
1: Arg args[10];
2: int argCnt;
3: argCnt = 0;
4:
5: XtSetArg( args[argCnt], XtNwidth, 220 ); argCnt++;
6: XtSetArg( args[argCnt], XtNheight, 250 ); argCnt++;
7:
8: /* args is not a predefined resource list able to
9: * be passed to functions like XtAppInitialize
10: */
```

When reading this code fragment, it is important to note use of the ++ operator on the variable argCnt. The ++ operator increments the variable by one and is equivalent to explicitly performing the following statement:

```
argCnt = argCnt + 1;
```

Also important to the use of the increment operator on the variable argCnt is that it is performed *external* to the macro XtSetArg.

As discussed in Chapter 3, "A Word on C," in the section "The define Directive" on page 108, macros operate by substitution by the compiler. Because this substitution can be quite complex, surprising results can occur if you are not careful. Consider the following example:

```
#define SQR(x) x*x
```

It is expected that using SQR in a body of code will double any number passed to the macro. For instance,

```
{
    val = SQR(5);
}
```

would be expanded by 5×5 and the value 25 assigned to the variable val. However, what would the result of the following be?

```
{
    val = SQR(5+1);
}
```

The expansion of the macro in this example is not as expected. Instead of $5+1\times5+1$ or 36, which is intended, it would be seen as $5+(1\times5)+1$ or 11 because the multiplication operator has a higher precedence than the addition operator and is therefore evaluated first.

The macro SQR in the previous example is called an *unsafe* macro. To make it *safe* would require use of parentheses to force the intended evaluation, as in

```
#define SQR(x) ((x)*(x))
```

Any macro you did not author should be treated as unsafe, and operators should not be nested in such macros. The result might not be simply an unexpected evaluation of the expression, but potentially incrementing a variable more times than expected. Improper control of variables tends to lead to segmentation violations or other equally unwelcome side effects.

EXCURSION

Understanding Argument Lists of Varying Sizes

A variable argument list is, as the name implies, an argument list of unknown or varying length. You've already witnessed use of variable argument lists in the discussion of the printf command in Chapter 3.

The printf function expects a varying number of arguments as determined by the number of substitution tokens nested in the format string passed to the function.

Similarly, Xt provides a variety of variable argument functions. These functions are consistently used for specifying resource names and resource value pairs.

As is true with the format string passed to printf, functions allowing a variable argument list can have a number of required arguments.

For the discussions in this chapter, if functions exist in the Xt library that accept a variable argument list then they will be used over functions that don't. Variable argument functions eliminate some overhead within the application and provide an elegant programming solution.

Listing 6.2 illustrates use of the XtVaAppInitialize function.

Listing 6.2 The XtVaAppInitialize Function

```
#include <stdlib.h>
1:
2: #include <stdio.h>3: #include <X11/Intrinsic.h> /* for creation routines */
4: #include <X11/StringDefs.h> /* for resource names
                                                                  */
5: #include <X11/Xaw/Form.h>
                                 /* to define the formWidgetClass */
6: {
7:
           XtAppContext appContext;
8:
           Widget toplevel, form;
9:
10:
        toplevel = XtVaAppInitialize( &appContext,
                                     "2D-Editor", /* Application class name
                                                                              */
11:
12:
                                 NULL,0, /* option list (not used)
                                                                               */
```

13:	&argc, argv, /*	command line parameters	*/
14:	NULL,	/* fallback resources	(not
14a:		used)	*/
15:	NULL	<pre>/* end of the variable</pre>	
15a:		argument list	*/
16:);		

The function performs several actions that are critical to the execution of the application.

The first parameter passes the address (&) of an XtAppContext structure I've called appContext. The XtAppContext is a structure Xt uses to maintain information associated with the application. Use of the appContext variable, once filled by the XtVaAppInitialize function, satisfies parameter requirements of subsequent Xt function calls.

From the second parameter, XtVaAppInitialize reads the class name of the application and queries the server's resource database for any values that may need to be applied to components of the application.

EXCURSION

Tailoring Widgets in an Application

To provide the extensibility required for meeting the varying needs of a large user community, widgets are highly configurable.

Configuring widgets entails changing the value of the resources available to them. As discussed in Chapter 5, resources are introduced by the individual widget and inherited based on their position in the class hierarchy.

A widget's configuration can be *hard-coded* by the programmer, in which case it is absolutely unalterable by users of the application.

Optionally, a widget's configuration can be entirely determined by the user. This requires that the user know some information, namely the instance hierarchy and the class name of the application. Remember that the source code is not always provided for determining this information. A user is usually given clues of the instance hierarchy by the presence of a default resource file provided by the author of the application.

In the absence of a default resource file, a user can employ the X tool editres for viewing the structure of Xt-based applications. The tool is useful for finding and setting resources with instant capability to view the results. Further, editres is invaluable for creating an application resource file when necessary information is not otherwise provided by the programmer.

The widget configuration (resource file) for an application can exist in several places in the X environment.

Linux commonly places a file corresponding to the class name of the application in the /usr/X11/app-defaults directory. Each line of the file contains a widget-specific resource/value pair to specify an element of the desired application configuration.

In the previous example illustrating the use of the XtVaAppInitialize function, the arbitrary class name assigned to the application is 2D-Editor. Therefore, a file of the same name could be placed in the app-defaults directory for specifying the widget configuration for the application. Because this file is read and loaded by the XtVaAppInitialize function at run-time, the configuration of the resource values is entirely at the discretion of the person who maintains the file.

Optionally, the contents of the file .Xdefaults placed in a user's home directory will override entries in the class name files found in the app-defaults directory for any X applications.

Finally, the resource values specified on the command line and passed to the XtVaAppInitialize through the argc and argv parameters are applied to the application.

All resources and their values expressed externally to an application (meaning other than the ones hard-coded) are maintained in a database within the X server known as the X Resource Manager Database. To see the values currently contained in the database for your server, use the command

bash[40]: xrdb -query

The XtVaAppInitialize function merges any options specified on the command line with those found in the X Resource Database for this application.

Table 6.1 shows the command-line options that the XtVaAppInitialize function understands. All these options are therefore available to any X applications that use XtVaAppInitialize to initialize a connection to the X server.

abie 0.1 Stanuaru At Comman		
Option	Resource File Syntax	Effect
-bg colorname	*background: colorname	Background color
-background colorname		
-bd colorname	*borderColor: colorname	Border color
-bordercolor colorname		
-bw number	*borderWidth: <i>number</i>	Border width
-borderwidth <i>number</i>		
-display displayname		X Server to use
-fg colorname	*foreground: colorname	Foreground color
-foreground colorname		
-fn <i>fontname</i>	*font: fontname	Font name
-font <i>fontname</i>		
-geometry string	*geometry: string	Size and position
-iconic	*iconic: True	Starts application iconified
-name string	*iconName: <i>string</i>	Name applied to application's icon

Table 6.1 Standard Xt Command-Line Parameters

-reverse	*reverseVideo: True	Reverse video on
- rv		
+reverseVideo	<pre>*reverseVideo: False</pre>	Reverse video off
+rv		
-synchronous	*synchronous: True	Synchronous mode on
+synchronous	*synchronous: False	Synchronous mode off
-title string	*title: string	Title applied\Title Bar provided by the window manager

EXCURSION

Uniquely Specifying Widget Paths

The syntax of the resource/value pair eligible for entry into the application's app-defaults file or a user's .Xdefaults file is in the middle column of Table 6.1.

Notice the wildcard prefacing each of the resource names and the colons separating the names from the resource values. The wildcards are optional syntax but the colons are required.

In lieu of wildcards, the *explicit widget path* referencing the placement of the widget in the instance hierarchy for the application can be used. (This information is obtainable from the **editres** client.)



An *explicit widget path* is a period-separated list of the instance names (or optionally class names) of all widgets preceding the widget in the instance hierarchy.

For instance,

```
2D-Editor.Form.Box.Command.foreground: red
```

refers to all **Command** widgets, which are contained by a **Box** widget held on a **Form** widget within the application with the class name **2D-Editor**.

Wildcards and portions of a widget's instance path can be combined as well:

```
*Command.foreground: red
```

The previous command refers to all Command widgets within the application.

Finally, as shown in the second column of Table 6.1,

*foreground: red

affects the foreground color of the entire application, which has the same effect as specifying the command-line parameter:

-foreground red

The final and most critical action XtVaAppInitialize performs is opening a connection with the X Server and creating the ApplicationShell widget.

If the function XtVaAppInitialize fails to establish a connection with the X Server, it will not create the ApplicationShell and returns NULL to indicate an error occurred. For this reason, it is important to test the result returned by XtVaAppInitialize:

Failing to test for a valid widget returned from XtVaAppInitialize results in a program crash in the instance that the connection failed.

How does XtVaAppInitialize know with which X server to seek a connection?

The function XtVaAppInitialize first considers the presence of the command-line parameter -display *displayname* for determining with which X server to establish a connection.

If the -display parameter is not specified, the presence of an environment variable called DISPLAY is consulted. If the variable is set in the environment, its value is used for determining with which server to attempt communication.

Barring the presence of the DISPLAY variable in the environment and the -display parameter on the command line, a display server on the local host is sought as a last resort.

The syntax for specifying *displayname* to the -display parameter or DISPLAY environment variable follows the syntax

```
machine:server.screen
```

where machine can be a valid hostname or IP address followed by a number indicating which server on the target machine to connect to. Specifying which server is necessary because a single host can run X servers for multiple workstations.

The field screen specifies on which monitor (in the case where the X server is supporting multiple monitors for a given display) the application should display.

Often there is only one server running a machine and it is supporting only a single monitor. A typical displayname setting is

```
machine:0
```

where the screen number is omitted and 0 refers to the only server present on the machine.

What causes an attempt by XtVaAppInitialize to connect with an X server to fail?

The two most common reasons for an attempted connection to fail are either a malformed *displayname* or access being denied, meaning that a server can deny the request because the host attempting the connection is not in the server's *access control list*.

An access control list is managed by the xhost command and determines which machines can connect with a given X Server syntax.

Access is given by using the syntax

xhost +hostname

or denied by

xhost -hostname

Enabling all connect requests to the server effectively turns off the access control list. Issuing the xhost command without any host specified does this.

xhost +

Similarly, all connect requests will be denied by issuing the command

xhost -

Having successfully established a connection to the X Server indicated by receiving a valid widget reference returned by XtVaAppInitialized, you are ready to continue creating the graphical interface of the application syntax.

Creating the Application Interface

Because one child widget is not sufficient to accomplish a significant graphical user interface, the sole child of the ApplicationShell widget is generally a widget from the Composite (manager) class such as the Form widget.



The ApplicationShell widget returned by the call to XtVaAppInitialize is the means of tying the application to the window manager. The ApplicationShell widget will be given the window manager decorations that enable the application to be resized, moved, and iconified.



Because *shell widgets* are able to parent only one child, the child will be the root of the application's instance tree.

Listing 6.3 demonstrates the Xt widget creation routine XtVaCreateManagedWidget.

Listing 6.3 The xtvaCreateManagedWidget Function

18:	<pre>form = XtVaCreateManagedWidget(</pre>	"topForm",	/*	instance name */
19:		formWidgetClass,	/*	widget class */
20:		toplevel,	/*	widget parent */
21:		NULL);	/*	terminate variable
22:				argument list */

You use the function XtVaCreateManagedWidget to create a widget of any class. As shown in the previous code snippet, the first argument is the instance name assigned to the widget.



Although *instance names* for widgets in an application need not be unique, it is more difficult to refer to them in a resource file when attempting to use an explicit widget path.

The second parameter to the function XtVaCreateManagedWidget is the class name of the widget to create. This variable is defined in the header file included specifically to employ widgets of this class. For instance, refer again to this excerpt from Listing 6.2:

```
5: #include <X11/Xaw/Form.h>
```

Header files exist to support all widget classes available within the Athena widget set.

The third argument to the function is the parent widget, which in this case is the shell returned from XtVaAppInitialize.

The final argument terminates the variable argument list.

Optionally, *hard-coded* resources could be specified for this widget, as shown in Listing 6.4.

Listing 6.4 Hard-Coding Resources

```
form = XtVaCreateManagedWidget( "topForm",
18:
                                                           /* instance name */
19:
                                      formWidgetClass,
                                                           /* widget class */
20:
                                      toplevel,
                                                           /* widget parent */
20a:
                                      XtNwidth.
                                                       220.
                                                       250
20b:
                                      XtNheight,
                                      XtNborderWidth, 5,
20c:
21:
                                                           /* terminate variable
                                      NULL );
22:
                                                              argument list */
```

Notice the addition of lines 20a–20c. Each line is a combination of a resource name and value. When resource values are specified in a program (hard-coded), changing the value in the code and recompiling is the only way to alter them.



Removing the user's ability to specify resource values for an application is riddled with pros and cons.

The list of cons includes loss of extensibility to suit a wider user base. Having user preferences always honored ensures the best display results on a greater number of platforms.

The pros extend to removing the dependency on user know-how, which can render the application unusable, and minimizing the necessary level of support and documentation to prevent user mistakes.

There is much to say in the way of personal commentary on this issue, but I will limit it to my own rule of what to hard-code and what to leave configurable. Anything I feel strongly about will get hard-coded and I feel strongly about most everything. However, the needs of your users can vary; clearly individual judgement must be applied.

Notice in Listing 6.4, the NULL termination is still required to mark the end of the variable argument list.



When specifying widgets internal to an application as shown in Listing 6.4, the naming convention of the resources varies slightly.

Compare the resources used in Listing 6.4 to the resource names specified for the Core widget in Table 5.1 of Chapter 5.

When resources are specified external to the program such as through entries in the app-defaults or the .Xdefaults file, the naming convention used is as seen in Chapter 5. However, internal to the application, you must tell Xt who has named the resource by prefacing the resource name with XtN.

Focus a moment on the semantics of the widget creation function name XtVaCreateManagedWidget. It is important to discuss the use of the word Managed in this context.

→ As described in Chapter 5, in the section "The Power and Convenience of Using Widget Sets" on page 125 a window is the basic component of X Window programming; a fact that remains true even when employing Xt and a widget set.



The term *managing a widget* refers to making the widget eligible for display. The widget will actually become visible when all its ancestors are managed and it is not obscured by other widgets.

The contrast to this is creating a widget using the Xt function XtVaCreateWidget, which expects the same parameter list as XtVaCreateManagedWidget but will not manage the widget it creates. Instead, Xt must be told explicitly to manage a widget created by XtVaCreateWidget through use of the function XtManageChild. This is useful for when the widget's appearance in the application is conditional.

For instance, a button to clear the Graphics Editor canvas could be left unmanaged until the user has drawn something and thereby necessitates the clear action.

Continuing in the construction of the application interface, the Form widget created by the call to XtVaCreateManagedWidget is capable of parenting a virtually indefinite number of child widgets. This enables the assembly of a meaningful graphical user interface.

Begin by adding a place for the drawing of objects in the Graphics Editor application.

Listing 6.5 shows a function employing the XtVaCreateManageWidget function to add a *canvas* to the application.



As is discussed in Chapter 7, when employing the drawing functions provided by X, any window will suffice as a *canvas*.

Because all widgets have an associated window, the choice of which widget from the Athena widget class to employ as the drawing canvas is arbitrary.

The Motif widget set, however, provides a widget explicitly for this purpose called the XmDrawingArea widget, which has features not available to any widget in the Athena set, as will be demonstrated shortly. Specifically, it will be necessary to inform the Athena Form widget to add specific events to its list of events that it monitors.

Listing 6.5 Graphics Editor Canvas Creation

```
void create_canvas( Widget parent )
1:
2:
     {
3:
         GxDrawArea = XtVaCreateWidget( "drawingArea",
4:
                                         formWidgetClass, parent,
5:
                                         XtNbackground,
6:
                                           WhitePixelOfScreen(XtScreen(parent)),
7:
                                         XtNtop.
                                                            XawChainTop.
8:
                                         XtNleft,
                                                            XawChainLeft,
9:
                                         XtNbottom,
                                                            XawChainBottom,
10:
                                         XtNright,
                                                            XawChainRight,
11:
                                         XtNheight,
                                                            220,
12:
                                         XtNwidth,
                                                            250,
13:
                                         NULL );
         XtAddEventHandler(
14:
               GxDrawArea,
                                                    /* widget to send events to */
14a:
               PointerMotionMask|ButtonPressMask, /* events to send
15:
                                                                                 */
                                                                                 */
16:
                                                    /* non-maskable events
               False,
               (XtEventHandler)drawAreaEventProc, /* function to call
17:
                                                                                 */
18:
               (XtPointer)NULL);
                                                    /* data to pass to function */
19: }
```

The creation of a Form widget with a call to XtVaCreateWidget is familiar from previous examples; however, in Listing 6.5 a number of resources have been specified in the variable argument list of the function call. In this example, the creation function is setting the background color, several edge *attachments*, and the width and height dimensions. The setting of the size of the Form is useful for initial values but will not be retained or enforced by the parent of the Form widget (also a Form) because the parent's management style is to honor the *relative* placement as indicated by the specified attachments.

III ⊂ Note

The semantics of the Form widget edge attachments are perhaps instantly confusing or less than intuitive; however, understanding them is imperative for properly using the Form.

Correct use of the Form widget ensures that the placement of widgets within an interface is consistently maintained by the application. Other manager widgets will rearrange (pack tightly) the child widgets they maintain, giving a less professional appearance to your interface.

The edge assignments in Listing 6.5 specify that all four edges (top, right, bottom, and left) for the child being created are to be chained (locked) to the corresponding edge of the parent.

7:	XtNtop,	XawChainTop,
8:	XtNleft,	XawChainLeft,
9:	XtNbottom,	XawChainBottom,
10:	XtNright,	XawChainRight,

The means of referring to the edge for which an attachment is being specified (XtNtop, XtNleft, XtNbottom, and XtNright) is clear from the example.

What varies is the way to specify the attachment for a given edge. An edge can be placed relative to an edge of the parent Form using the XawChainEdge, as seen in the previous example where Edge specifies what edge of the Form the placement is relative to. When chaining to an edge of the parent, the distance separating the child's edge from the parent is maintained when the application is resized, meaning the child's edge will follow the edge of the parent, keeping the relative distance assigned at creation.

When placing a child widget, remember that the resources used to express attachments are provided by the Constraint widget. In other words, the child is specifying how it wishes to be *constrained* by the parent. Therefore, the edge specifications refer to the child's edge and the values given for the child's edges are relative the parent.

Optionally, an edge can be relative to another child of the parent by using the XtNfromHoriz or XtNfromVert resources. These resources expect a reference to an *existing* sibling widget for placing the edge of the widget being created.

When a widget's edge is chained to the edge of a sibling, the relative distance between them is maintained during a resizing of the application.

Finally, an edge can be loosely placed relative to the edge of the parent or sibling using the edge value XawRubber where the distance is not maintained, but will vary.



You have not yet seen the specification of *color* with the use of the XtNbackground resource in Listing 6.5.

X provides several macros for specifying black and white colors or Pixel values, two of which are WhitePixelOfScreen and BlackPixelOfScreen. The macros require as a parameter the Screen structure, easily obtained from any existing widget with the macro XtScreen.

Color specification in X beyond black and white can be a tedious task. Therefore, an entire section in Chapter 7 is dedicated to creating and employing colors.

You have seen the creation of the GxDrawArea widget in Listing 6.5 and the widget's placement relative to the parent. Notice that the variable GxDrawArea is not declared in the code listing. As will be demonstrated shortly, the variable is declared globally for use throughout the application.

Continuing with Listing 6.5, focus on the introduction of a new Xt function XtAddEventHandler.

14:	XtAddEventHandler(
14a:	GxDrawArea, /* widget to send events to *,	/
15:	PointerMotionMask ButtonPressMask, /* events to send *,	/
16:	False, /* non-maskable events *,	/
17:	(XtEventHandler)drawAreaEventProc, /* function to call *,	/
18:	(XtPointer)NULL); /* data to pass to function *,	/

Listing 6.5 uses a call to XtAddEventHandler to register two events PointerMotion and ButtonPress with the GxDrawArea Form widget.

Widgets request only that the X Toolkit notify them of events that satisfy actions inherent to their behavior. PointerMotion and ButtonPress events are not typical for the Form widget because they don't generally respond to user input.

Using the GxDrawArea Form widget as a *canvas* for drawing the objects of our Graphics Editor, however, requires that we know when the user moves the mouse cursor through the canvas or when to click a mouse button to start or end a draw action.



The function XtAddEventHandler enables additional events to be added to the notification list of a widget.

The function expects as the first parameter the widget to which the events are to be added. Secondly, the XtAddEventHandler function expects an event mask where each bit represents one or more events. When multiple events are specified, as in this example, a *bitwise ORing* of the event masks to be added to the widget is used.

EXCURSION

Manipulating Variable Values One Bit at a Time

Bit field operations are common in X as an efficient means of combining unique values into a single variable. Elements of bit fields are called *masks* or *flags*.

Effectively, each binary position within the variable represents a unique value. A declaration such as

char fields:4;

would divide the variables fields into 4 bits. Setting a bit would represent a distinct value in the variable, as in

```
fields = (1<<3); /* sets the value of field to 1000 */
fields = (1<<2); /* sets the value of field to 0100 */
fields = (1<<1); /* sets the value of field to 0010 */
fields = (1<<0); /* sets the value of field to 0001 */</pre>
```

The left shift operator (<<) simply says to shift the 1 n times to the left (1<<n).

Using the OR operator (|) enables values to be combined.

fields = (1<<3) | (1<<1); /* sets the value of field to 1010 */

The values of PointerMotionMask and ButtonPressMask represent unique positions in the bit field of the second parameter to XtAddEventHandler for representing the PointerMotion and ButtonPress events, respectively.

Table 6.2 shows some of the X-defined event masks and the event types that they select.

Table 6.2 X Event Masks

Event Mask	Event Type
KeyPressMask	KeyPress
KeyReleaseMask	KeyRelease
ButtonPressMask	ButtonPress
OwnerGrabButtonMask	(none)
KeymapStateMask	KeymapNotify
PointerMotionMask	MotionNotify
ButtonMotionMask	MotionNotify
Button1MotionMask	MotionNotify
Button2MotionMask	MotionNotify
Button3MotionMask	MotionNotify
Button4MotionMask	MotionNotify
Button5MotionMask	MotionNotify
158 Part II *The Pieces of X*

Tab	le 6.2 Continued			
	Event Mask	Event Type		
	EnterWindowMask	EnterNotify		
	LeaveWindowMask	LeaveNotify		
	FocusChangeMask	FocusIn		
	FocusChangeMask	FocusOut		
	ExposureMask	Expose		
	ExposureMask	GraphicExpose\	MappingNotify\	UnmapNotify\
	DestroyNotify			

Some event masks in Table 6.2 select only a single event, whereas others select multiple events. Also, some event masks select the same event type. Table 6.2 also provides examples of events that have no corresponding event masks, called *nonmaskable* events.

The third parameter in the call to XtAddEventHandler informs Xt whether the event handler should be invoked for *nonmaskable* events. A nonmaskable event cannot be specifically selected because it does not have an associated mask. In Table 6.2, the example passes False for this parameter because our event handler is interested only in the events that have been registered.

The fourth parameter, the event handler function, specifies the function that Xt will call when the widget receives the events you've added to the notify list. The actual definition of the drawAreaEventProc event handler will appear later in this chapter.

The fifth and final parameter instructs Xt of *client data* to pass to the event handler function when it is invoked.



Client data is a parameter included with every function registered with Xt for invocation.

If there is data to be passed to the function, include it in the client data field and Xt will pass it to your function. If no data is desired, specifying NULL will act as a placeholder and no data will be passed to the function.

Having completed the description of creating the GxDrawArea widget and registering events that will make it a useful drawing canvas, you are ready to expand the interface to include controls for drawing actions.

Creating Buttons

Creating a place for the user to draw is an important element of the application's interface. However, the buttons used to request a drawing function are equally important.

The first consideration when creating the buttons for invoking drawing actions is where to place them.



The mental picture I hold for the Graphics Editor application is shown in Figure 6.1.



My mental image of the Graphics Editor layout.

Note



Supported by the mental image I have of the application layout, a box to contain the drawing buttons will be required. Listing 6.6 shows a function that satisfies this requirement.

Listing 6.6 Button Creation Function

```
#include <X11/Xaw/Command.h> /* for employing the Command widget */
1:
                                  /* for use of the Box widget
2:
   #include <X11/Xaw/Box.h>
                                                                       */
3:
   void create_buttons( Widget parent )
4:
5:
   {
6:
        Widget butnPanel, exitB;
7:
        /*
8:
9:
         * create a panel for the drawing icons
         */
10:
```

continues

Listing 6.6	Continued
11:	<pre>butnPanel = XtVaCreateWidget("drawButnPanel".</pre>
12:	boxWidgetClass. parent.
13:	XtNtop. XawChainTop.
14:	XtNright. XawChainRight.
15:	XtNbottom, XawChainTop,
16:	XtNleft, XawChainRight,
17:	XtNhorizDistance, 10,
18:	XtNfromHoriz, GxDrawArea,
19:	XtNhSpace, 1,
20:	XtNvSpace, 1,
21:	NULL);
22:	
23:	<pre>create_icons(butnPanel, gxDrawIcons, draw_manager);</pre>
24:	XtManageChild(butnPanel);
25:	
26:	<pre>exitB = XtVaCreateManagedWidget(" Exit ",</pre>
27:	commandWidgetClass, parent,
28:	XtNtop, XawChainBottom,
29:	XtNbottom, XawChainBottom,
30:	XtNleft, XawChainRight,
31:	XtNright, XawChainRight,
32:	XtNfromVert, butnPanel,
33:	XtNfromHoriz, GxStatusBar,
34:	NULL);
35:	
36:	XtAddCallback(exitB, XtNcallback, gx_exit, NULL);
37: }	

Again, you see the use of the XtVaCreateWidget. This time a Box widget is instantiated to contain the application's drawing buttons.

The position of the Box widget specified by its attachments is for the top and bottom edges of the widget to follow the bottom edge of the parent widget and the right and left edges to maintain a relative distance from the parent's right edge.

This placement ensures that resizing the application by moving the bottom or right resize decorations applied by the window manager will *move* the Box widget without *resizing* it because both the top and bottom edges of the widget move in unison, as do the right and left edges.

Both edge pairs (top/bottom and left/right) will retain their relative placement to the parent's edge, preventing the Box from resizing, which would ruin the aesthetics of the interface. The only resizing that is desirable is for the canvas (GxDrawArea) to grow or shrink with the ApplicationShell. Buttons bigger than the icon they hold serve no purpose, whereas a larger canvas means more room for additional drawing.

Before considering the function that actually places Command widgets (buttons) into the Box widget, consider the creation of the Exit button.

Refer again to the familiar routine for creating a widget in Listing 6.6, lines 26–34. You see some differences to identify; specifically, the Exit button is placed relative to two other widgets.

The use of XtNfromVert tells Xt that the top and bottom edge attachments are interpreted relative to the widget specified as the value of the XtNfromVert resource or the butnPanel Box widget. Similarly, the use of XtNfromHoriz instructs Xt that the left and right widget edges are relative to a widget called GxStatusBar, which you must assume has already been created for it to be used as the value of the fromHoriz resource.



The GxStatusBar variable refers to a Label widget that the application uses to display instructions or statuses to the user. Creation of the widget is reviewed in the final code listing at the end of this chapter. The GxStatusBar variable, like GxDrawArea, is global because it is used throughout the application.

Comparing Figure 6.1 with the code fragment that creates the Exit button, you will notice that the Command widget takes as its default label value the widget's instance name. (If necessary, review Command widget resources in Chapter 5, section "The Command Widget," page 139.) In other words, if you do not specifically employ the XtNlabel resource assigning a character string value for use in the label field of the Command widget, the name assigned to the widget at creation is used.

Line 36 of Listing 6.6 introduces the Xt function XtAddCallback.

```
36: XtAddCallback( exitB, XtNcallback, gx_exit, NULL );
```

The XtAddCallback function accepts four parameters. The first specifies the widget to which the callback will be registered. The second parameter instructs Xt the reason for invoking the callback function. The third is the function to invoke for the specified reason. The fourth is the client data to pass when the callback function is invoked.

Referring again to the resources available to the Command widget, the callback reason XtNcallback occurs when the user activates the widget, or clicks and releases the right mouse button while the mouse cursor is within the window associated with the widget.

We now focus on the function that places the buttons into the Box widget created in Listing 6.6.

23: create_icons(butnPanel, gxDrawIcons, draw_manager);

The function that creates the buttons for invoking drawing actions expects three parameters: the butnPanel Box widget to hold them, a gxDrawIcons pointer that contains elements necessary for creating the icons, and finally, a callback function to invoke when the button is clicked.

The gxDrawIcons pointer is a pointer to an array of elements of the data type GxIconData, defined as

```
typedef struct _gx_icon_data {
    XbmData *info;
    void (*func)(void);
    char *mesg;
} GxIconData;
```

where XbmData * is a reference to a nested structure that holds the information needed to create the Pixmap icon for each element of the GxIconData.

Two important points here are creating a button to enable the activation of a drawing function and creating an icon to represent the action, (see Figure 6.1).

Creating Pixmap Icons

To create a Pixmap icon, three unique data elements are necessary: the character representation or bits that define the icon, the width of the icon, and the height of the icon. All other parameters to the Pixmap creation function can be common.

An X library call for creating a ${\tt Pixmap}$ is ${\tt XcreatePixmapFromBitmapData}.$ It follows the form

The first parameter, *display*, is a pointer to the structure created when a connection to the X Server was established. It can be obtained from any existing widget using the Xt macro

```
XtDisplay( GxDrawArea )
```

The second parameter is generally the window in which the Pixmap will be placed. However, in our example, the window for the button has not been created (the window associated with the Command widget) so any existing window of the same depth can be used instead. A macro provided by X enables us to get the root window associated with the display where the application is being displayed, which we are assured will be an existing window; otherwise, the connection request made by XtVaAppInitialize would have failed.

DefaultRootWindow(XtDisplay(GxDrawArea))

Notice that DefaultRootWindow expects a pointer to the display structure as its only parameter, which is satisfied with the macro XtDisplay.

The third parameter passed to the XCreatePixmapFromBitmapData is the data bits representing the icon.

X provides a standard client called bitmap for creating this data. Other environments can provide superior tools for creating icons. For instance, Sun System's Common Desktop Environment includes a client called dtpaint. Similar to this is the dxpaint client, which was included with Digital Equipment Corporation's now-defunct Ultrix operating system.

The bitmap client is sufficient for creating simple icons if you are a patient individual. Figure 6.2 shows a sample of the bitmap client interface.



By setting cells within the grid defined to be a specific width and height, the icon is created.



Notice that the bitmap client does not enable specifying colors for the icon image's cells. This is because by definition a bitmap has a depth of 1. It is a map of bits, as suggested by its name. A single bit can only represent on or off.

More complex icon images can be created using clients that support the creation of a Pixmap directly. Because a Pixmap has a depth greater than 1, it is able to represent colors.

A way to understand this is to think of a monochrome monitor where the pixels of the monitor could only represent an on or off condition. Either the pixel is lit or it is black. The monochrome monitor in this context would have a depth of only 1.

Color monitors, however, are comprised of planes of bitmaps where a single cell is the alignment of the planes to express a more complex value than simply on or off.

Using this expression of color representation, the rule for determining the number of colors able to be presented on a monitor is understood. The formula is 2^{depth} (2 raised to the depth of the screen) because the screen's depth represents the number of bit planes and each bit plane is capable of representing 2 values (on and off).

Figure 6.3 shows the output of the file created when the icon in Figure 6.2 is saved.

Figure 6.3	🗐 xterm	-	
Figure 6.3 sample.xbm file con- tents.	Sterm Befrire sample_width 36 vefrire sample_bright 32 static unsigned char mample_blts[] = { tatic unsigne	0x00, 0xd0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,	

Notice that the file contains the required fields for defining a unique icon.

The XbmData data type provides a structure for storing these unique fields and satisfying the requirements for creating an icon within the Graphics Editor application.

```
typedef struct _xbm_data {
    unsigned char *bits;
    int
                  w, h;
} XbmData;
```

Listing 6.7 shows the definition of the bitmap data used to represent the various drawing icons.

Listing 6.7 Icon Definitions

```
1:
                               #define icon_static( name, bits, width, height ) \
2:
                               static XbmData name = { bits, width, height }
3:
                               /*
4:
                                     * drawing icons
                                     */
5:
                               static unsigned char line bits[] = {
6:
7:
                                            8:
                                            0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
9:
                                            10:
                                            11:
                                            0 \times 00, 0 \times 00, 0 \times 88, 0 \times 00, 0 \times 14, 0 \times 00, 0 \times 00, 0 \times 84, 0 \times 00, 0 \times 14, 0 \times 14, 0 \times 100, 0 \times 14, 0 \times 100, 0 \times 1
12:
                                            13:
                                            0x02,0x02,0x22,0x00,0x00,0x01,0x02,0x42,0x00,0x00,0x01,
14:
                                            0x04,0x42,0x00,0x80,0x00,0x04,0x42,0x00,0x80,0x00,0x02,
15:
                                            0x01,0x00,0x40,0x00,0x01,0x01,0x00,0x40,0x80,0x00,0x01,
16:
                                            0x00,0x20,0x40,0x00,0x01,0x00,0x20,0x20,0x00,0x01,0x00,
17:
                                            18:
                                            19:
                                            20:
                                            21:
                                            0 \times 00, 0 \times 00 \};
22:
                               icon_static( line_icon, line_bits, 36, 32 );
23:
24:
                              static unsigned char pencil bits[] = {
25:
                                            0 \times 00, 0 \times 
26:
                                            27:
                                            28:
                                            0 \times 00\,, 0 \times 00\,, 0 \times a6\,, 0 \times 00\,, 0 \times 00\,, 0 \times 00\,, 0 \times 00\,, 0 \times 49\,, 0 \times 00\,, 0 \times 00\,, 0 \times 00\,, 0 \times 80\,,
29:
                                            30:
                                            31:
                                            32:
                                            33:
                                            34:
                                            0x00,0x00,0x00,0x30,0x01,0x00,0x00,0x28,0xf0,0x00,0x00,0x00,
35:
                                            0x44,0x10,0x00,0x00,0x00,0x84,0x20,0x00,0x00,0x00,0x04,0x41,
                                            0x00,0x00,0x00,0x08,0x42,0x00,0x00,0x00,0x10,0x44,0x00,0x00,
36:
37:
                                            38:
                                            0 \times 00, 0 \times 00, 0 \times 00, 0 \times 00;
39:
                               icon static( pen icon, pencil bits, 36, 32 );
40:
41:
                               static unsigned char arc bits[] = {
42:
                                            0 \times 00, 0 \times 
43:
                                            0 \times 00, 0 \times 
44:
                                            0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x3f,0x00,0x00,0x00,
45:
                                            0x70,0xc0,0x01,0x00,0x00,0x0c,0x00,0x06,0x00,0x00,0x02,0x00,
46:
```

6

Listing 6.7 Continued

```
47:
                                                                 0x40,0x00,0x00,0x40,0x00,0x40,0x00,0x04,0x40,0x00,0x20,0x00,
48:
                                                                 0x04,0x80,0x00,0x20,0x00,0x1f,0x80,0x00,0x20,0x00,0x04,0x80,
                                                                 0 \times 00, 0 \times 40, 0 \times 00, 0 \times 04, 0 \times 40, 0 \times 00, 0 \times 40, 0 \times 00, 0 \times 40, 0 \times 00, 0 \times 80, 0 \times 80
49:
50:
                                                                 51:
                                                                 0x08,0x00,0x00,0x0c,0x00,0x86,0x00,0x00,0x70,0xc0,0x81,0x00,
52:
                                                                 53:
                                                                 54:
                                                                 55:
                                                                 0x00,0x00,0x00,0x00};
56:
                                               icon_static( arc_icon, arc_bits, 36, 32 );
57:
58:
                                               static unsigned char box bits[] = {
59:
                                                                 60:
                                                                 0 \times 00, 0 \times 
61:
                                                                 62:
                                                                 63:
                                                                 64:
                                                                 65:
                                                                 0×00,0×00,0×10,0×00,0×80,0×00,0×00,0×10,0×00,0×80,0×00,
66:
                                                                 67:
                                                                 68:
69:
                                                                 0x80,0x00,0x00,0x10,0x00,0x80,0xff,0xff,0x1f,0x00,0x00,
70:
                                                                 71:
                                                                 72:
                                                                 0 \times 00, 0 \times 
73:
                                                                 0 \times 00, 0 \times 00\};
74:
                                               icon static( box icon, box bits, 36, 32 );
75:
76:
                                               static unsigned char arrow bits[] = {
77:
                                                                 0 \times 00, 0 \times 
78:
                                                                 0 \times 00, 0 \times 01, 0 \times 00, 0 \times 
79:
                                                                 0 \times 00, 0 \times 00, 0 \times 80, 0 \times 02, 0 \times 00, 0 \times 00, 0 \times 00, 0 \times 40, 0 \times 04, 0 \times 00, 0 \times 
80:
                                                                 81:
                                                                 0 \times 00, 0 \times 00, 0 \times 00, 0 \times 04, 0 \times 40, 0 \times 00, 0 \times 00, 0 \times 00, 0 \times 02, 0 \times 80, 0 \times 00, 0 \times 
82:
                                                                 83:
                                                                 0x00,0x04,0x00,0x20,0x00,0x00,0x08,0x00,0xe0,0x07,0xc0,0x0f,
84:
                                                                 0x00,0x00,0x04,0x40,0x00,0x00,0x00,0x04,0x40,0x00,0x00,0x00,
85:
                                                                 0x04,0x40,0x00,0x00,0x00,0x04,0x40,0x00,0x00,0x00,0x00,0x04,0x40,
86:
                                                                 87:
                                                                 0x00,0x02,0x80,0x00,0x00,0xc0,0x01,0x00,0x07,0x00,0x00,0x00,
88:
                                                                 0 \times 00, 0 \times 
89:
                                                                 90:
                                                                 0 \times 00, 0 \times 00, 0 \times 00, 0 \times 00;
91:
                                               icon static( arr icon, arrow bits, 36, 32 );
92:
93:
                                               static unsigned char text bits[] = {
94:
                                                                 95:
                                                                 96:
```

```
97:
                                    0x1f,0x00,0x80,0x83,0x1f,0x1c,0x00,0x80,0x01,0x0f,0x18,
98:
                                    0x00,0x00,0x0f,0x00,0x00,0x00,0x00,0x0f,0x00,0x00,0x00,
99:
100:
                                    0x00,0x0f,0x00,0x00,0x00,0x00,0x0f,0x00,0x00,0x00,0x00,
101:
                                    102:
                                    103:
                                    104:
                                    0x00,0x00,0x0f,0x00,0x00,0x00,0x00,0x0f,0x00,0x00,0x00,
105:
                                    0x80,0x1f,0x00,0x00,0x00,0xe0,0x7f,0x00,0x00,0x00,0x00,
106:
                                    0 \times 00, 0 \times 
107:
                                    108:
                                    0 \times 00, 0 \times 00\};
109: icon static( text icon, text bits, 36, 32 );
```

The bulk of Listing 6.7 is the definition of the

static unsigned char bits[] = {

which defines each of the icons used to represent the draw functions.

EXCURSION

Understanding Bit-Mapped Data

The characters in the bit data represent the cells of the bitmap grouped into 16-bit entities and converted to hexadecimal representation using Binary Converted Decimal (BCD) notation, where groupings of 8 bits are converted at a time.

Consider the following example of an 8×2 bitmap where a 1 indicates that the cell is set (on).

	0	0	0	0	0	0	0	0
	0	0	0	0	1	1	1	1
The	hexad	ecimal	represe	entation	of this	would k	be	
	Øx		0					
	0	0	0	0	0	0	0	0
	0	0	0	0	1	1	1	1
	Øx		4					
or								
uns	signed	char	bits[[] = {	0x04}	;		

Focus for a moment on the following macro defined in Listing 6.7:

1: #define icon_static(name, bits, width, height) \

```
2: static XbmData name = { bits, width, height }
```

6

167

With every call to the macro a new variable is declared.

The first field passed to the macro (name) is substituted on the second line of the macro to be a variable of type XbmData.

The remaining fields of the macro are then used to fill the XbmData structure fields.

For instance,

```
109: icon_static( text_icon, text_bits, 36, 32 );
```

is equivalent to

XbmData text_icon = { text_bits, 36, 32 };

Therefore, Listing 6.7 declares and fills the following XbmData structures:

line_icon
pen_icon
arc_icon
box_icon
arr_icon
text_icon

These variables are capable of satisfying the first field of the GxIconData structure defined earlier.

```
typedef struct _gx_icon_data {
    XbmData *info;
    void (*func)(void);
    char *mesg;
} GxIconData;
```

Listing 6.8 shows the use of the XbmData variables to define the gxDrawIcons array needed for creating the buttons that will fill the Button box of Listing 6.6.

Listing 6.8 gxDrawIcons Array Declaration

```
1:
2: GxIconData gxDrawIcons[] = {
3: { &line_icon, gx_line, "Draw an elastic line..."
                                                   },
4: { &pen_icon, gx_pencil, "Draw a freestyle line..."
                                                  },
5: { &arc_icon, gx_arc, "Draw a circle..."
                                                  },
6: { &box icon, gx box,
                      "Draw a square or rectangle..." },
7: { &arr_icon, gx_arrow, "Draw an arrow..."
                                                  },
8: { &text_icon, gx_text, "Draw dynamic text..."
                                                  },
9: /*-----*/
10: /* this list MUST be NULL terminated */
11: /*----*/
12: { NULL },
13: \};
```

Looking closely at the definition of the gxDrawIcons array, you see that the array elements hold a GxIconData structure. Further, there are seven array elements defined, with the last one being set to NULL. The first six elements, however, are set as the value of the info field, a reference (&) to a XbmData structure, which satisfies the type requirement of XbmData *.

Assigning Actions

The elements then each assign as the value of the func field a function appropriate to the action implied by the icon. Shortly, I'll introduce the header file where these functions are prototyped and therefore eligible for referencing.

```
A new syntax is introduced with the func field of the GxIconData structure.
void (*func)(void);
A function is much like a variable in that it is a reference to a value. What differs
is the manner in which the value is evaluated. Clearly, with a function, the com-
piler will go to the point of entry for the function and begin evaluating its con-
tents.
This does not preclude a program from referencing the entry point (function
name) as it would any other variable.
The syntax of the func field in the GxIconData structure supports this because
it defines a function pointer (*func) that has no return type and accepts no
parameters. The program is then able to assign function entry points to this vari-
able as it does with the second field of the gxDrawIcons array element's initial-
```

The third and final field being set in the initialization of each of the array elements is the GxIconData mesg field. The mesg field is a character pointer, which is used in conjunction with the GxStatusBar described earlier for providing statuses and messages to the user.

Having understood the definition of the structures that form the gxDrawIcons array passed as the second parameter to the function create_icons in Listing 6.6, we are now able to introduce that function.

It was invoked on line 23 of Listing 6.6

ization.

23: create_icons(butnPanel, gxDrawIcons, draw_manager);

and its definition appears in Listing 6.9.

Listing 6.9 create_icons Function Definition

```
1: void create_icons( Widget parent, GxIconData *iconData,
2: void (*callback)( Widget, XtPointer, XtPointer ))
3: {
```

Listing 6.9 Continued

```
4:
        Widget btn;
5:
        Pixmap pix;
6:
7:
        while( iconData->info != NULL ) {
8:
             if( iconData->info->bits != NULL ) {
9:
                 pix = create pixmap( parent, iconData->info );
10:
                 btn = XtVaCreateManagedWidget( "",
11:
12:
                                              commandWidgetClass, parent,
13:
                                              XtNwidth, iconData->info->w + 1,
                                              XtNheight, iconData->info->h + 1,
14:
15.
                                              XtNbackgroundPixmap,
                                                                        pix,
16:
                                              XtNhighlightThickness,
                                                                        1,
17:
                                              NULL );
18:
19:
                 XtAddEventHandler( btn, EnterWindowMask, False,
20:
                                    (XtEventHandler)statusProc,
20a:
                                    (XtPointer)iconData->mesg);
21:
                 XtAddEventHandler( btn, LeaveWindowMask, False,
22:
                                    (XtEventHandler)statusProc, (XtPointer)NULL );
23:
24:
                 XtAddCallback( btn, XtNcallback,
                                 callback, (XtPointer)iconData->func );
24a:
25:
             }
            /*
26:
             * go to the next element
27:
             */
28:
29:
             iconData++;
30:
         }
31:
    }
```

Listing 6.9 presents a weighty example, reviewing everything we've discussed so far.

The function create_icons, as discussed previously, expects three parameters. The sample of Listing 6.6 satisfied them by passing the Box widget as the first parameter, the gxDrawIcons array as the second, and the function draw_manager as the third.



Although I've not introduced the function draw_manager, you should be comfortable with referring to functions as variables in the sense that they hold an entry point for interpretation.

The create_icons function parses the iconData array (passed as gxDrawIcons), looping until it discovers the NULL termination included in the array definition.

For each element found in the iconData array, a Pixmap is created based on the contents of info field stored in the array element. (Remember that the info field corresponds to the bit data, width, and height information for each of the icons created by the bitmap client.)

The create_pixmap function is shown in Listing 6.10.

Listing 6.10 create_pixmap Function Definition

```
Pixmap create pixmap( Widget w, XbmData *data )
1:
2:
   {
3:
        return( XCreatePixmapFromBitmapData( XtDisplay(w),
4:
                                              DefaultRootWindow(XtDisplay(w)),
5:
                                             (char *)data->bits,
6:
                                              data->w, data->h,
7:
                                              BlackPixelOfScreen(XtScreen(w)),
8:
                                              WhitePixelOfScreen(XtScreen(w)),
9:
                                              DefaultDepthOfScreen(XtScreen(w)));
10: }
```

Note

Use of the XCreatePixmapFromBitmapData was demonstrated earlier. Unique to this use, however, is that the specifics of the Graphics Editor application satisfy the parameter requirements. Specifically, note the use of the XbmData * for providing the icon-specific data required for creating this icon.

With a valid Pixmap named pix created from the info field of the iconData array, the create_icons function creates a Command widget setting the value of the resource XtNbackgroundPixmap to the Pixmap pix.



When the resource XtNbackgroundPixmap is employed for a Label or Command widget, the XtNlabel resource is ignored.



For this reason, no instance name is assigned to the buttons created by create_icons.

Before incrementing the array to reference the next element, seek the NULL termination with the following statement:

29: iconData++;

The create_icons function installs two event handlers and a callback function to the newly created Command widget.

EXCURSION

Referencing Array Elements with Pointer Arithmetic

The ++ operator was described earlier as incrementing a variable by 1. How, then, does it work to advance an array to the next element?

When the variable being advanced is a pointer (flagging again the dangers of pointer manipulation in C), the variable is advanced by the size of the *thing* to which it points.

In the example of iconData++, the statement is equivalent to

```
iconData = iconData + sizeof(iconData[0]);
```

The address to which *iconData* points is moved forward the distance in memory equal to the size of one element of the array.

The first element (0) is used because 1 is the minimum length of a valid array.

The first event handler monitors for the cursor entering the window associated with the widget. When the EnterNotify event occurs, the event handler statusProc function will be invoked and passed as client data to the mesg field of the iconData array.

The purpose of statusProc is to update the GxStatusBar Label widget with the message string passed. In the case of the LeaveNotify event where the client data is NULL, the content of the GxStatusBar is cleared.

The callback assigned to the Command widgets created by the create_icons routine is the same for all, namely, the draw_manager function passed to the function pointer callback.

Evaluating the declaration of the function pointer callback provides insight into how to declare a XtCallback function.

2: void (*callback)(Widget, XtPointer, XtPointer))

The function has no return type and expects three parameters.

The first parameter will be the widget for which the callback is invoked.

The second parameter is of the data type XtPointer and is the client data specified as the last parameter to the XtAddCallback function.

The actual type of the third parameter, though passed as an XtPointer, is determined by the *reason* the callback was invoked.

For instance, the Command widget has a unique data structure for providing information relating to a ButtonPress event and another for the information pertinent to the DestroyNotify event. These structures are always passed by reference to the third parameter of a callback function.



The data type XtPointer is defined in the X environment to be something similar to

typedef void *XtPointer.

because anything can legally (syntactically speaking) be referenced as a pointer to void.

It is necessary to cast the variable defined as XtPointer to a more appropriate data type before employing it.

Listing 6.11 shows the draw_manager function passed to the create_icons function as the function pointer callback.

Listing 6.11 draw_manager and drawAreaEventProc Function Definitions

```
void draw_manager( Widget w, XtPointer cd, XtPointer cbs )
1:
2: {
3:
        void (*draw_func)( XEvent * ) = (void (*)(XEvent *))cd;
4:
5:
        if( draw_func != NULL ) (*draw_func)( NULL );
            draw_mgr_func = draw_func;
6:
7:
        }
8:
9: void drawAreaEventProc( Widget w,
9a:
                            XtPointer cd, XEvent *event, Boolean flag )
10: {
11:
        if( draw_mgr_func != NULL ) (*draw_mgr_func)( event );
12: }
```

These functions will grow in complexity as more features and capabilities are added to the Graphics Editor application.

The draw_manager function demonstrates that when a Command widget is activated, the draw function is specified in the iconData array (the second field of the gxDrawIcons array) and is assigned as client data to the draw_manager callback function.

```
24: XtAddCallback( btn, XtNcallback,
24a: callback, (XtPointer)iconData->func );
```

The previous command is cast from being an XtPointer to a function pointer and then invoked, thus enabling the draw action.

The draw_manager function then assigns the draw_func function pointer to a global variable draw_mgr_func, which is visible to the drawEventProc. When the drawEventProc event handler is invoked due to a PointerMotion or ButtonPress event occurring in the window of the canvas, it is able to invoke the last draw_func

registered by the draw_manager callback function. The purpose of the drawAreaEventProc function is to continue processing the draw action instigated by the user when he activated a draw button icon.

More details will be added in future examples, but you should be able to follow the program flow to this point.

Having postponed until now the creation of the GxStatusBar Label widget and definition of the function statusProc, consider Listing 6.12.

```
Listing 6.12 GxStatusBar Creation and Management
```

```
1: #include <X11/Xaw/Label.h>
2: /*
    * create_status
3:
    */
4:
5: void create_status( Widget parent, Widget fvert )
6: {
7:
      GxStatusBar = XtVaCreateManagedWidget( "statusBar",
8:
                                             labelWidgetClass, parent,
9:
                                             XtNtop,
                                                            XawChainBottom,
10:
                                             XtNleft,
                                                             XawChainLeft,
11:
                                             XtNbottom,
                                                             XawChainBottom,
12:
                                             XtNright,
                                                             XawChainRight,
                                                             fvert,
13:
                                             XtNfromVert,
14:
                                             XtNborderWidth, 0,
15:
                                             NULL );
      setStatus( "2D-GX (c)Starry Knight Software - Ready..." );
16:
17: }
18: /*
19: * setStatus
    */
20
21: void setStatus( char *message )
22: {
23:
        XtVaSetValues( GxStatusBar, XtNlabel, message, NULL );
24: }
25: /*
25a: * statusProc
26: */
27: void statusProc( Widget w, XtPointer msg, XEvent *xe, Boolean *flag )
28: {
29:
        if( msg == NULL )
30:
            setStatus( "\0" );
31:
        else
32:
           setStatus( msg );
33: }
```

Listing 6.12 introduces three functions:

5: void create_status(Widget parent, Widget fvert)
which creates the GxStatusBar.
21: void setStatus(char *message)

which employs the Xt function XtVaSetValues for changing the resource values of widgets that have already been created. In the setStatus routine, the XtVaSetValues is used to alter the XtNlabel resource value of the GxStatusBar to be the status message passed to the function.

Finally, Listing 6.12 shows the definition of the event handler.

27: void statusProc(Widget w, XtPointer msg, XEvent *xe, Boolean *flag)

Like XtCallback functions, event handlers are invoked by Xt and therefore must have a predictable parameter list. Effectively, the functions you define as callbacks or event handlers will honor the parameter list expected by Xt.

Because we've already reviewed the parameter list associated with the XtCallback, consider now the event handler statusProc.

The first parameter is the widget in which the event occurred.

The second parameter is the client data specified to the function call XtAddEventHandler that registered the event handler.

The third argument is the actual XEvent structure definition for the event that triggered the invocation of the event handler.

Finally is a pointer to a Boolean variable that the event handler could set to False in order to prevent this event from being dispatched to other widgets that might have registered to receive it. Setting this flag to False is not recommended.

Having completed the steps of creating the application interface, adding buttons with Pixmap icons, and assigning actions to widgets, you are ready to *realize* the ApplicationShell to have it displayed on the screen and, lastly, process events.

Managing Windows

You have seen that all widgets have an associated window. A window follows rules for being displayed that are identical to displaying widgets. Specifically, for a window to be displayed, all its ancestors must be displayed, and it must not be obscured by another window.

In widget terms, the widget must be managed, as must its ancestors, and the windows associated with the widgets must be realized or *produced*. The act of creating a widget does not necessarily create its associated window. Xt must be instructed to do so for all widgets in the application.

A function exists within Xt for the purpose of realizing all the windows associated with the widgets parented by the ApplicationShell returned by the call to XtVaAppInitialize.

The function is XtRealizeWidget, which takes as its only parameter the ApplicationShell widget.

XtRealizeWidget(toplevel);

It is acceptable to create widgets after the call to XtRealizeWidget, which is usually done when you don't know in advance when the widget will be needed.

The act of realizing all the widget's windows also causes them to be *mapped*. Mapping a window means to make it eligible for display, dependent, of course, on the rules of display provided earlier.

With the windows realized and mapped, the final step is to process the events sent to the application by the X server.

Processing Events

The function for processing events in an application is XtAppMainLoop. It is invoked by specifying the XtAppContext variable filled in by the call to XtVaAppInitialize as illustrated in the following.

```
XtAppMainLoop( appContext );
```

The XtAppMainLoop function contains an infinite loop that intentionally never returns.



Invoking the function XtAppMainLoop transfers control of the application to Xt.

From this point on, Xt will be responsible for sending events to the widgets that have requested them. The widgets, in turn, process the events by invoking the callback functions, event handlers, or the widget's internal methods.

Only through dispatching events and responding to the generated requests will Xt communicate with the application.

Summary

A good deal of material was introduced in this chapter. For clarity and brevity, the code samples provided are not entirely complete. A point that should be evident to you is that many of the widget creation examples are contained in functions that were never invoked. Other details have been intentionally omitted to eliminate redundancy.

Appendix B, "Application Layout Code Listing," provides a complete code listing for laying out this phase of the Graphics Editor Project. Building the code in Appendix B will yield the image found in Figure 6.1.

The listing is arranged in logical functions contained within appropriate source code files. Further, the creation routines are linked to the function main, making this phase of the project functional. Issues such as function prototypes have been addressed as well, with the hope of making the code a solid example of good programming practice.

You are encouraged to dwell on the code listing. In doing so, you will notice that there are some slight differences (improvements) to what was introduced in this chapter.

Next Steps

The next chapter introduces the only time the lowest level of X (Xlib) *must* be invoked. The X Toolkit Intrinsics provide simpler methods for accomplishing everything X is capable of except for performing X graphic primitives.

As introduced in Chapter 7, Xlib graphic primitives are the basic drawing functions provided by X.

Chapter 7

In this chapter

- The Graphics Context
- Graphic Primitive Functions
- Next Steps

Xlib Graphic Primitives

The thorough discussion of programming with the X Toolkit Intrinsics in Chapter 6, "Components of an X Window Application," omitted one essential element necessary for approaching the Graphics Editor project. The missing piece is not addressed by any of the functions provided with Xt.

To perform the basic graphic functions of drawing lines, arcs, points, and rectangles, the X library, X11b, must be used. These basic functions, along with others for copying and for creating the graphic contexts needed to define the attributes of the operations, are known as *X Graphic Primitives*.



A *graphic primitive* is an Xlib call that sends a request to the X Server to draw a specific shape at a specific place in a window.

What's important in this statement is that graphic primitives are relative windows and only available from Xlib.

For efficiency, Xlib functions for performing graphic primitives do not include much information. Instead, attributes of the draw request such as foreground and background color, fill styles, line widths, textures, and more, are stored in a Graphics Context structure known as the GC.

The Graphics Context

A Graphics Context (GC) structure reference is included in the parameter list of *every* Xlib graphic primitive function call. It is as important to the request as the window in which the operation will be performed.

A Graphics Context can be created using either Xlib or Xt and serves the purpose of defining all the attributes of the draw request.

180 Part II The Pieces of X

Focusing on the Xlib call for creating a GC, consider the following function prototype:

GC XCreateGC(Display *, Window, XtGCMask, XGCValues)



The Xt function for creating a GC is very similar to the Xlib call except that the Xt call ensures that GCs are shared within the application.

Depending on the role of the application and the number of GCs being created, this may be important, as there is a certain amount of overhead associated with the GC. Creating too many GCs in an application may degrade the program's performance. What qualifies as *too many*, of course, is system dependent.

Because the Graphics Editor application employs only one GC at a time and the attributes of that GC vary with every draw request, it is better suited to use the Xlib call for creating GCs.

The XCreateGC function that Xlib provides creates a unique GC for defining a draw request's attributes.

The first parameter is a pointer to the Display structure created by the call to XtVaAppInitialize (or equivalent). The Display pointer can be obtained from any existing widget using the macro

XtDisplay(widget)

The second parameter refers to a window. Because every widget has a window, the macro

XtWindow(widget)

can be used to obtain a widget's associated window.

Warning

A word of caution...

When specifying a window value for the creation of a GC, the GC returned by the creation can only be used on that window or on windows of the same depth.

The caveat only applies to display servers using multiple monitors, because it is impossible to create windows of varying depth on the same screen.

The third parameter is the *value mask* of the fields you've set in the XGCValues structure reference passed as the fourth and last parameter.

EXCURSION

Streamlining Field References by the Use of a Value Mask

The use of a *value mask* in X is very common. As shown in Chapter 6, a *mask* is a method of using a single variable to represent multiple values by setting specific bits within bit fields. (See Chapter 6, in the section "Creating the Application Interface," page 151.)

For the third parameter of the XCreateGC function, the mask informs the X Server which fields to use from the XGCValues structure and which can be ignored.

Therefore, for fields in which an assignment is made in the XGCValues structure, the corresponding value mask must be ORed () together and passed as the value mask to the creation routine. Values not set in the XGCValues structure will be set to their default value.

Listing 7.1 shows the definition of the XGCValues structure and the corresponding value mask to signify when value is set as indicated in the comment adjacent to each field.

Listing 7.1 XGCValues Structure

1:	typedef struct {	/* value mask	*/
2:	int function;	/* GCFunction	*/
3:	unsigned long plane_mask;	/* GCPlaneMask	*/
4:	unsigned long foreground;	/* GCForeground	*/
5:	unsigned long background;	/* GCBackground	*/
6:	<pre>int line_width;</pre>	/* GCLineWidth	*/
7:	<pre>int line_style;</pre>	/* GCLineStyle	*/
8:	<pre>int cap_style;</pre>	/* GCCapStyle	*/
9:	<pre>int join_style;</pre>	/* GCJoinStyle	*/
10:	<pre>int fill_style;</pre>	/* GCFillStyle	*/
11:	<pre>int fill_rule;</pre>	/* GCFillRule	*/
12:	<pre>int arc_mode;</pre>	/* GCArcMode	*/
13:	Pixmap tile;	/* GCTile	*/
14:	Pixmap stipple;	/* GCStipple	*/
15:	<pre>int ts_x_origin;</pre>	/* GCTileStipXOrigin	*/
16:	<pre>int ts_y_origin;</pre>	/* GCTileStipYOrigin	*/
17:	Font font;	/* GCFont	*/
18:	int subwindow_mode;	/* GCSubwindowMode	*/
19:	Bool graphics_exposures;	<pre>/* GCGraphicExposures</pre>	*/
20:	<pre>int clip_x_origin;</pre>	/* GCClipXOrigin	*/
21:	<pre>int clip_y_origin;</pre>	/* GCClipYOrigin	*/
22:	Pixmap clip_mask;	/* GCClipMask	*/
23:	<pre>int dash_offset;</pre>	/* GCDashOffset	*/
24:	char dashes;	/* GCDasheList	*/
25:	<pre>} XGCValues;</pre>		

In the following sections, the fields of the XGCValues structure used in the Graphics Editor will be discussed with illustrations of valid field assignments.

The GC Function

The XGCValues function field has several possible values defined by Xlib. This field specifies how the draw request is accomplished, or how the pixels are drawn in the destination window.



Although I keep using the term window when referring to graphic primitive functions, the correct term is drawable.



A drawable is either a Window or a Pixmap.

The utility of drawing in a Window should be immediately evident; however, equally important is the ability to perform graphic operations against a Pixmap.

Because Pixmaps differ from Windows only in that they aren't capable of requesting and receiving events, they are frequently used as off-screen buffers for graphic operations. Anything drawn in an off-screen Pixmap cannot be affected by Expose events or other Windows obscuring their contents, as is the case with Windows.

Therefore, the contents of the Pixmap can be transferred to a Window in response to the Window receiving an Expose event indicating that a graphic refresh is necessary. This illustrates that both Windows and Pixmaps qualify as a Drawable, which is the required second parameter of Xlib graphic primitive function calls.

Table 7.1 shows the possible XGCValues function field values and a description of the type of operation each performs.



All the GC functions are specified in terms of the value of the source foreground color and the destination pixel value.

In other words, if GXand is specified as the GC function, a request to draw a line sets the pixels (points) in the source covered by the line to the foreground color of the GC. A Boolean AND operation is then applied between the pixels in the source and the current value of the corresponding pixels in the destination to determine the new state of the destination pixels.

Tab	able 7.1 gc Function Types				
	Function	Description			
	GXclear	Clears destination pixels where corresponding pixels in the source are set.			
	GXand	If source pixels and destination pixels are set, the destination pixels remain set.			

7

Function	Description
GXandReverse	If source pixels are set and not the destination pixels then set destination pixels.
GXcopy	Any pixels set in the source will be set in the destination.
GXandInvert	Pixels not set in the source and set in the destination result in destination pixels being set.
GXnoop	Destination pixels left unchanged.
GXxor	A Boolean EXCLUSIVE OR operation is performed between source and destination pixels. This function is used to create a rubber banding effect.
GXor	Source pixels or destination pixels result in the destination pixels being set.
GXnor	If the source pixels are not set and the destination pixels are not set, the pixels will be set in the destination.
GXequiv	The source pixels are inverted and EXCLUSIVE ORed with the destination pixels to determine the state of the destination pixels.
GXinvert	The destination pixels are inverted.
GXorReverse	The source pixels or the inverted destination pixels result in pixels being set in the destination.
GXcopyInverted	The source pixels are inverted and copied to the destination.
GXorInverted	The source pixels are inverted and 0Red with the destination pixels.
GXnand	If the source pixels are not set or the destination pixels are not set, the destina- tion pixels will be set.
GXset	All corresponding pixels from the source are set in the destination.

The default value for the GC function is GXcopy. To create a GC using a value other than the default would require the declaration and manipulation of a XGCValues structure as shown in the following code fragment:

```
GC gc;
XGCValues values;
XtGCMask value_mask = 0; /* clear the mask */
values.function = GXxor; /* to peform rubber banding */
value_mask |= GCFunction;
gc = XCreateGC( XtDisplay(GxDrawArea), /* pointer to Display */
XtWindow(GxDrawArea), /* window reference */
value_mask, /* indicate fields set in values */
&values ); /* value structure */
```

}

{



The assignment to value_mask in this previous example uses a syntax not yet introduced.

```
value mask |= GCFunction;
```

Combining an operator such as OR (|) with the assignment operator (=) is common in C syntax. These combined operations are semantically equivalent to

value_mask = value_mask | GCFunction;

Because the GCFunction mask is the only flag assigned (ORed) to the value_mask variable, the X Server will create a unique GC using default values for all fields of the XGCValues structure except the function field, which is set to the GXxor function.

The next elements of the XGCValues structure important to the Graphics Editor project are the foreground and background fields.

GCForeground and GCBackground

Creating and managing colors in X should be an independent science, or at least require a license in wizardry.

To specify a color in X, a pixel value is used. The pixel value is effectively (simply stated) an index into a *colormap*.



A *colormap* is an array of color cells actively being used by the X Server. When a color is no longer referenced by any X Client, the color cell in the colormap is made available for allocation by other clients.

Figure 7.1 shows a sample colormap obtained using the Linux X client gimp.

Figure 7.1	
Sample colormap.	

The size of the colormap array, as alluded to in the Note following Figure 6.2 in Chapter 6, in the section "Creating the Application Interface," page 151, equates to the number of colors available to the display as calculated by the function

max colors = 2^{depth of screen}.

Xlib provides several functions for allocating a color cell in a colormap. Color allocation can be performed by specifying a color *name* or an RGB (red, green, and blue) color component.

EXCURSION

A Closer Look at Color Management

A closer look at color management in the XA display server automatically creates a default colormap for use by all X clients it serves. However, based on the requirements of the client, there might not be a sufficient number of color cells available in the shared colormap. When this happens, an application is free to create a private colormap.



Only one *colormap* can be active in a server at a time.

Window focus then determines which colormap is active in the server at any given moment. A client with a private colormap gaining focus will have its colormap *installed* by the server.

This causes other applications running on the desktop to display in colors that might not be aesthetically pleasing or even make sense, because the pixel values referenced by a client using the default colormap are then interpreted relative to the private colormap.

For instance, the default colormap shown in Figure 7.1 has at index 0 the color red. An application creating a private colormap can allocate in its first cell the color black.

When the private colormap is active in the server, all applications employing the pixel value of 0 will display black because that is the color corresponding to index 0 in the *active* colormap.

Returning focus to a client using the default colormap returns the default colormap as active in the server and index 0 again refers to red.

Performing color allocation by specifying *named* colors is *generally* the best way to ensure that your X application obtains a valid color value.



The caveat *generally* is applied because the internal methods for color allocation are server dependent.

If all the color cells in the default colormap are filled by the applications using it, the method of satisfying subsequent allocation requests or performing a closest match algorithm differs from server to server.

The Xlib function for allocating a color by specifying a color name is XAllocNamedColor.

int XAllocNamedColor(display, colormap, color name, exact_color, closest_color)

The function expects as its first parameter a pointer to the Display structure created when a connection to the X Server was established.

The second parameter refers to the colormap where the color will be searched for to determine whether it has already been loaded. If it is not found in the colormap, and if there are empty color cells available for allocation, the color will be loaded.

The third parameter is the name of the color to be allocated.



Color names are not arbitrary. A file called rgb.txt exists on every system for specifying the named colors available to the X Server. The file also contains the RGB values that are used to create the color when it is requested.

The location of the rgb.txt file varies between systems, but is found in the directory /usr/lib/X11 under the Linux operating system.

The last two parameters required by XAllocNamedColor are references to XColor structures, which the function will fill to indicate the *exact* match for the color requested and the *closest* match.

The XColor structure is defined as

The function returns a non-zero value to indicate that the named color was found in the rgb.txt file, and zero to indicate that it wasn't.

The X Server cannot allocate a *named color* not found in the rgb.txt file because the RGB components will be unknown.



{

If the color is found in the rgb.txt file, the *exact_color* XColor structure filled by XAllocNamedColor will hold the RGB values found in the file.

The *closest_color* XColor structure is the one safely employed by the X application.

Consider the following code fragment for demonstrating the allocation and use of a named color:

```
XColor closest, exact;
int status;
GC gc;
XGCValues values;
XtGCMask value_mask = 0; /* clear the mask */
status = XAllocNamedColor( XtDisplay(GxDrawArea),
DefaultColormap(XtDisplay(GxDrawArea),
```

7

```
DefaultScreen(XtDisplay(GxDrawArea))),
                                        "lightblue",
        &exact, &closest );
if( status == 0 ) {
    printf( "failed to alloc color" );
    return; /* nothing more can be done */
}
values.foreground = closest.pixel; /* index where match was found/created */
values mask = GCForeground;
                                                                        */
gc = XCreateGC( XtDisplay(GxDrawArea), /* pointer to Display
                XtWindow(GxDrawArea), /* window reference
                                                                         */
                value_mask, /* indicate fields set in values
                                                                        */
                                      /* value structure
                                                                        */
               &values );
```

}

Notice in the previous example that the default colormap can be obtained using the macro

```
DefaultColormap(XtDisplay( widget ), DefaultScreen(XtDisplay( widget )))
```

which expects two parameters, the Display pointer and a screen number obtained with the macro DefaultScreen.

Also, note that the pixel field of the *closest* XColor structure is the valid reference to the color allocated by the function call.

The next field from the XGCValues structure to consider is the attribute for setting the line width of a draw function.

GCLineWidth

The simplest XGCValues field to discuss is the line_width field.

As implied by the name, this field sets the width of lines drawn by the graphic primitive. If the value of line_width is 0 (the default), the fastest method known by the server for drawing a line of a pixel width of 1 is used.

The following example illustrates setting the line width for a graphics context:

The final XGCValues field used in the Graphics Editor project is the tile field.

GCTile

The tile field of the XGCValues structure assigns a Pixmap for use by the GC function as the source Drawable for graphic operations.

When a background pixmap of the destination Drawable is used as the tile value, the operation effectively performs an erase.

The effect of tiling is to use the bits set in the tile pixmap that correspond to a specified graphic function for placement in the destination.

Because this is the method of removing objects from the canvas in the Graphics Editor application, the subject will be visited again in more detail in Chapter 16, "Object Manipulation," in the section "Deleting an Object" page 321.

A complete discussion of the Graphics Context value structure involves much more than the elements used in the Graphics Editor. Table 7.2 shows a summary of the fields and their allowable values for reference.

Field	Values	Description
function	see Table 7.1	Logical operation for performing graphic primitive requests
plane_mask	unsigned long	Specifies in which planes the primitive should be performed
foreground	Pixel	Index into the colormap
background	Pixel	Index into the colormap
line_width	int	Thickness of lines
line_style	LineSolid, LineOnOffDash, LineDoubleDash	Style of lines
cap_style	CapNotLast, CapButt, CapRound, CapProjecting	Manner in which the corners of line and rectangle primitives are drawn
join_style	JoinMiter,JoinRound, JoinBevel	Method of joining two lines
fill_style	FillSolid, FillTiled, FillStippled, FillOpaqueStippled	Method of performing fill primitive request
fill_rule	EvenOddRule, WindingRule	Rule for performing fill primitive request when the points of object being drawn are not ordered
arc_mode	ArcChord, ArcPieSlice	Describes whether the end points of an arc connect to each other (ArcChord) or to the center of the arc that defines them (ArcPieSlice).

Table 7.2 XGCValues Field Summary

Field	Values	Description
tile	Pixmap	Pixmap for tiling operations
stipple	Pixmap (depth of 1)	Applies texture to primitive requests
ts_x_origin		
tx_y_origin	int	Offset of tile or stipple pixmap in the destination Drawable
font	Font	Specifies the font to use in XDrawString primitive requests
subwindow_mod	eClipByChildren,	Controls whether subwindows obscure their parent
	IncludeInferiors	
graphic	True, False	Specifies whether an Expose event should be generated
_exposures		by the primitive request
clip_x_origin		
clip_y_origin	int	Sets offset of clip region in the destination Drawable
clip_mask	Pixmap	Applies alternative clipping Drawable for limiting effect of
		primitive requests
dash_offset	int	Offset of dashes field in the destination Drawable
dashes	char	Pattern to apply to dashed lines

A number of *convenience routines* exist in Xlib for altering the values of a GC after its creation. Some of these routines include XSetForeground, XSetBackground, and XSetFunction. They will be introduced in the context of the Graphics Editor application in later chapters.

Knowing how to specify the values desired for a graphics context and the Xlib method of creating them, let us shift our focus to the functions that employ GCs.

Graphic Primitive Functions

The graphic primitives provided by X for requesting the drawing of geometric shapes (using the attributes specified in the Graphics Context) all operate in an integer pixel coordinate system with the origin at the upper-left corner of the Drawable.

No transformations are performed by X beyond the clipping performed at the edge of the Drawable, and optionally requested by the clip_mask being defined in the GC.

The smallest entity that can be drawn using an X graphic primitive is a *point*.

XDrawPoint

The Xlib graphic primitive for drawing a single point to the screen is XDrawPoint.

The first three parameters the XDrawPoint function expects should be familiar to you already.

They are

XDrawPoint(display, drawable, gc, x, y)

a pointer to the Display structure, the destination Drawable, and a Graphics Context.

Unique to the XDrawPoint are the final two parameters used to specify a coordinate. Remember that the origin of a coordinate system is the upper-left corner of the Drawable with the X-Axis increasing to the right and the Y-Axis increasing downward as demonstrated in Figure 7.2.



The XDrawPoint graphic primitive draws a *single* point at the specified x, y location using the foreground color of the Graphics Context.

To draw multiple points, an array of XPoint structures is defined and passed to the function XDrawPoints.

An XPoint structure is defined as

```
typedef struct {
    short x, y;
  } Xpoint;
```

which is simply used to hold a single (x, y) coordinate.

The function XDrawPoints expects, in conjunction with the array of XPoint structures, an integer specifying the length of the array and a *mode* for interpreting the points as seen in the following:

```
XDrawPoints(display, drawable, gc, points, npoints, mode)
```

The value of the mode parameter can be either CoordModeOrigin or CoordModePrevious.

If CoordModeOrigin is used, each point of the array is interpreted as relative to the origin or the Drawable. CoordModePrevious instructs XDrawPoints to treat each consecutive point in the array as a *distance* from the previous point. This means that the first point is placed at the specified (x, y) relative to the upper-left corner of the window and consecutive (x, y) values are used as a distance from the previous point.

The graphic primitive for connecting two points to form a line is XDrawLine.

XDrawLine

The XDrawLine function, like all other graphic primitives, expects as the first three parameters the Display pointer, Drawable, and Graphics Context to be used for the operation.

In addition to these parameters, the XDrawLine function needs the endpoints that define the line to be drawn.

XDrawLine(display, drawable, gc, x1, y1, x2, y2)

The XDrawLine function uses the foreground of the GC to draw a line connecting the two points specified.

To draw a *polyline* (multiple-line segments), X provides the graphic primitive XDrawLines.

The XDrawLines function parameter list is identical to the XDrawPoints function; however, the XDrawLines function will connect the points specified in the points array.

XDrawLines(display, drawable, gc, points, num points, mode)

To request that predefined primitives be drawn, X provides the functions XDrawRectangle and XDrawArc.

XDrawRectangle

The Xlib graphic primitive XDrawRectangle has as its unique parameters (parameters beyond the Display pointer, Drawable, and GC) an x, y location specifying the upperleft corner of the object and the width and height to apply to the rectangle being drawn.

XDrawRectangle(Display, drawable, gc, x, y, width, height)

As with other graphic primitives, multiple rectangles can be drawn with a single request by creating an array of XRectangle structures and employing the function XDrawRectangles.

XDrawRectangles(display, drawable, gc, rectangles, num rectangles)

XDrawArc

The XDrawArc function works by specifying a *rectangle* to hold the arc and setting a start and stop angle.

XDrawArc(display,drawable,gc, x, y, width, height, angle1, angle2)

The rectangle defined is not actually drawn by the request. Instead, it is used to position the arc (x, y) and determine the arc's aspect ratio (proportion of width to height).

The *width* value is translated by the XDrawArc request as the x-axis diameter of the arc and the height value is tsranslated as the y-axis diameter. This enables the drawing of ellipses (width and height values are not equal) as well as circles (values are equal).

The angles passed to XDrawArc instruct the function of the starting and ending points of the arc extent. Because the unit of measure X expects angles expressed in is (degrees/64), a full 360-degree arc is requested by setting the value of *angle1* to 0×64 and *angle2* to 360×64.



Because the value 360 is meaningful to programmers, the angle is generally nested in the XDrawArc function as (360×64).

If you actually pull out a calculator and determine that 360×64 equals 23,040 and use this value in the call to XDrawArc, those following behind you must reverse the calculation to understand that the intent was for 360×64 .

If an arc less than 360 degrees is desired, two decisions must be made: Where will the arc begin and where will it end?

To express the starting point of an arc, *angle1* determines (in degrees×64) the distance from the 3 o'clock position of the bounding rectangle XDrawArc should begin drawing.

The extent of the arc (end point) is expressed by *angle2* as (degrees×64) relative to the starting angle.

Figure 7.3 illustrates the use of *angle1* and *angle2* to form a draw request using XDrawArc.

As shown in Figure 7.3 angle1 determines the start point of the arc and angle2 the distance from angle1.

Creating arcs and specifying start and end angles will be reviewed again in the context of the Graphics Editor project with the introduction to the gxArc object.



The last thing to consider with the introduction of Xlib graphic primitives is the manner in which draw requests are formed for filled objects.

Filled Graphics

For nearly every graphic *XDraw* function introduced in this chapter there is a corresponding *XFill*, which accepts an identical parameter list but produces a *filled* graphic object.

Table 7.3 shows the match of XDraw to XFill functions provided by Xlib.

lai	ble 7.3 Xlib Graphic Primitive Fill Functions		
	Draw Function	Fill Function	
	XDrawPoint, XDrawPoints	(none)	
	XDrawLine, XDrawLines	XFillPolygon (see exception 1)	- 67
	XDrawRectangle, XDrawRectangles	XFillRectangle, XFillRectangles	
	XDrawArc, XDrawArcs	XFillArc, XFillArcs	

The exception noted in Table 7.3 refers to the fact that the XFillPolygon parameter list does not exactly mirror the XDrawLines function.

The prototype for XFillPolygon is

XFillPolygon(display, drawable, gc, points, npoints, shape, mode)

where the added shape parameter is one of the following:

- **Complex**—The polygon can have any *shape* and the path defined by the points can intersect. In other words, the polygon overlaps itself similar to a *bowtie*.
- **Nonconvex**—The polygon cans be *concave* and the paths defined by the points do not intersect (overlap).
- **Convex**—The polygon must be convex. This is the least general case, but it is generally the most efficient for the X server to render.
Next Steps

This chapter provided a whirlwind introduction to the Xlib graphic primitives that will be used in later chapters to accomplish the drawing of objects supported by the Graphics Editor application.

Specific examples and more detail are provided when the functions are introduced in the context of authoring the application.

The next important step before employing the graphic primitives within the editor application is to review bitmap versus vector graphics.

The X Window System is a bitmapped-based graphic system. The Graphics Editor, however, is vector based. Reviewing both methods makes it clear how they cans coexist between the application and the windowing environment.



Back to School

Chapter 8

In this chapter

- Vector Graphics
- Raster Graphics
- Next Steps

Vector Versus Raster Graphics

Concepts of both raster- and vector-based graphics are important for authoring the Graphics Editor Project and employing the X Window System.

This chapter provides a description of each, showing their similarities, differences, and importance to window and graphics programming.

The X Window System, as introduced over the past few chapters, expects every graphics request to include all the details necessary for performing the task. The required level of detail is obtained by X requiring that the Window and GC structures be passed to every Xlib Graphic Primitive function call.

The information contained in the graphics request is sent through a process known as the *graphics pipeline*.

The graphics pipeline determines which pixels in the destination Drawable will be set based on the request made and its associated parameters. Elements of a request that influence the determination include the nature of the primitive (line, arc, rectangle, and so forth) and attributes such as fill, stipple, clip mask, line width, line style, foreground, background, and so on.

The X server evaluates the request to determine the state of the pixels in the source, then based on the logical function specified in the GC, the source is *applied* to the destination. Finally, clipping is performed based on the dimensions of the destination and any clip region specified in the GC.

Figure 8.1 demonstrates the graphics pipeline process for drawing a line.



The request shown in Figure 8.1 is to connect the points between (10,10) and (10,20). The first step in the graphics pipeline is to determine all the source pixels that should be *set* based on the request. This phase of the process incorporates elements of the GC, such as the stipple, dashes, line style, and line width. If the line width were 4, for instance, four adjacent pixels would be affected by the request. Similarly, if a dashed line were requested, not all of the consecutive pixels between the points would be set, but only those representing the dash pattern specified.

The actual value of the pixels determined to be set depends on the foreground, background, and possibly tile values specified in the GC.

The next step of the pipeline is to apply the source pixels to the destination using the logical function specified in the GC.

Finally, any necessary clipping is performed either to bring the graphics request to the bounds of the Drawable destination or to honor any clip mask specified within the GC.

The purpose of applying the graphics pipeline in response to a graphics request is to determine the final state of the pixels in the destination. These pixel values are used as indexes into a *color look up table* (CLUT) to determine the values needed to drive the monitor (CRT).

The description of Figure 8.2 and the X Server's response to the graphics request leads to the two following concepts:

- Initial requests made of the X server using the Xlib Graphic Primitives covered in Chapter 7, "Xlib Graphic Primitives," are *vector based*.
- The graphics pipeline, however, converts the results of the Xlib primitive functions into a raster (bitmapped) image that can then be mapped to the screen.

Vector Graphics

Vector-based graphics employ a process of creating images using functions based on mathematical statements that place shapes precisely within a given space of the window.



In physics, a *vector* is the representation of both a quantity and a direction.

Functions that create an image in vector-based graphics employ a series of vector specifications such as the Xlib primitive XDrawLines function, which specifies a number of vertices to be connected.



A vertex (singular of vertices) is a point that marks the junction of two line segments.

Like popular graphic programs such as Adobe Illustrator, the Graphics Editor is entirely vector based, meaning the graphic objects created and altered by the program will consist of *geometric descriptions*.

These descriptions will be retained in terms of origins, vertices, angles, dimensions (width and height), and bounds as is pertinent to a specific shape. In fact, geometric descriptions specific to the shapes supported by the Graphics Editor form the graphic object definitions.



The benefit of maintaining graphics as *vectors* is the flexibility of manipulating them with minimal data loss.

Vector-based images can be scaled, rotated, and transposed without greatly affecting the quality of the image. If you have ever resized a bitmapped image or font, you might have noticed the deterioration in the image quality as the vertices (not explicitly defined) do not continue to connect, giving the image a jagged appearance. Vector-based images, however, retain their original quality during such transformations. Further, vector-based graphics require that less data be stored to represent them because geometric descriptions are all that is required to reproduce them.

Chapters 9–11 review the mathematics necessary for altering vector graphic descriptions to accomplish various forms of object manipulation.

As stated earlier, at some point a vector-based image must be converted into a raster image for mapping to the display.

Raster Graphics

Raster graphics, often called bitmapped graphics, are images described by pixel values that can be directly mapped to the screen.

A *raster* is a grid of x and y coordinates describing either an entire display or some portion of it. A raster-based image identifies which of these coordinates to illuminate based on the pixel value stored in the corresponding cell of the grid (see Figure 8.2).

The amount of information required to represent a raster image is significantly greater than that required to represent a vector image because every cell within the raster image must specify a pixel value for the image mapping.



Modifying a raster image is difficult, slow, and generally results in loss of information that affects the quality of the image. Modifications to raster images are applied to every pixel value comprising the entire image.

Contrast this with modifying vector-based images, which affects only the pixels that connect the geometric description of the objects.

Reversing the alteration of a vector-based image results in an image very near the original, whereas raster images are degraded to some extent.



Next Steps

It is crucial to understand the relationship between the vector-based Xlib calls and the bitmapped display nature of the X Window System before proceeding to the next chapter.

Chapter 9, "Object Bounds Checking," focuses on the best means for determining the bounds of an object based on the graphic descriptions associated with different geometric shapes.

Because an object's bounds are used later in the Graphics Editor project to determine whether an X ButtonPress event has selected the graphic (made it active for manipulation), correctly calculating the occurrence of an event within an object's bounds is necessary for installing proper object control.

Chapter 9

In this chapter

- Point-Array-Based Object Bounds
- Arc Object Bounds
- Employing Object Bounds
- Next Steps

Object Bounds Checking

Every object of the Graphics Editor application has an associated size, position, width, and height. This chapter discusses methods of using the descriptive information available to the editor objects for determining whether an X ButtonPress Event occurred within the bounds of the object. This test is necessary to support the function of *selection* on the objects, making them eligible for manipulation and insuring proper object control.

The list of objects created by the Graphics Editor includes LatexLine, PolyLine, Box, Arrow, Arc, and Text.

Calculating the *bounds* of an object is dependent on the geometric shape represented by the object.

The bounds of an object refer to its dimensions and relative placement within the canvas window.

Because all events that occur within the window used for drawing are relative to the window's origin, the placement and extents of an object must be viewed similarly. Therefore, the calculation of an object's bounds allows events to be compared to objects to determine whether the event occurred within or on the object.

A structure useful for representing the bounds of an object is shown in Listing 9.1.

Listing 9.1 Bounds Structure

1: typedef struct _bounds {
2: int x;
3: int y;
4: Dimension width;
5: Dimension height;
6: } Bounds;

Listing 9.1 defines the data type used in the Graphics Editor project for representing the bounds of the object.



The data type Dimension used for the width and height fields of the Bounds structure defined in Listing 9.1 is defined in the X Window environment as

typedef unsigned int Dimension;

Because the LatexLine, PolyLine, Box, Arrow, and Text objects are all represented with similar internal structures, namely as a series of XPoints contained in an array, the method of calculating the bounds of these objects will be the same. Taking into account the Arc object, two methods are required for calculating object bounds. One method serves point-array-based objects and one method is explicit to the Arc object.

Point-Array–Based Object Bounds

Calculating the bounds of the editor objects represented as an array of points is as simple as assigning the (x,y) origin of an object into the Bounds structure and calculating the minimum and maximum points contained in the array defining the object. This is shown in the code fragment of Listing 9.2.

```
Listing 9.2 Line Object Bounds Calculation
```

```
// defintion of GXLine Object
1:
2:
       typedef struct gx line {
3:
4:
         int num_pts; // length of points array
5:
         XPoint pts[]; // array to hold points defining object
6:
       } GXLine, *GXLinePtr;
7:
8:
       // macros for determining minimum and maximum values
9:
       #define GXMIN(a, b) ((a) < (b) ? (a) : (b))
10:
       #define GXMAX(a, b) ((a) < (b) ? (b) : (a))
11.
       // sample of parsing GXLine points array
12:
       Bounds *getLineBounds( GXLinePtr line )
13:
14.
       {
         static Bounds bounds; // static makes the variable
15:
                               // persistent after the function
16.
17:
                               // returns
18:
         int i, min_x = 999, min_y = 999, max_x = -999, max_y = -999;
19:
20:
         bounds.x = line.x
21:
         for( i = 0; i < line->num_pts; i++ ) {
           min x = GXMIN( min x, line->pts[i].x );
22:
23:
           min_y = GXMIN( min_y, line->pts[i].y );
```

```
24: max_x = GXMAX( max_x, line->pts[i].x );
25: max_y = GXMAX( max_y, line->pts[i].y );
26: }
27: bounds.width = (max_x - min_x);
28: bounds.height = (max_y - min_y);
29:
30: return &bounds; // return a pointer to the structure
31: }
```

Listing 9.2 shows the definition of the GXLine structure used to represent the data specific to the point-array-based objects used in the Graphics Editor.

The difference of the minimum and maximum points representing the line object determines the dimensions of the object as illustrated in Figure 9.1.



The points defined for the line object shown in Figure 9.1 can be defined as XPoint pts[6] = {{54, 97}, {113, 55}, {155, 120}, {101, 107}, {54, 97}}; There are five points in the array accounting for the last line that closes the object. The maximum x point found is at element two (155) and the smallest is at the first and last elements (54). The largest y point is at element three (107) and the smallest is at element two (55).

Applying the width and height calculations seen in Listing 9.2 reveals

```
width = 155 - 54 = 99;
height = 107 - 55 = 52;
```

At the return of the function getLineBounds, the bounds structure for this object is

```
bounds.x = 50;
bounds.y = 50;
bounds.width = 99;
bounds.height = 52;
```

After illustrating the calculation for determining the bounds of an arc object, I will show you how to use the bounds of an object to accomplish object selection.

Arc Object Bounds

The definition of the arc-specific object data, as shown in Listing 9.3, implicitly describes the bounds of the object. Because of this, the function for calculating the arc's bounds is significantly simpler than seen with the line object.

```
Listing 9.3 Arc Object Bounds Calculation
```

```
1:
       typedef struct _gx _arc {
2:
         int x, y;
3:
         Dimension width, height;
         int angle1, angle2;
4:
5:
6:
       } GXArc, *GXArcPtr;
7:
       void getArcBounds( GXArcPtr arc )
8:
9:
       {
         static Bounds bounds;
10:
11:
12:
         bounds.x = arc - x;
13:
         bounds.y = arc - y;
14.
15:
         bounds.width = arc->width;
16:
         bounds.height = arc->height;
17:
       }
```

Recall the definition of the XArc for representing arcs in X, as demonstrated in Figure 9.2.



Consider the example of the arc defined in Figure 9.3.



The example in Figure 9.3 produces an XArc defined as

```
arc.x = 10;
arc.y = 20;
arc.width = 100;
arc.height = 100;
arc.angle = 0;
arc.angle = 270*64;
```

The data defining the arc object directly transfers to define the bounds of the object.

```
bounds.x = 10;
bounds.y = 20;
bounds.width = 100;
bounds.height = 100;
```

Now that we have bounds definitions for sample objects of a line and an arc type, let us see how to employ them.

Employing Object Bounds

Consider the case where a user right-clicks in the drawing window to process a ButtonPress event for the location x,y as shown in Figure 9.4.



By comparing the bounds calculated for each of the objects in the drawing window to the ButtonPress event location, an object can be deemed intended for selection by the user.

Consider the bounds calculated for the line object seen in Figure 9.1 and repeated in Figure 9.4 as compared to the sample event using the gxObjSelect function found in Listing 9.4.

```
Listing 9.4 Testing Object Selection
```

```
Boolean gxObjSelect( Bounds bnds, XEvent *xe )
1:
2:
       {
3:
         if( xe->xbutton.x > bnds->x &&
4:
             xe->xbutton.y > bnds->y &&
             xe->xbutton.x < (bnds->x + bnds.width) &&
5:
6:
             xe->xbutton.y < (bnds->y + bnds.height)) {
7:
8:
           return True; // object selected
9:
         }
10:
         return False; // object not selected
11:
       }
```

The function shown in Listing 9.4 uses the bounds calculated from the line object passed to the getLineBounds shown in Listing 9.2.

Comparisons are made to determine whether the event occurred at a point greater than the origin of the object

and less than the extents of the object.

```
5: xe->xbutton.x < (bnds->x + bnds.width) &&
6: xe->xbutton.y < (bnds->y + bnds.height)) {
```

Notice that the extents of the object are defined as where the horizontal edge and vertical edge of the object are placed on the screen relative the object's origin.

If any element of the if statement fails, the entire test will fail because of the use of the Boolean AND (&&) function.

Applying the values of the illustration, the line object would appear selected by the user using this method of determination.

Refer to Listing 9.4 for a demonstration of the application of the ButtonPress event to the XArc example from Figure 9.3. It too appears to be selected.

The shortcoming highlighted in these examples is the confusion that occurs using the method of selection shown in Listing 9.4.

Not only do both objects appear to be selected, but also the ButtonPress event did not even touch the line object.

Consider the ButtonPress event illustrated in Figure 9.5.

ButtonPress for object selection.

Figure 9.5



Again the ButtonPress event is within the bounds of both the sample objects. This time the event touches neither of the objects, which, when using the gxObjSelect from Listing 9.4, makes both objects appear selected.

This is a limitation of the selection function that must be overcome to ensure correct object control and manipulation.

Next Steps

To select objects properly using the Graphics Editor, a method for determining whether an object is selected must return true only if the ButtonPress event explicitly falls on the object.

To accomplish a selection function of the required sophistication, an understanding of certain trigonometric functions must first be discussed for detecting whether a point falls on a line or lies beyond the arc defined by the end and center points of an arc object.

Chapter 10, "Trigonometric and Geometric Functions," demonstrates a finer method of determining object selection and reviews the mathematical skills necessary for the task.

Chapter 10

In this chapter

- Calculating Point and Line Intersections
- Calculating Slope
- Calculating Point and Arc Intersections
- Next Steps

Trigonometric and Geometric Functions

Chapter 9, "Object Bounds Checking," showed the necessity of accurately selecting graphic objects created by the editor and the frailty of using solely the bounds of the object for making the determination of selection. This chapter reviews the mathematics required for a more precise selection algorithm.

Chapter 9 discussed the objects of the editor as being point-array-based or as being an arc. We first consider the intersection of a point and a line.

Calculating Point and Line Intersections

Objects created by our editor that are defined as an array of points can be considered a series of line segments where each pair of points in the array defines the segment's endpoints.

For instance, an array of four points defined as

XPoint pts[4] = { $\{10, 10\}, \{20, 10\}, \{20, 20\}, \{10, 20\}\};$

would consist of three line segments:

- 1. (10,10) to (20,10)
- 2. (20, 10) to (20, 20)
- 3. (20, 20) to (10, 20)



The function XDrawLines does not implicitly *close* the object represented by the point array, meaning that the endpoints are not connected. To accomplish this, the array would have to be expanded to contain five points and the starting point repeated as the last element of the array.

Having reduced our point-array-based object to a series of line segments, we can further clarify the precise selection issue by viewing the ButtonPress event entered in the canvas window if the user wants to select an object as the point where the event occurred.

Defining the graphic object as a series of line segments and the event for selection as merely a point clarifies the requirement of precise object selection. Knowing when an object is selected requires the capability to determine when the point represented by the event intersects a line segment of the object.

Listing 10.1 shows the algorithm for accomplishing this form of selection.

Listing 10.1 The near_segment Function

```
/*
1:
2:
       * near_segment
       */
3:
      Boolean near_segment( int x1, int y1, int x2, int y2, int xp, int yp )
4:
5:
      {
6:
        int xmin, ymin, xmax, ymax;
7:
        float slope, x, y, dx, dy;
8:
9:
        if( abs(xp - x1) <= TOLERANCE && abs(yp - y1) <= TOLERANCE ) {
10:
          return True;
11:
        }
12:
13:
        if (abs(xp - x2) \le TOLERANCE \& abs(yp - y2) \le TOLERANCE)
14:
          return True;
15:
        }
16:
        if( x1 < x2 ) {
17:
          xmin = x1 - TOLERANCE;
18.
19:
          xmax = x2 + TOLERANCE;
20:
        } else {
          xmin = x2 - TOLERANCE;
21:
          xmax = x1 + TOLERANCE;
22:
23:
        }
24:
25:
        if( xp < xmin || xmax < xp ) {
26:
          return False;
27:
        }
28:
        if( y1 < y2 ) {
29:
```

```
30:
          ymin = y1 - TOLERANCE;
          ymax = y2 + TOLERANCE;
31:
32:
        } else {
33:
          ymin = y2 - TOLERANCE;
34:
          ymax = y1 + TOLERANCE;
35:
        }
36:
37:
        if (yp < ymin || ymax < yp)</pre>
38:
          return False;
39:
40:
        if( x2 == x1 ) {
          x = x1;
41:
42:
          y = yp;
43:
        } else if( y1 == y2 ) {
44:
          x = xp;
45:
          y = y1;
        } else {
46:
47:
          slope = ((float) (x2 - x1)) / ((float) (y2 - y1));
48:
          y = (slope * (xp - x1 + slope * y1) + yp) / (1 + slope * slope);
49:
          x = ((float) x1) + slope * (y - y1);
50:
        }
51:
52:
        dx = ((float) xp) - x;
53:
        dy = ((float) yp) - y;
54:
        if ( (float)(dx * dx + dy * dy) < (float)(TOLERANCE*TOLERANCE) ) {
55:
56:
          return True;
57:
        }
58:
59:
        return False;
60:
      }
```

The function near_segment shown in Listing 10.1 considers only a single line segment represented by endpoints

```
int x1, int y1, int x2, int y2
```

passed as the first four parameters of the function.

```
4: Boolean near_segment( int x1, int y1, int x2, int y2, int xp, int yp )
```

The final two parameters are the x and y points contained in the ButtonPress event.

The function begins by testing the distance of the event point xp and yp from each of the endpoints defining the line segment.

```
9: if( abs(xp - x1) \le TOLERANCE \& abs(yp - y1) \le TOLERANCE ) { and
```

13: if(abs(xp - x2) <= TOLERANCE && abs(yp - y2) <= TOLERANCE) {



The TOLERANCE variable used in various places throughout the Graphics Editor project is defined in a header file not yet introduced.

Its use is to set the allowable margin of error for interactive functions. Specifically, when the user attempts to click on the line object, he might be TOLERANCE distance from it and still successfully accomplish the selection.

This is necessary because the line object can only be a single pixel wide (line width of 1). As the event is also a single pixel (x, y) point, the level of precision required to have the event land exactly on the line object could be too great considering the steadiness of most of our hands.

EXCURSION

You Must Exercise Your abs

The function **abs** used in testing the difference between the event points and the line segment endpoints

abs(xp - x1)

and

abs(yp - y1)

is the absolute value function.

Because the near_segment function does not test whether the event point is less than the endpoint before doing the subtraction, it is possible to have a negative result.

The **abs** function requests that the resulting sign of the operation be stripped, forcing the result to be positive.

Without the **abs** function, the determination of the result to be less than or equal **TOLERANCE** would not always have the intended results, as an event point that was to the right of the endpoint would appear to be within a distance of **TOLERANCE** or 3 pixels.

The intended semantic is to determine whether the event is within 3 pixels to *either* side of the endpoint, necessitating that the absolute value of the distance be tested.

If the event points are an acceptable distance from one of the line segment's endpoints, the function returns true and the line object is considered selected.

If the event point is somewhere within the line segment, the function continues by ordering the points defining the line segment.

17: if(x1 < x2) {
18: xmin = x1 - TOLERANCE;
19: xmax = x2 + TOLERANCE;
20: } else {
21: xmin = x2 - TOLERANCE;</pre>

```
22: xmax = x1 + TOLERANCE;
23: }
```

The ordering of the points is accomplished by testing to see whether x1 is less than x2. If x1 is the lesser of the two points, its value is assigned as xmin, making the value of x2 the maximum or xmax value.



By subtracting TOLERANCE from the xmin value and adding it to the xmax value, you increase the bounding box defined by the line segment.

By ordering the points, near_segment can perform a *range check* to determine whether the event point occurred in close enough proximity to the line segment to merit continuing in the function.

If the event point is outside of the bounds of the line segment, there is no reason to perform any more calculations and tests because it would be impossible for the event point to intersect the line.



Notice that the bounds checking being performed here is not the same as the bounds checking done in the previous chapter.

Here only a single line segment comprising the whole object is being tested, making for a much finer determination. Further, an event point lying within the bounds of the segment is used only to determine whether further calculations and testing should be performed and not whether the object should be considered selected.

After determining that the event point is within the bounds of the line object, a test is made to ensure that the x value of the endpoints are not equal, indicative of a vertical line. If the line is vertical, the variables x and y are assigned the x value of the endpoints (x1) and the y component of the event point (yp) respectively.

```
40: if(x2 == x1) {
41: x = x1;
42: y = yp;
```

If the line is not vertical (x components of the endpoints are not equal), the line is tested for being horizontal. A horizontal line will have equal y values for the endpoints and the y value of the endpoints and event x component will be used for the values of the variables x and y.

If the line is neither vertical nor horizontal, the slope of the line is calculated and used to determine the values of the x and y variables.

```
46: } else {
47: slope = ((float) (x2 - x1)) / ((float) (y2 - y1));
48: y = (slope * (xp - x1 + slope * y1) + yp) / (1 + slope * slope);
49: x = ((float) x1) + slope * (y - y1);
50: }
```

Figure 10.1 demonstrates the relationship of the variables slope; x1, y1; x2, y2; xp, yp; and x, y.



As shown in Figure 10.1, the variables x and y will be some point on the line as determined by one of the three steps we've seen.

Calculating Slope

The slope of a line is defined as the steepness or tilt of the line. It is measured as a ratio of vertical change (rise) over horizontal change (run). In mathematics, slope is usually designated by the letter m.

Given a line passing through points (x1, y1), (x2, y2) the slope m of the line determined by

 $m = \frac{y^2 - y^1}{x^2 - x^1}$

as long as $x^2 \neq x^1$.



The slope of a vertical line is undefined and the slope of a horizontal line is zero. For this reason, near_segment special cases these conditions.

Converting the equation for calculating slope to the C programming language is straightforward

47: slope = ((float) (x2 - x1)) / ((float) (y2 - y1));



The use of the keyword float in the calculation of slope is called a *cast*.

Casting variables is necessary to promote their data types either for compatibility during assignments or precision in calculations.

The cast of the result of $(x_2 - x_1)$ and $(y_2 - y_1)$ to the data type of float ensures that the division operation is done with floating point numbers. A great deal of precision would be lost if slope were calculated using integers or whole numbers.

Calculating the x, y point on the line as required in Case 3 of Figure 10.1 is not as straightforward.

Case 3 requires that functions for determining x and y be derived from one or more of the forms for writing *linear equations* and applied to what is known about the line segment and event point.

48: y = (slope * (xp - x1 + slope * y1) + yp) / (1 + slope * slope); 49: x = ((float) x1) + slope * (y - y1);

EXCURSION

Common Equations for Representing a Line

Linear Equations are functions for representing lines based on the information known about them.

Table 10.1 shows three of the more common forms for writing linear equations.

Form	Function	Description
General	Ax + By = C	A, B, and C are not fractions
Slope-Intercept	y = mx + b	b is the y-intercept or point at which the y-axis is crossed
Point-Slope	y - y1 = m(x - x1)	The line is known to contain the point (x1, y1)

For instance, employing the Point-Slope form and solving independently for x and y identifies the point needed by Case 3 of Figure 10.1.

When the point (x, y) is found, the distance of this point from the event point is calculated:

```
52: dx = ((float) xp) - x;
53: dy = ((float) yp) - y;
```

Finally, the line is considered selected if twice the sum of the deltas squared (distance between the x and y component of the event point and the point x, y found on the line) is less than the square of the tolerance.

```
55: if ( (float)(dx * dx + dy * dy) < (float)(TOLERANCE + TOLERANCE) ) {
56: return True;
57: }</pre>
```

When the evaluation of the if condition results in False, the program will fall through to

59: return False;

because the line segment was not selected.

If the near_segment evaluation for any one of the line segments comprising the entire point-array-based object results in a return value of True, the object is considered selected and no further consideration is necessary.

The solution presented by the near_segment function is clearly superior albeit more complex than using only the bounds of the object for determining the users intent for object selection.

Selection of point-array-based objects has been improved greatly over the bounds method introduced in Chapter 9. We now turn our attention to the selection of arcs.

Calculating Point and Arc Intersections

As introduced in the previous chapter and iterated again in the context of refining the selection process for point-array-based objects, using solely the bounds of an object to determine whether a ButtonPress event point was meant to select the object is insufficient.

In the same way that near_segment calculated the proximity of the event point to the segments defining the object, a method is needed to accomplish the same for arcs.

As introduced in Chapter 7, "Xlib Graphic Primitives," in the section "XDrawArc", page 192, arcs are described in the X Window System by

x, y, width, height, angle1, angle2

To facilitate this representation, X provides the XArc structure for holding the required arc data.

```
typedef struct {
    int x, y;
    Dimension width, height;
    int angle1, angle2;
} XArc;
```

The purpose of the arc_find function found in Listing 10.2 is to determine whether an event point intersects the arc defined by an XArc structure and specified as the first parameter.

```
Listing 10.2 The arc_find Function
```

```
1:
      /*
2:
       * arc_find
3:
       */
4:
      static Boolean arc_find( XArc *arc_data, XEvent *event )
5:
      {
6:
        double rx,
                        ry,
7:
               cx,
                        cy,
8:
               ex,
                        ey,
9:
               f1x,
                        f1y,
10:
               f2x,
                        f2y,
11:
               hmaj,
                        hmin,
                        angle,
12:
               d,
13:
               angle1, angle2;
14:
        rx = (float)arc data->width / 2;
15:
16:
        ry = (float)arc_data->height / 2;
17:
18:
        cx = arc data -> x + rx;
19:
        cy = arc_data->y + ry;
20:
21:
        d = sqrt( fabs( sqr(rx) - sqr(ry) ) );
22:
        if( rx >= ry ) {
23:
          f1x = cx - d;
24:
          f1y = cy;
25:
          f2x = cx + d;
          f2y = cy;
26:
27:
          hmaj = rx;
28:
          hmin = ry;
        } else {
29:
30:
          f1x = cx;
31:
          f1y = cy - d;
32:
          f2x = cx;
33:
          f2y = cy + d;
34:
          hmaj = ry;
35:
          hmin = rx;
36:
        }
37:
```

10

Listing 10.2 Continued

```
38:
        ex = event->xbutton.x;
39:
        ey = event->xbutton.y;
40:
41:
        angle = atan2(cy - ey, ex - cx) * 180/M PI;
42:
        if( angle < 0 )
43:
          angle += 360;
44:
45:
        angle1 = (double)arc data->angle1 / 64.0;
46:
        angle2 = (double)arc data->angle2 / 64.0;
47.
48:
        if( sqrt( sqr(f1x - ex) + sqr(f1y - ey) ) +
49:
          sqrt(sqr(f2x - ex) + sqr(f2y - ey)) > 2 * (hmaj + TOLERANCE))
50:
          return( False );
51:
        if( sqrt( sqr(f1x - ex) + sqr(f1y - ey) ) +
52:
          sqrt(sqr(f2x - ex) + sqr(f2y - ey)) < 2 * hmaj - TOLERANCE)
53:
          return( False );
54:
        return( True );
55:
      }
```

Following the declaration of the many variables used to determine whether the event point intersects the arc is the calculation of half of the *major* and *minor* axes.

```
15: rx = (float)arc_data->width / 2;
16: ry = (float)arc_data->height / 2;
```

∐≣¦< Note

The *major* and *minor axis* refer to the diameter on the x-axis and y-axis. The longer of the two is called the major axis and the shorter is the minor axis.

Half the distance of the major and minor axis determines the radius on the x-axis and the radius on the y-axis.

Independent values are maintained for radius on the x-axis and radius on the yaxis because the *aspect ratio* of the arc might not be constant.

I∎ **⊂** Note

Aspect ratio is the proportion of width to height.

A 360° arc with a constant aspect ratio has a width and height that are equal and the resulting object is a circle. Therefore, a circle's major and minor axes are equal, or in other words, the diameter is constant on both the x-axis and the y-axis.

If the width and height for the same arc object are not equal (the aspect ratio is not constant), the resulting object is an ellipse. An ellipse has a different diameter on the x-axis and the y-axis.

Care has been taken to not use the term *circle* or *ellipse* to describe the corresponding Graphics Editor objects because there is no enforcement of a constant aspect ratio. Instead, the generic description *arc* is used to describe both circles and ellipses as well as actual arcs (objects that are less than 360°) created by a user.

Following the calculation of the radii for the x and y-axes, the center point of the arc is found.

```
18: cx = arc_data->x + rx;
19: cy = arc_data->y + ry;
```

Next, we calculate the distance between the center of the ellipse and the *foci* as well as the coordinates of the foci. This calculation is dependent upon the orientation of arc (in the case of a non-constant aspect ratio). So a test is made on whether the radius on the x-axis is greater than or equal to the radius on the y-axis.

EXCURSION

It Was Not Foci Bearing Gifts at the Nativity?

The foci are *fixed points* from which all points on the *arc* (circle or ellipse) are a constant distance. Other than being a cool word to say, foci are crucial to the definition of an arc.

Foci is plural and in this context refers to two fixed points, one relative to the major axis and one to the minor axis.

In the case of a circle, the two points are the same and correspond with the center of the circle.

For an ellipse, the major axis containing the foci is always longer than the minor axis. Thus, a test to determine the longer of the two axes is necessary to correctly calculate the distance of the foci from the center and the foci coordinates.

22: if(rx >= ry) {

This definition of foci prepares you for a definition of a circle and an ellipse to demonstrate the use of foci.

A circle is defined as

PO + PO = 2a

where a is the radius of the circle, P is any point on the circle, and O is the foci. (Remember that the major and minor axes are equal and therefore the foci are equal.)

Figure 10.2 illustrates the definition of a circle.



As demonstrated in Figure 10.2, PO = a is the radius of the circle and therefore 2a is the diameter. In other words, distance *PO* from any point *P* on the circle to the (convergence) of the foci *O* is the constant *a*.

In an ellipse, however, the foci are not equal and they do not correspond to the center point of the object, as demonstrated in Figure 10.3.



If we name one focus (singular for foci) f and the other r, we can form the definition of an ellipse as

Pf + Pr = 2a

where P is any point on the ellipse.

The definition reads that the distance from the foci of an ellipse is the same for all points on the curve. Thus if *r* and *f* are the foci, the total distance Pf + Pr from the foci to any point *P* is constant (*2a*).

Consider the first case demonstrated in Figure 10.3 where the major axis coincides with the x-axis

22: if(rx >= ry) {

The foci are calculated to be

23:	f1x	= cx - d;
24:	f1y	= cy;
1		
and		
25:	f2x	= cx + d;
26:	f2y	= cy;

Consistent with the explanation, the major axis (hmaj) is assigned the value of radius on the x-axis, and the minor axis (hmin) the value on the y-axis.

27: hmaj = rx; 28: hmin = ry;

Case 2 illustrated in Figure 10.3, however, sets the major and minor axis to

34: hmaj = ry; 35: hmin = rx; and assigns the coordinates of the foci to be

30:	f1x	= cx;	
31:	f1y	= cy -	d
32:	f2x	= cx;	
33:	f2y	= cy +	d

More information is required before actually determining whether the event point intersects the angle.

First, determine the angle measured from the positive x-axis (3 o'clock position) of the vector (ex, ey), (cx, cy) measured in degrees:

atan2(cy - ey, ex - cx) * $180/M_{PI}$



If the angle is less than zero, rotate it so

42: if(angle < 0)
43: angle += 360;</pre>

Next, convert the angle1 and angle2 measurements used by X to degrees. As you recall from the introduction to XDrawArc in Chapter 7, section "XDrawArc," page 192, X maintains these angles as integers in the form "degrees × 64."

```
45: angle1 = (double)arc_data->angle1 / 64.0;
46: angle2 = (double)arc_data->angle2 / 64.0;
```

Finally, apply the information gathered and see whether the arc has been selected.

The determination is whether the sum of the distances between the event point and the foci is greater than the major axis by more than TOLERANCE. If it is greater, the arc has not been selected.

Consider each focus separately. First, (f1x, f1y)

```
48: if( sqrt( sqr(f1x - ex) + sqr(f1y - ey) ) +
49: sqrt( sqr(f2x - ex) + sqr(f2y - ey) ) > 2 * (hmaj + TOLERANCE))
50: return( False );
```

10

```
Then, (f2x, f2y)

51: if( sqrt( sqr(f1x - ex) + sqr(f1y - ey) ) +

52: sqrt( sqr(f2x - ex) + sqr(f2y - ey) ) < 2 * hmaj - TOLERANCE )

53: return( False );
```

If the sum of the differences is not greater, the function returns true to indicate that the arc has successfully been selected by the event point.

```
54: return( True );
```

EXCURSION

Introducing a Limitation of the X Window System

A limitation of the X Window System is the inability to draw ellipses that have a major axis and minor axis that are not parallel to the x-axis and y-axis. Figure 10.4 illustrates such an ellipse.



One solution is to convert the ellipse into a series of line segments using a Bézier algorithm and then treat it like a point-array–based object.

With a superior method of selecting both point-array-based objects and arc objects, you are ready to review the concepts required to perform geometric transformations.

Next Steps

Chapter 11, "Graphic Transformations," introduces and satisfies the requirements for performing actions such as moving, scaling, and rotating the objects of the Graphics Editor. Understanding these actions (known as geometric transformations) will bring you close to completing the foundation the last several chapters have so meticulously worked to lay out.

Chapter 11

In this chapter

- Moving
- Scaling
- Rotating
 Next Steps

Graphic Transformations

We have studied calculations for accurately determining when objects in the Graphics Editor are selected from the canvas, and we now shift our focus to the actions that can be applied to an object that has been successfully selected.

Graphic transformations, as implied by the phrase, are actions that alter the graphic object. The Graphics Editor supports the move, scale, and rotate functions for altering the objects of the editor.

Each of these is discussed in the following sections. As was necessary with the analysis of proper object selection, consideration is given separately to point-based objects and arc objects because the transformation methods are unique to the data fields of the varying objects.

Moving

Moving an object, regardless of type, requires the application to maintain the delta or distance the cursor has traveled between iterations of the move request.

To move an object in the Graphics Editor, first select the object and then press and hold the right mouse button over the object while moving the cursor. The distance the cursor travels is applied to the *location* designator of the object.

For point-array-based objects, the object's location is implicit to the values of the points contained in the array.

Arcs, however, have an explicit x, y value that describes their placement in the canvas window.

The following two sections show the functions that manage the updates required for the point-based and arc objects to move, based on the coordinates of an X event understood to be a PointerMotion event structure.

Moving a Line

Listing 11.1 demonstrates how the Graphics Editor tracks and applies the distance the cursor has traveled. Although the GXObj has not been defined, you know from Listing 9.2 in Chapter 9, "Object Bounds Checking," the definition of the GXLine structure. This structure provides the unique data field definition for GXObjs of type line.

→ The GXLine structure is presented in Chapter 9, section "Point-Array-Based Object Bounds," page 204.

Listing 11.1 The line_move Function

```
static void line move( GXObjPtr line, XEvent *event )
1:
2:
      {
3:
        static int x = 0, y = 0;
4:
5:
        GXLinePtr line data = (GXLinePtr)line->data;
6:
        int i;
7:
8;
        if( x && y ) {
9:
          XDrawLines( XtDisplay(GxDrawArea), XtWindow(GxDrawArea), rubberGC,
10:
                       line data->pts, line data->num pts, CoordModeOrigin );
11:
        } else {
12:
          /* our first time through */
13:
          (*line->erase)( line );
14:
          x = event ? event->xbutton.x : 0;
15:
          y = event ? event->xbutton.y : 0;
16:
17:
        }
18:
19:
        if( event ) {
          for( i = 0; i < line_data->num_pts; i++ ) {
20:
            line_data->pts[i].x += (event->xbutton.x - x);
21:
22:
            line data->pts[i].y += (event->xbutton.y - y);
23:
          }
24:
25:
          /*
26:
           * draw rubberband line
           */
27:
          XDrawLines( XtDisplay(GxDrawArea), XtWindow(GxDrawArea), rubberGC,
28:
29:
                       line_data->pts, line_data->num_pts, CoordModeOrigin );
30:
31:
          x = event->xbutton.x;
32:
          y = event->xbutton.y;
33:
        } else {
```

34: x = 0; 35: y = 0; 36: } 37: }

The line_move function in Listing 11.1 begins by declaring the local static variables:

```
3: static int x = 0, y = 0;
```

EXCURSION

The Keyword static Has Multiple Uses

Contrast the use of the keyword **static** as applied to local variables compared to its use in the declaration of the **line move** function

1: static void line_move

→ See Chapter 3, page 103 for a discussion of the **static** keyword.

When applied to local variables, the static keyword ensures that their values are residual (maintained) between calls to the function. This is critical to the line_move function because it must retain the (x, y) values of the PointerMotion event driving the placement of the object being moved.

Applied to the function declaration, however, static simply limits the scope (visibility) of the function to this file.

The static variables x, y retain the coordinates of the X PointerMotion event, allowing only the incremental delta of the distance the pointer has traveled to be applied to the points defining the object.

In a moment, you will see that the value of 0 assigned to the x and y is important for knowing when the move action begins.

The next executable line of the function

5: GXLinePtr line_data = (GXLinePtr)line->data;

declares a variable line_data and assigns it the unique data field from the GXObjPtr line passed as the first parameter to the line_move function.

After declaring a simple integer for looping, the function tests to see if x and y have non-zero values:

6: int i;
7:
8; if(x && y) {

Non-zero values for the x and y indicate that this is not the first iteration of the move action, an important thing to know because the *rubber-banding* object must be erased from the screen.

9: XDrawLines(XtDisplay(GxDrawArea), XtWindow(GxDrawArea), rubberGC, 10: line_data->pts, line_data->num_pts, CoordModeOrigin);

I■ ⊂ Note	The rubber-banding effect used during object creation and transformation in the Graphics Editor project refers to drawing the object in an <i>interactive mode</i> .
	Allowing the user to move an object around the screen, other objects existing on the canvas are not erased by the drawing and erasing of the object in interactive mode.
	→ Review the immediate graphic nature of X as discussed in Chapter 4, "Windowing Concepts," section "Expose," page 122.
	→ To prevent erroneously erasing other objects on the canvas, the GC func- tion GXxor introduced in Chapter 7, "Xlib Graphic Primitives," section "The GC Function," page 182, is assigned the global GC rubbergc.
	The rubbergc graphic context will draw the object the first time the X Graphic Primitive is requested and erase it the second time without harming the under-



Caution (and careful management) must be applied to the number of calls made to a primitive employing a GC with GCFunction set to GXxor because an odd number of calls will leave residual rubber-banded objects onscreen.

If the x and y values are zero, this is seen as the first iteration of the move request, in which case the else is reached. In the body of the else, the erase method of the GXObj is invoked to clear the object from the canvas. It is immediately redrawn in the interactive (non-destructive) mode using the rubbergc.

lying background or objects.

In the body of the else, calling the object's erase method removes the object from the canvas, and then the values of x and y are set to equal the coordinates of the PointerMotion event *if it exists*.

III ⊂ Note

Although the specific GX0bj definition is not introduced until Chapter 15, "Common Object Definition," suffice it to say that an object will have methods (internal functions) for accomplishing all necessary self-management tasks pertinent to the type of object it is.

Object methods include functions for drawing, erasing, moving, scaling, rotating, copying, saving, restoring, and deleting.

Perhaps Chapter 15 should have been titled "A Horse for the Cart."

The object's method for erasing is not compatible with the call to the X primitive XDrawLine that is used to manage the rubber-banding effect.

The incompatibility results from the difference in GC function used to draw the object on the screen.

All the GX0bj methods for drawing employ the GXcopy function that cannot be erased or undone with the GXxor function.

→ As Chapter 20, "Latex Line Object," demonstrates, and Chapter 7, "X Lib Graphic Primitives," section "GCTile," page 188 introduced, the method of erasing objects placed onscreen with the GXcopy is to *tile* in the canvas background where the object being erased currently resides.

EXCURSION

A New Form of the if then else Construct

The C language syntax

```
var = test ? value1 : value2;
```

is equivalent to

```
if( test )
  var = value1;
else
  var = value2;
```

Although not clear or readily readable, the notation is very common with C programmers.

The necessity of testing for the implicit non-null value of event is to account for the transformation function line_move being interrupted or ended.

To abort or end the move action, a null event pointer is passed to the line_move function, allowing it to reset the values of x and y to zero so the transformation can start over with another object.

11
The implicit test for a non-null event follows the body of the else entered when the values of x and y were zero.

```
An implicit test follows the form

if ( variablePtr )

C interprets this as a test for a non-zero value. As NULL is defined in C

as #define NULL (unsigned long)0 or #define NULL (void *)0.

The test is compatible with pointer as well as non-pointer variables.

An explicit test is clearly preferred over an implicit test because it prevents

semantic errors and makes the test more readable. Explicit tests, as the phrase

implies, explicitly test for the expected value. The equivalent explicit test for the

sample given above is

if ( variablePtr != NULL )
```

As discussed earlier, sending a null event pointer causes the function to reset the values of x and y so the move action can be repeated.

```
19: if( event ) {
20: for( i = 0; i < line_data->num_pts; i++ ) {
21: line_data->pts[i].x += (event->xbutton.x - x);
22: line_data->pts[i].y += (event->xbutton.y - y);
23: }
```

If there is a valid event pointer, the difference between the previous and current PointerMotion event location is applied to every point in the line_data->pts array.

After updating all the points, the object is implicitly placed at the new location and the rubber-banding object is drawn to reflect this by using the global GC variable rubberGC.

28: XDrawLines(XtDisplay(GxDrawArea), XtWindow(GxDrawArea), rubberGC, 29: line_data->pts, line_data->num_pts, CoordModeOrigin);

With the interactive object drawn to the screen, it is imperative that the current event location be retained so that the next iteration of the function can erase it.

```
31: x = event->xbutton.x;
32: y = event->xbutton.y;
```

As the line_move function is invoked, the line object is continually updated to the new cursor location by adding the difference that the cursor moved from the previous call.

A similar move function must exist for the GXObj of type arc.

Moving an Arc

The method of moving an arc object is simpler than the point-array–based object because its location is assigned explicitly by the x and y data fields in the XArc structure.

Listing 11.2 shows the function for performing the move transformation on an arc object.

```
Listing 11.2 The arc move Function
```

```
1:
      /*
2:
       * arc_move
3:
       */
      static void arc_move( GXObjPtr arc, XEvent *event )
4:
5:
      {
        static int x = 0, y = 0;
6:
        XArc *arc_data = (XArc *) arc->data;
7:
8:
9:
        if( x && y ) {
10:
          /*
11:
           * erase the rubberband arc
           */
12:
          XDrawArc( XtDisplay(GxDrawArea), XtWindow(GxDrawArea), rubberGC,
13:
14:
                    arc data->x, arc data->y,
15:
                    arc data->width, arc data->height,
                    arc_data->angle1, arc_data->angle2 );
16:
17:
18:
        } else {
19:
          /*
           \ast our first time through - erase the actual arc...
20:
21:
           */
22:
          (*arc->erase)( arc );
23:
24:
          /*
25:
           * ...store the current event location
           */
26:
27:
          x = event ? event->xbutton.x : 0;
          y = event ? event->xbutton.y : 0;
28:
29:
        }
30:
31:
        if( event ) {
32:
          /*
33:
           * get the x,y delta
           */
34:
35:
          arc_data->x += (event->xbutton.x - x);
36:
          arc_data->y += (event->xbutton.y - y);
37:
38:
          /*
39:
           * draw a rubberband arc
40:
           */
```

11

Listing 11.2 Continued

```
41:
          XDrawArc( XtDisplay(GxDrawArea), XtWindow(GxDrawArea), rubberGC,
42:
                     arc_data->x, arc_data->y,
43:
                     arc_data->width, arc_data->height,
44:
                     arc data->angle1, arc data->angle2 );
45:
46:
          x = event - xbutton.x;
47:
          y = event->xbutton.y;
48:
        } else {
49:
          x = 0;
50:
          y = 0;
51:
        }
52:
       }
```

The arc_move function shown in Listing 11.2 should be very familiar because it follows the form of the line_move function except that the data type XArc is being managed instead of a GXLine.

Focusing only on the differences between the line_move and arc_move functions, consider the assignments that explicitly place the arc object at the new location.

```
35: arc_data->x += (event->xbutton.x - x);
36: arc_data->y += (event->xbutton.y - y);
```

and the graphic primitive, which draws and erases the interactive object:

```
41: XDrawArc( XtDisplay(GxDrawArea), XtWindow(GxDrawArea), rubberGC,
42: arc_data->x, arc_data->y,
43: arc_data->width, arc_data->height,
44: arc_data->angle1, arc_data->angle2 );
```

Beyond these few lines, the arc_move function behaves exactly like the line_move function.

Advancing an understanding of graphic transformations, the next action to consider is scaling.

Scaling

The transformation of scaling an object refers to changing the object's height and width.

For many objects the width and height values refer only to the values defined by the object's bounds. Altering the width and height of the bounds of the object will affect each of the segments comprising the object because they must be *proportionally* altered to accomplish or obtain the new bounds of the object.

The challenge of scaling an object is that the x and y deltas (determined as they were with the move transformation by calculating the difference of the current and

previous event locations) are not applied linearly to the object. Rather, the *direction* of the scale action must be considered and a greater portion of the delta applied to the side corresponding to the direction of the scale than to the opposing side.

The Graphics Editor applied eight handles to an object made active by the selection process.

Figure 11.1 and Figure 11.2 illustrate how the handles are placed on Line and Arc objects respectively.



The placement of object handles when the user selects an object corresponds to every corner and every side of the object as shown in Figures 11.1 and 11.2.

Object found...

A *scale action* is instigated when the user selects an object to reveal its handles, positions the mouse cursor over the desired handle, and presses and holds the left mouse button to select the handle.

Exit

As long as the left mouse button is pressed, subsequent PointerMotion events will alter the width or height of the object in a manner relative to the handle selected.

When the mouse button is released, the object is redrawn with the new dimensions.

Returning to the discussion of the scaling challenge, consider the object shown in Figure 11.1. If the handle on the right side of the object is selected for the scale action, the point within the object closest to the handle must change with a greater proportion of the delta than the point furthest away.



If all points changed an equal amount, the result would be a move action instead of the intended scale action.

The final challenge when performing the scale transformation is the loss of *data integrity*.

As calculations are made to determine the proportion of the delta to apply to the object fields controlling the object's dimensions, *rounding errors* will occur.

EXCURSION

Rounding Errors Are a Loss of Precision

Rounding errors occur when a floating-point number such as 1.5 is assigned to a variable of type int (integer). The portion of the number following the decimal point is lost because an integer cannot represent a decimal value.

This can occur when the delta being applied to a point is 3; however, due to the position of the point within the object relative to the handle controlling the scale action, only 50 percent of the delta is applied.

The result is (3×0.5) or 1.5 applied to the data point.

Further, it is necessary to use integer variables to represent the points or data fields within the graphic objects because it is impossible to draw half a pixel; therefore, everything must be rounded to the nearest integer number or whole pixel.

As rounding errors compound, the data representing the object is affected, and, for instance, lines are not as straight as they should be.

The function implementing the scale transformation must account for rounding errors. One method to minimize the effect of rounding errors is to ensure that all transformations are done to the original points or data fields. In this way, the rounding errors will not compound. At worst only a fraction of a pixel is lost, as opposed to a fraction of a pixel being lost for each iteration of the scale transformation.

The following sections present the functions for scaling point-array–based and arc objects accounting for applying only a portion of the delta based on the handle selected and ensuring data integrity by minimizing rounding errors.

Scaling a Line

The code in Listing 11.3 is the entry point for accomplishing scaling of pointarray-based objects. As discussion focuses on the body of the function, keep in mind the tasks that must be accomplished by the scale transformation: namely, properly proportioning the delta applied and minimizing the effects of rounding errors.

```
Listing 11.3 The line scale Function
```

```
static void line scale( GXObjPtr line, XEvent *event )
1:
2:
      {
        static GXLinePtr tmp_data = NULL;
3:
4:
        GXLinePtr line_data = (GXLinePtr)line->data;
5:
6:
        if( tmp_data ) {
7:
          XDrawLines( XtDisplay(GxDrawArea), XtWindow(GxDrawArea), rubberGC,
8:
                      tmp_data->pts, tmp_data->num_pts, CoordModeOrigin );
9:
        } else {
          /* our first time... */
10:
11:
          (*line->erase)( line );
12:
          tmp data = (GXLinePtr)XtNew(GXLine);
13:
          tmp data->num pts = line data->num pts;
14:
15.
          tmp data->pts =
                 (XPoint *)XtMalloc( sizeof(XPoint) * tmp data->num pts );
16:
17:
          get_bounds( line_data->pts, line_data->num_pts,
                      &OrigX, &OrigY, &ExntX, &ExntY );
18:
19:
        }
20:
21:
        if( event ) {
22:
          memcpy( (char *)tmp_data->pts, (char *)line_data->pts,
23:
                  sizeof(XPoint) * tmp data->num pts );
24:
25:
          apply delta( tmp data->pts, tmp data->num pts,
26:
                       FixedX - event->xbutton.x,
27:
                       FixedY - event->xbutton.y );
28:
29:
           XDrawLines( XtDisplay(GxDrawArea), XtWindow(GxDrawArea), rubberGC,
30:
                       tmp data->pts, tmp data->num pts, CoordModeOrigin );
31:
        } else {
32:
          if( tmp data ) {
33:
            memcpy( (char *)line_data->pts, (char *)tmp_data->pts,
                    sizeof(XPoint) * line_data->num_pts );
34:
35:
36:
            XtFree((char *)tmp_data->pts);
37:
            XtFree((char *)tmp data);
38:
39:
            tmp data = NULL;
40:
          }
41:
        }
42:
      }
```

In reading through Listing 11.3, you have probably identified that the basic structure of the functions follows the management behavior of the move functions seen in previous sections.

It is still essential for the function to distinguish between the first iteration of a scale action and subsequent iterations.

11

As with the move functions seen in Listing 11.1 and Listing 11.2, the actual graphic object must be erased from the screen and the interactive rubber-banding object drawn instead.

The scale action, however, takes on another responsibility as well. Specifically, it maintains a pointer to a temporary GXLine structure

3: static GXLinePtr tmp_data = NULL;

for managing when to erase the rubber-banding object, and for ensuring that the transformation is only applied once to the original set of points defining the object.

Consider the body of the else, which like the move function's use of x and y indicates that this is the first iteration for the current scale action, as no tmp_data has been created (implicit null test).

In addition to ensuring that the object's method is invoked to erase the object, the body of the else creates a new reference to a GXLine structure and assigns this value to tmp_data.

13: tmp_data = (GXLinePtr)XtNew(GXLine);

EXCURSION

Introducing a New Function for Allocating Memory

The XtNew function is provided by the X Toolkit Intrinsics library. It is a function that could have an equivalent in C that looks like

```
tmp_data = malloc( sizeof(GXLine) );
```

However, this is a good time to introduce it. It is important to remember that any memory allocated by an application must at some point be freed.

The call to free the memory allocated with the XtNew function is XtFree as you'll see shortly.

The function then assigns to tmp_data the number of points it will manage and creates a valid array for storing the same number of points as the object being scaled:

```
14: tmp_data->num_pts = line_data->num_pts;
15: tmp_data->pts =
16: (XPoint *)XtMalloc( sizeof(XPoint) * tmp_data->num_pts );
```

The tmp_data structure stores the original points defining the object. This allows the scale function to apply the transformation to the temporary copy of the points without affecting the original object. Subsequent calls to the line_scale function result in the original points being copied into the tmp_data structure and again having the transformation applied to the temporary data structure, leaving the original data unaffected.

```
22: memcpy( (char *)tmp_data->pts, (char *)line_data->pts,
23: sizeof(XPoint) * tmp_data->num_pts );
24:
25: apply_delta( tmp_data->pts, tmp_data->num_pts,
26: FixedX - event->xbutton.x,
27: FixedY - event->xbutton.y );
```

The result is that only one transformation is ever applied to the data points in order to minimize the effects of rounding described earlier.

EXCURSION

Adding Functions to Your Memory Management Repertoire

The function memcpy is provided by the standard C library, and, as the name implies, serves to copy data from one area of memory to another.

In this context it is used to copy the original points contained in the line_data->pts array to the tmp_data->pts array.

The syntax of the call is

memcpy(destination, source, total size);

Because the memcpy function is written to transfer data of any type from the source address to the destination address, the structure references are cast to character points (char *) to satisfy the function's prototype found in the header

```
#include <stdlib.h>
```

Optionally, a for loop could be formed to explicitly copy all the points from one array to another.

Return to the body of the else indicating the first iteration of the scale action to examine the function call

17: get_bounds(line_data->pts, line_data->num_pts, 18: &OrigX, &OrigY, &ExntX, &ExntY);

The variables OrigX, OrigY, ExntX, and ExntY are global variables used to define the upper and lower bounds of the object being scaled. Their use will become clear when we examine the apply_delta function.

Before discussing apply_delta however, let's look at the use of FixedX and FixedY passed as parameters the apply_delta function.

Having reserved the introduction of the cursor management function that invokes the line_scale, line_move, and arc_move functions evaluated in this chapter, accept for the moment that the value of FixedX and FixedY are assigned the location of the object handle at the start of the scale action.

Unlike the move function, which applied an incremental delta to the points of the object, the scale functions apply the total delta with each iteration.

This is necessary because the original object data points are copied into the tmp_data structure with each call to the line_scale function. The goal is to affect the points a total of one time by the scale operation. So the total delta from the start of the operation is applied to the original data points, instead of an incremental delta being applied to previously scaled points.

To calculate the overall delta from the start of the operation, the variables FixedX and FixedY record the location of the ButtonDown event that started the scale action. The values of FixedX and FixedY are then subtracted from the current event point to determine the distance the mouse cursor has moved since the start of the operation.

Now that you understand how the parameters to apply_delta are formed and the purpose they serve, consider Listing 11.4, which introduces the apply_delta function declaration.

Listing 11.4 The apply_delta Function

```
1:
      void apply delta( XPoint *data, int num, int dx, int dy )
2:
      {
3:
4:
        switch( GxActiveHandle ) {
5:
        case 0:
6:
          apply delta top left( data, num, dx, dy );
7:
          break;
8:
        case 1:
          apply_delta_top_side( data, num, dx, dy );
9:
10:
          break;
        case 2:
11:
          apply_delta_top_right( data, num, dx, dy );
12:
13:
          break;
        case 3:
14:
15:
          apply delta right side( data, num, dx, dy );
16:
          break;
17:
        case 4:
          apply delta bottom right( data, num, dx, dy );
18:
19:
          break;
        case 5:
20:
          apply delta bottom side( data, num, dx, dy );
21:
22:
          break;
23:
        case 6:
          apply_delta_bottom_left( data, num, dx, dy );
24:
25:
          break;
```

```
26: case 7:
27: apply_delta_left_side( data, num, dx, dy );
28: break;
29: default:
30: setStatus( "LINE: The end is nigh!" );
31: }
32: }
```

The apply_delta function invokes the correct apply_delta_<direction> based on the current GxActiveHandle value. The GxActiveHandle is set when the FixedX and FixedY values are assigned at the start of the scale action.

Listing 11.5 introduces the apply_delta_<direction> functions.

```
Listing 11.5 The apply_delta_<direction> Functions
```

```
1:
      static void
2:
      apply_delta_bottom_side( XPoint *pts, int num_pts, int dx, int dy )
3:
      {
4:
        int i;
5:
        if( ExntY == 0 ) return;
6:
7:
        for( i = 0; i < num pts; i++ ) {</pre>
8:
9:
          pts[i].y -= (int)( dy *
10:
                               ((float)(pts[i].y - OrigY) / (float)ExntY));
11:
        }
12:
      }
13:
      static void
      apply delta right side( XPoint *pts, int num pts, int dx, int dy )
14:
15:
      {
16:
        int i;
17:
        if( ExntX == 0 ) return;
18:
19:
        for( i = 0; i < num pts; i++ ) {</pre>
20:
21:
          pts[i].x -= (int)( dx *
22:
                               ((float)(pts[i].x - OrigX) / (float)ExntX));
23:
        }
24:
      }
25:
      static void
      apply_delta_top_side( XPoint *pts, int num_pts, int dx, int dy )
27:
28:
      {
29:
        int i;
30:
        if( OrigY == 0 ) return;
31:
32:
        for( i = 0; i < num pts; i++ ) {</pre>
33:
          pts[i].y +=
34:
            (int)( dy * ((float)(pts[i].y - ExntY) / (float)OrigY));
35:
        }
36:
      }
```

11

Listing 11.5 Continued

```
37:
      static void
38:
      apply delta left side( XPoint *pts, int num pts, int dx, int dy )
39:
      {
40:
        int i;
41:
42:
        if( OrigX == 0 ) return;
43:
44:
        for( i = 0; i < num pts; i++ ) {</pre>
45:
          pts[i].x +=
45:
            (int)( dx * ((float)(pts[i].x - ExntX) / (float)OrigX));
46:
        }
47:
      }
48:
      static void
      apply delta bottom right( XPoint *pts, int num pts, int dx, int dy )
49:
50:
      {
51:
        apply_delta_right_side( pts, num_pts, dx, dy );
52:
        apply_delta_bottom_side( pts, num_pts, dx, dy );
53:
      }
54:
      static void
55:
      apply_delta_bottom_left( XPoint *pts, int num_pts, int dx, int dy )
56:
      {
57:
        apply_delta_bottom_side( pts, num_pts, dx, dy );
58:
        apply_delta_left_side( pts, num_pts, dx, dy );
59:
      }
60:
      static void
61:
      apply_delta_top_right( XPoint *pts, int num_pts, int dx, int dy )
62:
      {
63:
        apply delta top side( pts, num pts, dx, dy );
64:
        apply_delta_right_side( pts, num_pts, dx, dy );
65:
      }
66:
      static void
67:
      apply delta top left( XPoint *pts, int num pts, int dx, int dy )
68:
      {
69:
        apply_delta_top_side( pts, num_pts, dx, dy );
70:
        apply_delta_left_side( pts, num_pts, dx, dy );
71:
      }
```

Only four directions require calculations when applying the scale delta to a line object. They are apply_delta_right_side, apply_delta_left_side, apply_delta_top_side, and apply_delta_bottom_side.

The functions shown in Listing 11.5 that combine directions are straightforward and require no explanation.

Consider instead the function

2: apply_delta_bottom_side(XPoint *pts, int num_pts, int dx, int dy)

After declaring an integer for looping,

4: int i;

the function tests the ExntY value to ensure that it is non-zero. This prevents a fatal divide by zero, as you'll see shortly.

5: if(ExntY == 0) return;

The function continues by looping through all the points in the points array pts. Remember, this array was tmp_data->pts as passed by the calling function, meaning the values being altered are a copy of the original points.

Because the function's purpose is to apply only the scale delta to the bottom of the object, only the y values are affected.

The variables dx and dy were passed from the calling function as the result of the evaluation of

```
FixedX - event->xbutton.x
and
FixedY - event->xbutton.y
```

To determine the proportion of the dy value to apply to each data point, the ratio of the point's distance from the origin of the object and the extent of the object is used.



The variable ExntY (globally set in the calling function line_scale) is the y value of the lower-right corner of the object. Similarly, OrigY is the upper-left corner of the object.

The effect of this calculation is that when the bottom handle (or either bottom corner handle) is used to control the scale action, the farther from the origin of the object the data point lies the greater the portion of the delta applied to that data point.

The same ratio is used in the other *side functions* differing only in which component of the delta is applied as dictated by the direction of the scale action. In other words, the left and right side actions only apply the x delta (dx) to the x component of the data points and the top, like the bottom, scale direction only affects the y value by some portion of the y delta (dy).



The reference to side functions for applying the scale delta is meant to generically refer to any of the apply_scale_<direction> functions, where direction is top_side, bottom_side, left_side, and right_side.

A final difference is that the bottom scale direction deducts the calculated portion of the delta, and the top scale direction adds the portion that is determined by the ratio of the point's distance from the extent relative to the origin. Effectively, the top scale direction inverts the bottom scale operation as is necessary for scaling in the opposite direction. For the same reason, the left and right scale operations are inverted.

Having successfully applied the delta to the sides dictated by the active scale handle, the line_scale function draws the rubber-banding object to the canvas window, as seen in Listing 11.3, lines 29–30:

29: XDrawLines(XtDisplay(GxDrawArea), XtWindow(GxDrawArea), rubberGC, 30: tmp_data->pts, tmp_data->num_pts, CoordModeOrigin);

At some point the line_scale function will end as indicated by the passing of a null event point. When this happens, the second else statement is evaluated

```
31: } else {
```

If there is a valid tmp_data structure determined by an implicit test for null

32: if(tmp_data) {

the contents of the temporary storage structure's XPoint array pts are transferred to the line object's data pts array

33: memcpy((char *)line_data->pts, (char *)tmp_data->pts, 34: sizeof(XPoint) * line_data->num_pts);



Remember that the points contained in the temporary structure were only transformed once. The points transferred to the line object are the points gained from the last iteration of the line_scale function.

Because the line_scale function is ending, the tmp_data reference is no longer needed. Therefore, it is appropriate to free the memory allocated for its use.

```
36: XtFree((char *)tmp_data->pts);
37: XtFree((char *)tmp_data);
```



When freeing *nested pointers* such as the pts field contained within the allocated tmp_data structure, you must free them in the reverse order that they were allocated.

This is necessary because after a pointer is freed it can no longer be legally referenced.

In others words, it would be a dangerous (and potentially fatal) mistake to free tmp_data and then reference it to free the pts field.

When freed, the value referenced by the tmp_data variable is no longer valid. Therefore set its value back to NULL in preparation for a subsequent scale action request.

39: tmp_data = NULL;

Look now at the arc equivalent to the line_scale function shown in Listing 11.3.

Scaling an Arc

The arc_scale function shown in Listing 11.6 is nearly identical in its structure to the line_scale function seen in Listing 11.3.

Listing 11.6 The arc_scale Function

```
1:
      static void arc_scale( GXObjPtr arc, XEvent *event )
2:
      {
3:
        static XArc *tmp data = NULL;
4:
5:
        if( tmp data ) {
6:
          /*
7:
           * erase the rubberband arc
           */
8:
         XDrawArc( XtDisplay(GxDrawArea), XtWindow(GxDrawArea), rubberGC,
9:
10:
                    tmp data->x, tmp data->y,
                    tmp data->width, tmp data->height,
11:
                    tmp_data->angle1, tmp_data->angle2 );
12:
13:
14:
        } else {
15:
          /*
16:
           * our first time through - erase the actual arc
17:
           */
          (*arc->erase)( arc );
18:
19:
20:
          tmp_data = (XArc *)XtNew( XArc );
21:
        }
22:
        if( event ) {
23:
24:
          memcpy( (char *)tmp data, (char *)arc->data, sizeof(XArc));
```

11

```
Listing 11.6 Continued
```

```
25:
          apply delta( tmp data,
                        FixedX - event->xbutton.x,
26:
27:
                        FixedY - event->xbutton.y );
28:
29:
          /*
           * draw a rubberband arc
30:
           */
31:
          XDrawArc( XtDisplay(GxDrawArea), XtWindow(GxDrawArea), rubberGC,
32:
33:
                     tmp_data->x, tmp_data->y,
34:
                     tmp data->width, tmp data->height,
35:
                     tmp_data->angle1, tmp_data->angle2 );
36:
37:
        } else {
38:
          if( tmp_data ) {
             memcpy( (char *)arc->data, (char *)tmp_data,
39:
40:
                     sizeof(XArc));
41:
42:
            XtFree( (char *)tmp_data );
            tmp data = NULL;
43:
44:
          }
45:
        }
46:
      }
```

Comfortable with the structure of this function based on the study of the line_scale function, note the changes in data references to account for the different object type.

Listing 11.7 shows the apply_delta function for applying the scale delta to the arc object. It, too, is nearly identical in structure to the apply_delta function used by the line_scale function.

EXCURSION

A Review of the Graphics Editor Project Structure

As demonstrated in the Graphics Editor program structure seen in Chapter 6, "Components of an X Window Application," there is total separation of source code for line, arc, and text object routines.

This, as indicated in Appendix B, "Application Layout Code Listing," is accomplished by having multiple program files. In this way the code controlling line objects is placed in a different file than the code for arcs.

Further, through use of the keyword static within the files containing the source code for the various objects, the function names can be reused for routines of similar purpose.

This is the case with the apply_delta function. As is made clear in Chapter 13, "Application Structure," the resolution of the varying apply_delta functions is determined by the version of the function co-located in the file containing the source code for the different objects.

```
Listing 11.7 The apply delta Function for Arcs
```

```
static void apply delta( XArc *data, int dx, int dy )
1:
2:
      {
3:
        switch( GxActiveHandle ) {
4:
        case 0:
5:
          apply_delta_top_left( data, dx, dy );
6:
          break;
7:
        case 1:
8:
          apply_delta_top_side( data, dx, dy );
9:
          break;
10:
        case 2:
11:
          apply_delta_top_right( data, dx, dy );
12:
          break;
13:
        case 3:
          apply delta right side( data, dx, dy );
14:
15:
          break;
16:
        case 4:
17:
          apply_delta_bottom_right( data, dx, dy );
18:
          break;
19:
        case 5:
20:
          apply_delta_bottom_side( data, dx, dy );
21:
          break;
22:
        case 6:
23:
          apply delta bottom left( data, dx, dy );
24:
          break;
25:
        case 7:
26:
          apply delta left side( data, dx, dy );
27:
          break;
28:
        default:
29:
          /* -shouldn't- happen in my time */
30:
          setStatus( "ARC: The end is nigh!" );
31:
        }
32:
      }
```

Though similar in structure to the apply_delta function employed by the line_scale function, some subtle differences do exist, specifically, the parameter lists for apply_delta_<direction> functions.

Listing 11.8 shows the definition of these functions for use by the arc object.

```
Listing 11.8 The Arc apply_delta_<direction> Functions
```

```
static void apply_delta_top_side( XArc *data, int dx, int dy )
1:
2:
       {
3:
         int y1, y2;
4:
5:
         y1 = data ->y;
6:
         y2 = data->y + data->height;
7:
8:
         data -> y = min(y1 - dy, y2 + dy);
9:
         data \rightarrow height = max(y1 - dy, y2 + dy) - data \rightarrow y;
```

11

Listing 11.8 Continued

```
10:
      }
11:
      static void apply delta right side( XArc *data, int dx, int dy )
12:
      {
13:
        int x1, x2;
14:
15:
        x1 = data - x;
16:
        x^2 = data - x + data - width;
17:
18:
        data - x = min(x1 + dx, x2 - dx);
19:
        data \rightarrow width = max(x1 + dx, x2 - dx) - data \rightarrow x;
20:
      }
21:
      static void apply delta top right( XArc *data, int dx, int dy )
22:
      {
23:
        apply delta right side( data, dx, dy );
24:
        apply_delta_top_side( data, dx, dy );
25:
      }
26:
      static void apply_delta_bottom_side( XArc *data, int dx, int dy )
27:
      ł
28:
        int y1, y2;
29:
30:
        y1 = data ->y;
        y2 = data->y + data->height;
31:
32:
33:
        data - y = min(y1 + dy, y2 - dy);
34:
        data \rightarrow height = max(y1 + dy, y2 - dy) - data \rightarrow y;
35:
      }
36:
      static void apply delta bottom right( XArc *data, int dx, int dy )
37:
      {
38:
        apply delta bottom side( data, dx, dy );
39:
        apply_delta_right_side( data, dx, dy );
40:
      }
41:
      static void apply delta left side( XArc *data, int dx, int dy )
42:
      {
43:
        int x1, x2;
44:
45:
        x1 = data - x;
46:
        x^2 = data - x + data - width;
47:
48:
        data - x = min(x1 - dx, x2 + dx);
49:
        data - width = max(x1 - dx, x2 + dx) - data - >x;
50:
      }
51:
      static void apply delta bottom left( XArc *data, int dx, int dy )
52:
      {
53:
        apply delta bottom side( data, dx, dy );
54:
        apply_delta_left_side( data, dx, dy );
55:
      }
56:
      static void apply_delta_top_left( XArc *data, int dx, int dy )
57:
      {
58:
        apply_delta_top_side( data, dx, dy );
59:
        apply_delta_left_side( data, dx, dy );
60:
      }
```

As with line scaling, only the side functions actually perform calculations. The corner functions simply invoke the correct combination of side routines to accomplish their task.

Consider the function

1: static void apply_delta_top_side(XArc *data, int dx, int dy)

for applying the scale delta to the top side of the arc object. Evaluating it will shed light on how all the arc side functions work because the relationships follow those specified with the line side functions

```
5: y1 = data->y;
6: y2 = data->y + data->height;
7:
8: data->y = min( y1 - dy, y2 + dy );
9: data->height = max( y1 - dy, y2 + dy ) - data->y;
```

By storing the upper and lower corner y values, the apply_delta_top_side function can determine the new values for the y and height field of the XArc structure.

The logic applied here is that the minimum point between the upper-right value *less* the delta and the lower-right value *plus* the delta will be the new y value for the arc.

This *minimum* evaluation of the points is important for determining when the user has, for instance, selected the handle in the lower-right corner and then dragged it to a value less than the upper-right corner of the object.

In other words, if the scale action *flips* the arc object, the apply_delta_<*direction*> routine must account for this, and not allow the y value to be less than the y + height value.

A look at the remaining *side* functions shows that the logic is the same, changing only the components affected and the direction in which the delta is applied. For instance, left and right side functions change only x and width values, where the top and bottom affect only the y and height values.

As you advance in the development of the Graphics Editor project and apply the concepts introduced in this chapter, the ideas will become clearer.

The last transformation to consider before doing so, however, is rotation.

Rotating

As the name implies, the rotation transformation revolves objects around a center point. For the purpose of the context editor, this center point will coincide with the center of the object as is demonstrated in the next section.

Rotating a Line

Rotating a point-based object such as a line is most simply done by rotating each individually.

Figure 11.3 shows the relationship of a point to a 90-degree angle. By representing a point in this way, the dissection of the right angle allows a relationship to be derived for performing the rotation.



The point (x, y) in Figure 11.3 is the point prior to rotation and point (x^1, y^1) is the point after rotation. To move from (x, y) to (x^1, y^1) , you must apply the definition of the sine and cosine of an angle.

As you might recall from early trigonometry study, sine is defined as a function that maps an angle to the y coordinate of the angle's intersection with the unit circle. Similarly, cosine is the function that maps an angle to the x coordinate of the intersection.

Applying these definitions, the points x and y can be represented in terms of the angle > and radius r as

```
x = r \cos >
y = r \sin >
```

Therefore, to determine the values of x^1 and y^1 you only need to add the angle = to > as seen below.

```
x^{1} = r \cos (> + =)
y^{1} = r \sin (> + =)
```

Solving for x¹

```
r \cos(> + =) = r \cos > \cos = + r \sin > \sin =
```

Substituting the value of x for $(r \cos >)$ and y for $(r \sin >)$ yields

 $x^1 = x \cos x = -y \sin x$

Solving for y^1 in the same way

 $r \sin(> + =) = r \sin > \cos = + r \cos > \sin =$

or

 $y^1 = y \cos = +x \sin =$

There are only a couple more issues to resolve before coding the solution for rotating a point.

First, the proof shown for calculating x^1 and y^1 assumes that the angle's origin is at (0, 0). Second, as demonstrated in Figure 11.3, the point (x, y) does not sit exactly on the perimeter of the circle.

Both can be resolved by applying a principle introduced in Chapter 10, "Trigonometric and Geometric Functions." All points are a constant distance from the foci of a circle. Therefore, the point before rotation must be transformed to an origin of (0, 0) and the point after rotation must be transformed to a point equal to the magnitude of the original point's distance from the circle the point is perceived to sit on.

→ See Chapter 10, page 218, for a discussion of **point and arc intersection**.

Listing 11.9 demonstrates the implementation of rotating a point through an angle.

Listing 11.9 The rotate_point Function

```
1:
      #define gxround(a)
                                 (int)(((a)<0.0)?(a)-.5:(a)+.5)
2:
      void rotate point( XPoint *pt, int deg, double cx, double cy )
3:
      {
4:
        double dx, dy, theta, cosa, sina, mag;
5:
6:
        dx = (double)pt -> x - cx;
7:
        dy = cy - (double)pt ->y;
8:
        if (dx == 0.0 && dy == 0.0) {
9:
10:
          return;
11:
        }
12:
13:
        if( deg == 0 ) return;
15:
16:
        theta = gx compute angle( dx, dy );
17:
        theta -= (deg * M PI / 180.0 );
18:
```

11

```
Listing 11.9 Continued
```

```
19:
        if (theta < 0.0) {
20:
          theta += M 2PI;
21:
        } else if (theta >= M_2PI - 0.001) {
22:
          theta -= M 2PI;
23:
        }
24:
25:
        mag = sqrt(dx * dx + dy * dy);
26:
27:
        \cos a = \max ( \cos ( \sinh a ) ;
28:
        sina = mag * sin(theta);
29:
30:
        pt - x = gxround(cx + cosa);
31:
        pt->y = gxround(cy - sina);
32:
      }
33:
      #undef gxround
```

Listing 11.9 begins by defining a macro for rounding a number

1: #define gxround(a) (int)(((a)<0.0)?(a)-.5:(a)+.5)

The function itself expects that a pointer to the point being rotated be passed as the first parameter, the degrees that the point should be rotated as the second, and lastly the coordinates of the center point of the object containing the point

```
2: void rotate_point( XPoint *pt, int deg, double cx, double cy )
```

The variables employed by the function are of the data type double

```
4: double dx, dy, theta, cosa, sina, mag;
```

→ As you might recall from Chapter 2, "Programming Constructs," section "Data Types," page 70, the data type double is significantly more precise than float.

Precision during rotation is a serious issue. Consider the quantity of floating point operations required to accomplish the rotation of a point. Further, moving the floating point results to an integer variable (whole pixel representation) inherently causes further loss of precision. The use of the data type double aids in minimizing loss of precision during the rotation calculations.

Next, the rotate_point function must calculate the foci in the same way it was done in Chapter 10.

6: dx = (double)pt ->x - cx;7: dy = cy - (double)pt ->y;

The angle > is then calculated and converted to radians:

16: theta = gx_compute_angle(dx, dy); 17: theta -= (deg * M_PI / 180.0); The function gx_compute_angle applies the definition of sine and cosine by working backward to determine the angle based on the intersection points. The gx_compute_angle is introduced later.

Bounds checking is performed on the angle theta to ensure that it is in the proper range for the rotation operation.

```
19: if (theta < 0.0) {
20: theta += M_2PI;
21: } else if (theta >= M_2PI - 0.001) {
22: theta -= M_2PI;
23: }
```

The magnitude of the foci is calculated so that it can be applied to the new point resulting from the rotation

25: mag = sqrt(dx * dx + dy * dy);

The actual rotation of the point is calculated

```
27: cosa = mag * cos(theta);
28: sina = mag * sin(theta);
```

Finally, the value of the rotation point is assigned to the XPoint structure pointer, replacing the previous values of the point with the rotated values.

```
30: pt->x = gxround(cx + cosa);
31: pt->y = gxround(cy - sina);
```

Notice that in conjunction with the assignment of the rotated point to the XPoint structure, the point is transformed relative to the center coordinate of the object containing the data point. Also, the gxround macro is employed to ensure that the values are rounded up to nearest whole pixel representation.

I show the management function that invokes the rotate_point for pointarray-based objects shortly, but first consider the concept of rotating arcs.

Rotating an Arc

Because rotating a circle of 360 degrees has no meaning, the issue of rotating an arc is limited to ellipses and arcs of less than 360 degrees.

Rotating an Ellipse

As a caveat at the end of Chapter 10, the X Window System is incapable of representing an ellipse with a major and minor axis that is not parallel with the x and y axes.

Therefore, the rotation of an ellipse is not supported by the Graphics Editor projects.

Rotating Arcs Less than 360 Degrees

The X Window System inherently supports the rotation of an arc by management of the values of angle1 and angle2 provided in the XArc structure.

→ If necessary, review the definitions of these fields as introduced with the XDrawArc primitive in Chapter 7, section "XDrawArc," page 192.

Next Steps

The next chapter will introduce *coordinate systems* and illustrate the need to understand the system in which graphic transformations are performed. This is not important to the success of the Graphics Editor project, but will provide clarity to the issue of graphic transformations.

Demonstrated with the use of the X Window System primitives and X Window creation in Chapters 7 and 8, the origin is consistently in the upper-left corner of the screen. However, when the mathematics for rotating a point is discussed, the origin relative the operation was silently made the lower-left corner (refer to Figure 11.3).

Determining the origin for an operation or calculation is a factor of the coordinate system being employed.

Following the brief discussion on coordinate systems, the foundation will be complete, and full attention will focus on furthering the Graphics Editor.

Chapter 12

In this chapter

- Rectangular Coordinates
- Polar Coordinate System
- Next Steps

Coordinate Systems

The concept of coordinate systems must be addressed to make a book on the X Window System and graphic transformations complete.

This chapter introduces coordinate systems and programmatic considerations for each of them.

As the last building block in the foundation preparing us to focus solely on programming the Graphics Editor, this chapter marks the end of Section One.

A *coordinate system* determines the positive and negative directions of the x and y axes and their relationship to horizontal and vertical placement. Further, a coordinate system dictates the location of the origin or (0,0) position within the representation.

In computer graphics, a primary concern is in locating points on the surface of the canvas or drawing area. To specify a location requires knowledge of the coordinate system in use.

In a windowing environment, coordinates within the system are *discrete*, meaning that they represent the pixels used to position the elements bearing the coordinates.

Every Drawable (Window or Pixmap) in the X Window environment maintains its own coordinate system or management of element placement within its bounds.

The coordinate system employed by the Drawables in the X Window environment has its origin in the upper-left corner of the Drawable with the x-axis increasing to the right and the y-axis increasing downward, as depicted in Figure 12.1.

Figure 12.1 The coordinate system used in the X Window environment.

Within the X Window System's coordinate system, negative numbers can be used to represent coordinates; however, negative coordinates affect visibility.

The following sections introduce two common coordinate systems employed in computer graphics and windowing environments.

Rectangular Coordinates

A *rectangular coordinate system* has horizontal and vertical axes that are right angles to each other. The coordinates within the rectangular coordinate system are evenly spaced in both the x and y directions. The origin of the rectangular coordinate system is at the junction of the two axes, as shown in Figure 12.2.

Figure 12.2

The rectangular coordinate system.



The rectangular coordinate system is also known as the *Cartesian coordinate system*. It is commonly used in digital graphics systems because of its easy adaptation to raster displays.

→ Review Chapter 8, section "Raster Graphics," page 199, for a review of vector versus raster graphics.

Polar Coordinate System

The *polar coordinate system* is rarely used in graphics, because it is less valuable than the rectangular coordinate system for mapping pixels to a display.

A polar coordinate system defines angles and distances rather than scalar values.



A *scalar value* in this context refers to an integer data type that is scaled in a linear manner.

The coordinate points in a rectangular coordinate system are known as scalar values, whereas the polar coordinate system uses a combination of angles and distances to indicate placement in system.

In the polar coordinate system, the position of a point (coordinate) is defined as the angular deflection of a line.

The endpoints of the line are formed from the origin and the specified location of the point.

To represent a location in the polar coordinate system, points are represented in terms of an angle called *theta* (θ) and a radius (the length of the line) referred to as *rbo* (ρ).

Figure 12.3 illustrates the relationship between theta and rho within the polar coordinate system



Applying what you now understand about coordinate systems, it should be obvious why both coordinate systems are useful to software engineers.

The placement of windows and the drawing of primitives in the X Window System environment uses a coordinate system very similar to and obeying many of the same properties as the rectangular coordinate system.

However, accomplishing graphic transformation such as rotation is most easily done using the polar coordinate system.

Related to the employment of a coordinate system, software engineers also must be aware of measuring conventions used within the system. This issue will present itself again when adding a print driver to the Graphics Editor application because the representation of coordinates on a screen or monitor is affected by units of measure and scale factors when transferring the coordinates to paper using the PostScript language.

Next Steps

This chapter concludes Section One, "Starting Points," and the foundation I thought imperative before approaching the Graphics Editor Project.

As you begin the Section Two, "Graphics Editor Application," you should be adequately prepared, either directly by skipping over the Starting Points or indirectly by spending time with the introduction the initial chapters provide.

Graphics Editor Application

This section focuses strictly on developing the Graphics Editor project.

With a solid understanding of the concepts necessary for accomplishing C and X Window System programming provided in Section One, I pay less attention to the details of the language and environment and more attention to the structure and execution of the Graphics Editor project.

Where to Begin

Unlike Section One, this section does not afford the flexibility to start anywhere except at the beginning. Each chapter in this section builds on the previous one.

As the project advances, code samples reflect only the changes to functions or files previously introduced, when possible to do so without sacrificing clarity.

What's at the End

At the completion of Section Two, you will have a functional, two-dimensional graphical editor capable of drawing, moving, scaling, rotating, and altering the attributes of a variety of graphic objects.



Laying Out the Parts

Chapter 13

In this chapter

- Project Structure
- Parsing the Command Line
- The getopt Function
- *The* XtVaGetApplicationResources *Function*
- Setting Up a Canvas
- Building the Project
- Next Steps

Application Structure

This chapter launches the Graphics Editor project by structuring the main program file and creating the graphical user interface.

Before proceeding, it is important that, independent of the knowledge and experience you bring to the project, you have reviewed the necessary sections in Part I to prepare adequately for the information presented here.

Project structure and organizations are critical elements of software design. As the Graphics Editor project progresses, it will grow in complexity very quickly.

The goal of this chapter is to lay out the application structure, including graphical user interface, parsing the command line, and finally, configuring the Canvas for receiving and processing the X events that will control user actions.

Because much of the source code for structuring the application is introduced in Chapter 6, focus here is given mainly to new ideas.



As new features are added to code listings already introduced, I will show *incremental listings*. An incremental listing generally shows only the line before and after the lines being inserted.

When starting a new application, you should first consider how the code will be structured based on elements present in the project. Second, you should consider how the application will be structured.

Project Structure

Structuring the code addresses the issue of where source and header files will reside relative to each other. Addressed as well is the placement of the object and executable files.

I∎ **Note**

The directory structure for the Graphics Editor was introduced in Figure 1.11. If you haven't reviewed make file syntax (discussed in Chapter 1, "UNIX for Developers," in the section "Makefile") and how to structure the project correctly, as discussed in the "make.defines" section of Chapter 1, you should do so now.

The layout and project management presented in Chapter 1 is intended for multiple-platform support. In other words, through use of the GNU make utility and directory layout introduced in Chapter 1, the project can be built successfully using any version of the UNIX operating system.

Table 13.1 shows the organization of the application source code by describing the contents of each of the files contained in the project and showing a relative path for their placement.

File	Purpose
<pre>src/GNUmakefile</pre>	Configuration file for GNU gmake utility. Defines source files, virtual paths, and build options for project management.
make.defines	Included by the GNUmakefile to set flags, paths, and variables required by the GNUmakefile.
<pre>src/gxMain.c</pre>	Defines the program entry point, initializes a connection to the X Server, and manages control of the graphical user interface.
src/gxGraphics.c	Holds functions invoked from gxMain for creating the user interface and defines and assigns global variables used during program exe- cution.
<pre>src/gxGx.c</pre>	Functions used for controlling graphic objects created by the editor and managing events for the widgets of the graphical user interface.
<pre>src/include/gxGraphics.h</pre>	Declares macros, global variables, and structures used by all source files.
<pre>src/include/gxIcons.h</pre>	Contains all bitmaps used as icons for the various buttons of the graphical user interface.
<pre>src/include/gxProto.h</pre>	Contains prototypes for all functions called externally from the file in which they are declared.
<pre>src/gxLine.c</pre>	Contains GXLine object creation and method definitions.

Table 13.1 Graphics Editor Project Layout

<pre>src/gxArc.c</pre>	Contains GXArc object creation and method definitions.
<pre>src/gxText.c</pre>	Contains GXText object creation and method definitions.
<pre>src/include/vfonts</pre>	Contains files necessary for accomplishing the vector fonts used by
	the GXText object.

The descriptions provided in Table 13.1 will serve as a guide for how the source code of the Graphics Editor project is structured.

Because the files introduced in Table 13.1 are relative, it is expected that a project root directory (value of GxHome variable in the make.defines) be decided before implementing the structure to support the introduction of the files in the project.

Let us focus now on the structure of the application by deciding how best to lay out the graphical user interface.



.

-

. .

- ----

Note

Making the decision of *what goes where* on a graphical user interface requires understanding the features that the application will support.

Attention to the feature list will enable you to determine whether the required number of icons can be placed on the interface or whether their number requires a menu system or some management method to prevent the application from becoming cluttered.

Table 13.2 shows the feature list that the graphical user interface of the Graphics Editor must support.

lable	a 13.2 Graphics Editor Feature List	
Creation Functions		
	Latex Line Object	
	Pencil Line Object	
	Arc Object	
	Box Object	
	Arrow Object	
	Text Object	
Management Functions		
	Сору	
	Cut	

Table 13.2	continued	
Management Functions		
	Print	
	Export	
	Save	
	Load	
Assignin	g Attributes	
	Foreground color	
	Background color	
	Fill Style	
	Line Width	
	Degrees of Rotation	
Miscella	neous	
	Exit	

Great care must be given to the production of *any* graphical user interface to ensure that elements of aesthetics and intuitiveness are well balanced.

Intuitiveness

If features are hidden from the users or not logical in their placement within the user interface, users will probably not use them much. This has an obvious and direct impact on the impression of quality assigned to the application.

Aesthetics

The aesthetics of the interface is as important as creating an interface that is intuitive in its use. The attractiveness of the design has equal impact on the impression of quality.

Addressing the issues of creating an intuitive and aesthetically pleasing interface based on the features that must be supported by the application, you can begin to sketch either mentally or literally what the interface should look like.

Figure 13.1 shows my mental image of what the interface should look like.



Notice in Figure 13.1 the incredible clarity of my thoughts.

13





The only feature listed in Table 13.2 not addressed in the proposed interface shown in Figure 13.1 is a mechanism for changing the attributes of objects.

To address this requirement, a menu listing the possible attributes will be presented to the user when she right-clicks over an active object.

This approach does not adhere to the requirement of being initially intuitive because the user will not know that the menu exists until instructed of its presence. The solution does offer two things, however. First is the chance to demonstrate programmatically how to create pop-up and cascading menus and second is a tidy workspace that is not littered with icons that are not general to the entire application.

The following section shows how to create the interface presented in Figure 13.1.

Laying Out the User Interface

The first file to create for the Graphics Editor is the gxMain.c file, which controls the entry point and creation of the interface components. This file is the easiest to create for the project because the functions it invokes are external to the file.

Listing 13.1 shows the contents of the gxMain.c file.

Listing 13.1 The gxMain.c File

```
/**
1:
        ** 2D Graphical Editor (2d-gx)
2:
        * *
3:
        ** gxMain.c
4:
         */
5:
6:
       #include <stdlib.h>
7:
       #include <stdio.h>
8:
9:
       #include <X11/Intrinsic.h>
                                      /* for creation routines */
10:
       #include <X11/StringDefs.h> /* resource names
                                                                 */
       #include <X11/Xaw/Form.h>
11:
12:
13:
       #include "gxGraphics.h"
14:
       #include "gxProtos.h"
```
Listing 13.1 continued

```
15:
16:
       /*
        * Entry point for the application
17:
18:
        */
19:
       int main( int argc, char **argv )
20:
       {
21:
         XtAppContext appContext;
22:
         Widget
                       toplevel;
         Widget
                       form;
23:
24:
         Widget
                       canvas;
25:
         toplevel = XtVaAppInitialize( &appContext, "2D Graphical Editor",
26:
27:
                                         NULL,0, &argc, argv, NULL,
28:
                                         NULL );
29:
30:
         form = XtVaCreateWidget( "topForm",
                                     formWidgetClass, toplevel,
31:
                                    NULL );
32:
33:
34:
         canvas = create canvas( form );
35:
         create status( form, canvas );
36:
         create_buttons( form );
37:
38:
         XtManageChild( canvas );
39:
         XtManageChild( form );
40:
41:
         XtRealizeWidget( toplevel );
43:
         XtAppMainLoop( appContext );
44:
45:
         exit(0);
46:
       }
47:
48:
       /**
49:
        ** end of gxMain.c
50:
        */
```

Listing 13.1 begins, following some comments, by including header files from various sources.

The first two come from the C environment and the delimiters < and > surrounding the filename provide a hint for where the C preprocessor should look for them:

6: #include <stdlib.h>
7: #include <stdio.h>

These header files provide most of the commonly used built-in functions provided by the C language and are generally included at the beginning of all C source files.

The next few included header files are supplied by the X Window System development environment:

```
9: #include <X11/Intrinsic.h> /* for creation routines */
10: #include <X11/StringDefs.h> /* resource names */
11: #include <X11/Xaw/Form.h>
```

As the comments following the preprocessor include directives indicate, Intrinsic.h contains the prototypes for the X initialization and widget creation routines used by the functions in this file. The StringDefs.h file resolves the resource names used to configure the attributes of the widgets created and, finally, the Form.h header file contains the form widget class definition, and therefore must be included before instantiating a widget of type formWidgetClass.

EXCURSION

Including a Path Component with Header Files

Header files often contain path components in addition to the actual filename as seen with the three header files included from the X environment.

The convention is to specify only to the compiler the *home* directory of header locations when using the compiler flag - I. From the home or root component the compiler is instructed to search, subdirectories are nested with the header filename.

For example, the compiler flag -I used to compile the file shown in Listing 13.1 would include the path component

-I /usr/include

for the Linux operating system or

-I /usr/openwin/include

under the Sun Solaris operating system.

Then the remaining path components needed by the preprocessor for finding the file are embedded in the include directive as seen with X11/Intrinsic.h, X11/StringDefs.h, and X11/Xaw/Form.h.

In other words, the actual path under Linux for finding the file Form.h is /usr/include/ X11/Xaw. A portion of this is provided to the compiler by merit of the -I flag, and the remainder is nested with the header filename.

Notice that the header files provided by X, like those provided by C, are delimited with < and > symbols. The following two header files, however, use double quotes (" ") to contain the filenames, indicating to the preprocessor that the files are local.

13: #include "gxGraphics.h"
14: #include "gxProtos.h"

Although the C preprocessor will find the files regardless of how the filenames are delimited, the search order is determined by these delimiters and therefore affects the speed with which the files are located.

Another advantage of correctly delimiting the filenames is to ensure that anyone who follows behind you will know where to begin looking as well.

The header file gxProtos.h, however delimited, is included to ensure that the functions invoked in this file but defined elsewhere in the project are correctly prototyped.

As you scan further through Listing 13.1 you will identify create_canvas, create_status, and create_buttons as functions unique to the Graphics Editor project. The forward declaration for these is contained in gxProtos.h and required before the compiler can successfully check whether the parameter types and return values are consistent with the usage. This level of error checking done by the compiler prevents many semantic and syntactical errors introduced by the programmer.

Because many of the functions prototyped in gxProtos.h require references to structures defined by the project, the inclusion of gxGraphics.h provides the necessary type definitions to satisfy the compiler. This implies that the order the files are included is important as well, as the compiler must have the type definitions before they are valid for use.

Listing 13.2 shows the contents of the gxGraphics.h file.

Listing 13.2 The gxGraphics.h File

```
/**
1:
2:
        ** 2D Graphical Editor (2d-gx)
3:
        * *
4:
        ** gxGraphics.h
5:
        */
       #include <X11/Intrinsic.h>
6:
7:
8:
       #ifndef _GX_GRAPHICS_H_INC_
9:
       #define _GX_GRAPHICS_H_INC_
10:
11:
       #endif /* GX GRAPHICS H INC */
12:
       #ifndef GLOBAL
13:
14:
       #define GLOBAL
15:
       #else
       #undef GLOBAL
16:
       #define GLOBAL extern
17.
18:
       #endif
19:
20:
       GLOBAL Widget GxStatusBar;
21:
       GLOBAL Widget GxDrawArea;
22:
23:
       /**
        ** end of gxGraphics.h
24:
25:
        */
```

In its current state, gxGraphics.h is a very simple file. As the Graphics Editor project grows in complexity, so will the gxGraphics.h header file.

After ensuring that the Intrinsic.h header file has been included to resolve the Widget data type used later in the file, gxGraphics.h checks for the existence of an environment directive _GX_GRAPHICS_H_INC_. If the variable has not been defined, gxGraphics.h defines it to ensure that a portion of the file is included only once:

```
8: #ifndef _GX_GRAPHICS_H_INC_
9: #define _GX_GRAPHICS_H_INC_
```

This mechanism for preventing multiple inclusions of the same file is common in C syntax when new data types or variables are created because multiple inclusions would redefine the types or variables defined the first time the file was included.

Although gxGraphics.h is in its fledgling state, the syntax is included for future expansion. Line 11 of Listing 13.2 closes the implicit body of the ifndef directive:

```
11: #endif /* _GX_GRAPHICS_H_INC_ */
```

A similar syntax is seen with the environment variable GLOBAL:

13: #ifndef GLOBAL
14: #define GLOBAL

Not meant to prevent multiple inclusions, the test of the presence of GLOBAL indicates whether this is the first inclusion of the file. If so, it *declares* the variables prefaced with the GLOBAL variable as in the first iteration. GLOBAL is defined to be *nothing*:

20: GLOBAL Widget GxStatusBar;
 21: GLOBAL Widget GxDrawArea;

Subsequent inclusions of the header file and the test for the presence of GLOBAL result in GLOBAL being redefined to the keyword extern.

This is a powerful mechanism that enables a variable to first be declared and subsequently externed. Effectively, the first file to include this header is responsible for declaring all the global variables used by the project. Further, inclusions simply inform the compiler that the variable has already been declared for use.

Listing 13.3 shows the contents of the gxProtos.h file in its current state. This file also grows in complexity as more and more functions are shared throughout the project.

Listing 13.3 The gxProtos.h File

```
1: /**

2: ** 2D Graphical Editor (2d-gx)

3: **

4: ** gxProtos.h

5: */
```

Listing 13.3 continued

```
6:
       #include <X11/Intrinsic.h>
7:
       #include "gxIcons.h"
8:
9:
       #ifndef EXTERN
10:
       #define EXTERN
11:
       #else
       #undef EXTERN
12:
13:
       #define EXTERN extern
14:
       #endif
15:
       /*
16:
       * Creations routines in gxGraphics.c
17:
18:
       */
19:
       EXTERN Widget create canvas ( Widget );
                   create_status ( Widget, Widget );
20:
       EXTERN void
21:
       EXTERN void create_buttons( Widget );
22:
       EXTERN void drawAreaEventProc(Widget, XtPointer, XEvent *, Boolean);
23:
24:
       /*
25:
       * Drawing functions used in GxIcons.h
       */
26:
27:
       EXTERN void gx_line( void );
28:
       EXTERN void gx pencil( void );
29:
       EXTERN void gx_arc( void );
30:
       EXTERN void gx box( void );
31:
       EXTERN void gx arrow( void );
32:
       EXTERN void gx text( void );
33:
34:
       /*
35:
       * Control functions used in GxIcons.h
36:
       */
37:
       EXTERN void gx_copy( void );
38:
       EXTERN void gx_delete( void );
39:
       EXTERN void gx_select( void );
40:
       EXTERN void gx_save( void );
41:
       EXTERN void gx_load( void );
42:
       EXTERN void gx_export( void );
43:
       EXTERN void gx_exit( Widget, XtPointer, XtPointer );
44:
45:
       /*
       * Utilities in gxGx.c
46:
47:
       */
48:
       EXTERN void setStatus( char * );
49:
       EXTERN void draw manager( Widget, XtPointer, XtPointer );
       EXTERN void ctrl_manager( Widget, XtPointer, XtPointer );
50:
51:
52:
       /**
       ** end of gxProtos.h
53:
54:
       */
```

Notice that the use of EXTERN follows the same form and serves the same purpose as the environment directive GLOBAL used in gxGraphics.h. However, a different name is used here to avoid conflicts in the compiler environment.

Many functions here have not been introduced yet. Focus for the moment on only the lines

```
19: EXTERN Widget create_canvas ( Widget );
20: EXTERN void create_status ( Widget, Widget );
21: EXTERN void create_buttons( Widget );
```

which are pertinent to our current analysis of Listing 13.1.

The gxIcons.h file and remaining prototypes will be introduced shortly as you begin to create the buttons and fill them with their respective icons.

The prototypes that are found on lines 19–20 satisfy function calls made from the gxMain.c file and justify the inclusion of the gxProtos.h header file.

Continuing the discussion of gxMain.c with lines 26-28

26:	toplevel	=	XtVaAppInitialize(&appCont	text,	"2D	Grap	ohical	Editor",
27:				NULL,0,	&argc	, ar	·gν,	NULL,	
28:				NULL);					

a call to XtVaAppInitialize is made to establish a connection to the X server, initialize the X Resource Database for the application, and fill the application context.

→ Review Chapter 6, "Components of an X Window Application," see the section "Connecting to the X Server," page 144, for a description of the components in an X Window application and use of the XtVaAppInitialize function.

To the toplevel widget returned from the initialization of the X environment, function main in gxMain.c adds a form widget as its sole child:

```
30: form = XtVaCreateWidget( "topForm",
31: formWidgetClass, toplevel,
32: NULL );
```

To the form widget are added a canvas area, a status bar, and the object management and control buttons:

```
34: canvas = create_canvas( form );
35: create_status( form, canvas );
36: create_buttons( form );
```

Prototypes for these functions are found in gxProtos.h; however, the actual function definitions are in the file gxGraphics.c as seen in Listing 13.4.

```
Listing 13.4 The gxGraphics.c File
```

```
/**
1:
2:
       ** 2D Graphical Editor (2d-gx)
       * *
3:
       ** gxGraphics.c
4:
5:
       */
6:
      #include <stdio.h>
7:
8:
      #include <X11/Intrinsic.h>
9:
      #include <X11/StringDefs.h>
10:
      #include <X11/Xaw/Form.h>
11:
      #include <X11/Xaw/Command.h>
12:
      #include <X11/Xaw/Box.h>
13:
14:
      #include "gxGraphics.h"
15:
      #include "gxIcons.h"
16:
17:
      /*
18:
       * Create the region of the application where we will draw
19:
       */
20:
      Widget create_canvas( Widget parent )
21:
      {
        GxDrawArea = XtVaCreateWidget( "drawingArea",
22:
23:
                                         formWidgetClass, parent,
24:
                                         XtNbackground,
25:
                                           WhitePixelOfScreen(XtScreen(parent)),
26:
                                                           XawChainTop,
                                         XtNtop,
27:
                                                            XawChainLeft,
                                         XtNleft,
28:
                                                           XawChainBottom,
                                         XtNbottom,
29:
                                         XtNright,
                                                           XawChainRight,
30:
                                         XtNheight,
                                                            220,
31:
                                         XtNwidth,
                                                            250,
32:
                                         NULL );
33:
34:
        XtAddEventHandler(GxDrawArea, PointerMotionMask, False,
35:
                           (XtEventHandler)drawAreaEventProc, (XtPointer)NULL);
36:
        XtAddEventHandler(GxDrawArea, ButtonPressMask|ButtonReleaseMask,False,
37:
                           (XtEventHandler)drawAreaEventProc, (XtPointer)NULL);
38:
39:
        return GxDrawArea;
40:
      }
41:
42:
      /*
43:
       * create status
44:
       */
45:
      void create status( Widget parent, Widget fvert )
46:
      {
47:
        GxStatusBar = XtVaCreateManagedWidget( "statusBar",
48:
                                                 labelWidgetClass, parent,
49:
                                                 XtNtop,
                                                                  XawChainBottom,
50:
                                                 XtNleft,
                                                                  XawChainLeft,
```

```
51:
                                                 XtNbottom,
                                                                  XawChainBottom,
52:
                                                                  XawChainRight,
                                                 XtNright,
53:
                                                 XtNfromVert,
                                                                  fvert,
54:
                                                 XtNborderWidth,0,
55:
                                                 NULL );
56:
        setStatus( "2D-GX (c)Starry Knight Software - Ready..." );
57:
      }
58:
59:
      /*
60:
       * statusProc
       */
61:
62:
      void statusProc( Widget w, XtPointer msg, XEvent *xe, Boolean flag )
63:
      {
64:
        if( msg != NULL )
65:
          setStatus( msg );
66:
        else
67:
          setStatus( "\0" );
68:
      }
69:
70:
      /*
71:
       *
          create_icons
       */
72:
73:
      void create_icons( Widget parent, GxIconData *iconData,
74:
                          void (*callback)( Widget, XtPointer, XtPointer ))
75:
      {
76:
        Widget btn;
77:
        Pixmap pix;
78:
79:
        while( iconData->info != NULL ) {
80:
          if( iconData->info->bits != NULL ) {
81:
            pix = create pixmap( parent, iconData->info );
82:
83:
            btn = XtVaCreateManagedWidget( "",
84:
                                             commandWidgetClass, parent,
85:
                                             XtNwidth, iconData->info->w + 1,
86:
                                             XtNheight, iconData->info->h + 1,
87:
                                             XtNbackgroundPixmap,
                                                                       pix,
88:
                                             XtNhighlightThickness,
                                                                       1,
89:
                                             NULL );
90:
91:
            XtAddEventHandler( btn, EnterWindowMask, False,
92:
                               (XtEventHandler) statusProc,
                               ➡(XtPointer)iconData->mesg);
93:
            XtAddEventHandler( btn, LeaveWindowMask, False,
94:
                               (XtEventHandler)statusProc, (XtPointer)NULL );
95:
96:
            XtAddCallback( btn, XtNcallback, callback,
              ➡(XtPointer)iconData->func );
97:
          }
98:
          /*
99:
          * go to the next element
           */
100:
```

Listing 13.4 continued

```
101:
          iconData++;
102:
        }
103:
      }
104:
105:
      /*
106:
       * Create a panel of buttons that will allow control of the application
107:
       */
108:
      void create_buttons( Widget parent )
109:
      {
110:
        Widget butnPanel, exitB;
111:
112:
        /*
113:
         * create a panel for the drawing icons
114:
         */
115:
        butnPanel = XtVaCreateWidget( "drawButnPanel",
116:
                                         boxWidgetClass, parent,
117:
                                         XtNtop,
                                                            XawChainTop,
118:
                                         XtNright,
                                                            XawChainRight,
                                         XtNbottom,
119:
                                                            XawChainTop,
120:
                                                            XawChainRight,
                                         XtNleft,
                                         XtNhorizDistance, 10,
121:
122:
                                         XtNfromHoriz,
                                                            GxDrawArea,
123:
                                         XtNhSpace,
                                                            1,
124:
                                         XtNvSpace,
                                                            1,
125:
                                         NULL );
126:
127:
        create_icons( butnPanel, gxDrawIcons, draw_manager );
128:
        XtManageChild( butnPanel );
129:
130:
        /*
131:
         * create a panel for the control icons
132:
         */
133:
        butnPanel = XtVaCreateWidget( "ctrlButnPanel",
134:
                                        boxWidgetClass, parent,
135:
                                        XtNtop,
                                                       XawChainTop,
136:
                                        XtNright,
                                                      XawChainRight,
137:
                                        XtNbottom,
                                                      XawChainTop,
138:
                                        XtNleft,
                                                       XawChainRight,
139:
                                        XtNfromHoriz, butnPanel,
140:
                                        XtNhorizDistance,0,
141:
                                        XtNhSpace,
                                                           1,
142:
                                        XtNvSpace,
                                                           1,
143:
                                        NULL );
144:
145:
        create icons( butnPanel, gxCntrlIcons, ctrl manager );
146:
        XtManageChild( butnPanel );
147:
148:
        exitB = XtVaCreateManagedWidget( "
                                                Exit
149:
                                           commandWidgetClass, parent,
150:
                                           XtNtop,
                                                          XawChainBottom,
```

13

```
151:
                                           XtNbottom,
                                                         XawChainBottom,
152:
                                           XtNleft,
                                                         XawChainRight,
153:
                                           XtNright,
                                                         XawChainRight,
154:
                                           XtNfromVert, butnPanel,
                                           XtNfromHoriz, GxStatusBar,
155:
156:
                                           NULL );
157:
158:
        XtAddCallback( exitB, XtNcallback, gx_exit, NULL );
      }
159:
160:
161:
      /*
       * create pixmap
162:
       */
163:
      Pixmap create pixmap( Widget w, XbmDataPtr data )
164:
165:
      {
166:
       return( XCreatePixmapFromBitmapData(XtDisplay(w),
167:
                                             DefaultRootWindow(XtDisplay(w)),
168:
                                             (char *)data->bits,
169:
                                             data->w, data->h,
170:
                                             BlackPixelOfScreen(XtScreen(w)),
171:
                                             WhitePixelOfScreen(XtScreen(w)),
172:
                                             DefaultDepthOfScreen(XtScreen(w)));
173: }
174:
175: /**
176: ** end of gxGraphics.c
177:
     */
```

Lines 8–12 of Listing 13.4 issue the include compiler directive for inclusion of the header files required to resolve variable, structure, and prototype definitions external to the gxGraphics.c file.

The first function defined in gxGraphics.c shown in Listing 13.4 is the create_canvas routine, and at lines 17-40.

The create_canvas function accepts one parameter for specifying the parent of the form widget it creates to act as the canvas area. Notice that the variable storing the value returned by the creation function is the global variable GxDrawArea, which is found in gxGraphics.h.

The placement of the canvas area created by create_canvas is specified as having its edges chained to the corresponding edges of the parent. Clearly, the parent is expected to be of the class formWidgetClass because relative positioning has no meaning with manager widgets of other classes.

→ In Chapter 5, the section "The Form Widget," page 136, demonstrates the use of composite resources for specifying relative placement of the children it manages.

276 Part IV Laying Out the Parts

Finally, the create_canvas function registers for the receipt of X events that would not normally be sent to a form widget.

By use of the X Toolkit Intrinsic call XtAddEventHandler, the create_canvas function can request that the function drawAreaEventProc is called when PointerMotion events are received. The prototype for drawAreaEventProc is found in gxProtos.h, and its definition will be introduced shortly.

Although the variable GxDrawArea is global to the project, its value is returned for use by the calling function main found in gxMain.c. Even though main could refer to the value by the global reference, the return is included to demonstrate its use.

Continuing in the review of gxGraphics.c, consider lines 42-57, which define the function create_status.

The create_status function creates a label widget for use as place to display context-sensitive help messages. Assignment of the help message is seen with the definition of the object drawing and management buttons. Optionally, messages can be explicitly displayed by invoking the setStatus function that will be introduced shortly.

Notice in the placement of the label widget (lines 49–54) that the resource XtNfromVert used to indicate the vertical placement of the widget is relative to the value of the second parameter passed to the function. Looking back at Listing 13.1 at the invocation of the create_status function reveals that the second parameter is the GxDrawArea widget.

Lines 59–68 of the file gxGraphics.c shown in Listing 13.4 define the function statusProc that is used to process the event handler assigned to buttons created by the function create_icons.

The function statusProc is responsible for updating the GxStatusBar label widget to reflect the context-sensitive help message for any button that gains focus.

The create_icons function is located at lines 70–97 in Listing 13.4 and is responsible for parsing the GxIconData reference passed as the second parameter from the function create_buttons function.

The definition of the GxIconData structure pointer dictates everything about the buttons being added to the interface.

Listing 13.5 introduces the gxIcon.h header file containing the GxIconData references, bitmap icons, callback functions, and context-sensitive help strings used to define the icons created for the application.

```
Listing 13.5 The gxIcons.h File
```

```
/**
1:
2:
                        ** 2D Graphical Editor (2d-gx)
                        * *
3:
                        ** gxIcons.h
4:
5:
                       */
6:
                     #ifndef _GX_ICONS_H_INC_
7:
                     #define GX ICONS H INC
8:
9:
                     #include "gxProtos.h"
10:
11:
                        /*
                           * Storage for pertinent XBM data
12:
                          */
13:
14:
                        typedef struct xbm data {
15:
                             unsigned char *bits;
16:
                              int
                                                                       w, h;
17:
                        } XbmData, *XbmDataPtr;
18:
19:
                        /*
20:
                           * IconData necessary to create icon
21:
                           */
22:
                        typedef struct _gx_icon_data {
23:
                              XbmDataPtr info;
24:
25:
                              void (*func)(void);
26:
                              char *mesg;
                        } GxIconData, *GxIconDataPtr;
27:
28:
29:
                       #define icon static( name, bits, width, height ) \
30:
                          static XbmData name = { bits, width, height }
31:
32:
                        /*
33:
                           * drawing icons
34:
                          */
35:
                        static unsigned char line_bits[] = {
36:
                                0 \times 00\,, 0
37:
                                 38:
                                39:
                                 40:
                                0x00,0x00,0x88,0x00,0x14,0x00,0x00,0x84,0x00,0x14,0x00,
                                41:
42:
                                0x02,0x02,0x22,0x00,0x00,0x01,0x02,0x42,0x00,0x00,0x01,
43:
                                0x04,0x42,0x00,0x80,0x00,0x04,0x42,0x00,0x80,0x00,0x02,
44:
                                0 \times 01, 0 \times 00, 0 \times 40, 0 \times 00, 0 \times 01, 0 \times 01, 0 \times 00, 0 \times 40, 0 \times 80, 0 \times 00, 0 \times 01,
45:
                                0x00,0x20,0x40,0x00,0x01,0x00,0x20,0x20,0x00,0x01,0x00,
46:
                                 47:
                                48:
                                 49:
                                50:
                                 0 \times 00, 0 \times 00 \};
51:
                        icon static( line icon, line bits, 36, 32 );
```

Listing 13.5 continued

52:	
53	<pre>static unsigned char pencil bits[] = {</pre>
54.	
55.	α_{X00}
56.	0×03 0×00 0×00 0×00 0×08 0×03 0×00 0×00 0×00 0×01 0×01 0×01
57.	$0 \times 00, 0 \times $
58.	0,000,0,000,0,000,0,000,0,000,0,000,0,0,
50.	0,000,0,000,0,000,0,000,00,000,0,000,0,0
59.	0.00, 0.00, 0.01, 0.00
61.	$0 \times 04, 0 \times 01, 0 \times 00, 0 \times 00, 0 \times 00, 0 \times 02, 0 \times 00, 0 \times $
61:	
62:	
63:	0x00,0x00,0x00,0x30,0x01,0x00,0x00,0x28,0x10,0x00,0x00,0x00,
64:	0x44,0x10,0x00,0x00,0x00,0x84,0x20,0x00,0x00,0x00,0x00,0x04,0x41,
65:	0x00,0x00,0x00,0x08,0x42,0x00,0x00,0x00,0x10,0x44,0x00,0x00,
66:	0x00,0x20,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x0
67:	0×00,0×00,0×00,0×00};
68:	icon_static(pen_icon, pencil_bits, 36, 32);
69:	
70:	static unsigned char arc_bits[] = {
71:	$0 \times 00, 0 \times $
72	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
73:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x3f,0x00,0x00,0x00,
74:	0x70,0xc0,0x01,0x00,0x00,0x0c,0x00,0x06,0x00,0x00,0x02,0x00,
75:	0x08,0x00,0x00,0x01,0x00,0x10,0x00,0x80,0x00,0x0
76:	0x40,0x00,0x00,0x40,0x00,0x40,0x00,0x04,0x40,0x00,0x20,0x00,
77:	0x04,0x80,0x00,0x20,0x00,0x1f,0x80,0x00,0x20,0x00,0x04,0x80,
78:	0x00,0x40,0x00,0x04,0x40,0x00,0x40,0x00,0x00,0x40,0x00,0x80,
79:	0x00,0x00,0x20,0x00,0x00,0x01,0x00,0x10,0x00,0x0
80:	0x08,0x00,0x00,0x0c,0x00,0x86,0x00,0x00,0x70,0xc0,0x81,0x00,
81:	0x00.0x80.0x3f.0xe0.0x03.0x00.0x00.0x00.0x80.0x00.0x00.0x0
82:	0x00,0x80,0x00,0x00,0x00,0x00,0x00,0x00
83:	0x00.0x00.0x00.0x00.0x00.0x00.0x00.0x00.0x00.0x00.0x00.0x00.0x00.
84:	0x00.0x00.0x00.0x00}:
85:	icon static(arc icon, arc bits, 36, 32):
86:	,
87.	<pre>static unsigned char how hits[] = {</pre>
88.	
89.	0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0
00.	$0 \times 00, 0 \times $
01·	0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x0
00.	0,000,0,000,0,000,0,000,0,000,0,000,0,000,0,
92.	
93. 04:	UNDU,UNUU,UNUU,UNIU,UNUU,UNOU,UNUU,UNUU,UNU
34. 05:	UNUU,UNUU,UNUU,UNUU,UNUU,UNUU,UNUU,UNU
90.	
96:	UX 1U, UXUU, UX8U, UXUU, UXUU, UX1U, UXUU, UX8U, UXUU, UXUU, UX1U,
9/:	0x00,0x80,0x00,0x00,0x10,0x00,0x80,0x00,0x0
98:	UX8U,UX0U,UX0U,UX1U,UX0U,UX8U,UXTT,UXTT,UX1T,UX0U,UX0U,
99:	uxuu,uxuu,uxuu,uxuu,uxuu,uxuu,0xu0,0x00,0x00,0x00,0x00,0x00,
100:	uxuu,ux00,0x00,0x00,0x00,0x00,0x00,0x00

```
101:
                                                                          0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
 102:
                                                                          0 \times 00, 0 \times 00 \};
 103:
                                                       icon static( box icon, box bits, 36, 32 );
 104:
105:
                                                      static unsigned char arrow_bits[] = {
 106:
                                                                          107:
                                                                         108:
                                                                          109:
 110:
                                                                          0 \times 00, 0 \times 00, 0 \times 00, 0 \times 04, 0 \times 40, 0 \times 00, 0 \times 00, 0 \times 00, 0 \times 02, 0 \times 80, 0 \times 00, 0 \times 
 111:
                                                                          112:
                                                                         0x00,0x04,0x00,0x20,0x00,0x00,0x08,0x00,0xe0,0x07,0xc0,0x0f,
 113:
                                                                          0 \times 00, 0 \times 00, 0 \times 04, 0 \times 40, 0 \times 00, 0 \times 00, 0 \times 00, 0 \times 04, 0 \times 40, 0 \times 00, 0 \times 
 114:
                                                                         0x04,0x40,0x00,0x00,0x00,0x04,0x40,0x00,0x00,0x00,0x00,0x04,0x40,
                                                                          115:
116:
                                                                         0x00,0x02,0x80,0x00,0x00,0xc0,0x01,0x00,0x07,0x00,0x00,0x00,
 117:
                                                                         118:
                                                                          0 \times 00, 0 \times 
119:
                                                                          0 \times 00, 0 \times 00, 0 \times 00, 0 \times 00;
 120:
                                                       icon_static( arr_icon, arrow_bits, 36, 32 );
 121:
 123:
                                                       static unsigned char text bits[] = {
124:
                                                                          125:
                                                                          0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
 126:
                                                                          127:
                                                                         0x1f,0x00,0x80,0x83,0x1f,0x1c,0x00,0x80,0x01,0x0f,0x18,
 128:
                                                                          129:
                                                                          0x00,0x00,0x0f,0x00,0x00,0x00,0x00,0x0f,0x00,0x00,0x00,
 130:
                                                                          0x00,0x0f,0x00,0x00,0x00,0x00,0x0f,0x00,0x00,0x00,0x00,
                                                                          131:
 132:
                                                                          133:
                                                                         0x00,0x00,0x0f,0x00,0x00,0x00,0x00,0x0f,0x00,0x00,0x00,
 134;
 135:
                                                                          0x80,0x1f,0x00,0x00,0x00,0xe0,0x7f,0x00,0x00,0x00,0x00,
                                                                          136:
 137:
                                                                          138:
                                                                          0 \times 00, 0 \times 00 \};
 139:
                                                       icon_static( text_icon, text_bits, 36, 32 );
 140:
 141:
                                                       /*
                                                           * control icons
 142:
                                                            */
 143:
 144:
                                                       static unsigned char
                                                                                                                                                                                                                     copy bits[] = {
 145:
                                                                          0 \times 00, 0 \times 
 146:
                                                                          0 \times 00, 0 \times 
 147:
                                                                          148:
                                                                         149:
                                                                          0x00,0x00,0x80,0x40,0x10,0x00,0x00,0x80,0x20,0x20,0x00,0x00,
 150:
                                                                         151:
                                                                          152:
 153:
```

Listing 13.5 continued

154:	0x00,0x00,0x60,0x73,0xdf,0x1d,0x00,0x60,0xd8,0xb6,0x0d,0x00,
155:	0x60,0xd8,0xb6,0x0d,0x00,0x60,0xdb,0x36,0x05,0x00,0xc0,0x71,
156:	0x1e,0x07,0x00,0x00,0x00,0x06,0x03,0x00,0x00,0x00
157:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
158:	0x00,0x00,0x00,0x00};
159:	icon_static(copy_icon, copy_bits, 36, 32);
160:	
161:	<pre>static unsigned char delete_bits[] = {</pre>
162:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
163:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
164:	0x00,0x00,0x05,0x80,0x02,0x00,0x00,0x09,0x40,0x02,0x00,0x00,
165;	0x11,0x20,0x01,0x00,0x00,0x22,0x90,0x00,0x00,0x00,0x44,0x48,
166:	0x00,0x00,0x00,0x88,0x25,0x00,0x00,0x00,0x10,0x1f,0x00,0x00,
167:	0x00,0xe0,0x0f,0x00,0x00,0x00,0xc0,0x0e,0x00,0x00,0x00
168:	0xfe,0x01,0x00,0x00,0xfe,0x9f,0x01,0x00,0x00,0x73,0x1e,0x03,
169:	0x00,0x00,0x33,0x18,0x03,0x00,0x00,0x1f,0x78,0x03,0x00,0x00,
170:	0x0e,0xf0,0x01,0x00,0x00,0x00,0x00,0x00,0x00
171:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
172:	0x00,0x3c,0xc0,0x00,0x00,0x00,0xf6,0xed,0x03,0x00,0x00,0x86,
173:	0xcd,0x00,0x00,0x00,0x86,0xcd,0x00,0x00,0x00,0xb6,0xcd,0x06,
174:	0x00,0x00,0x1c,0x9f,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x0
175:	0x00,0x00,0x00,0x00};
176:	icon_static(delete_icon, delete_bits, 36, 32);
177:	
178:	qstatic unsigned char select_bits[] = {
179:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
180:	0xf0,0xff,0x07,0x00,0x00,0x10,0x00,0x04,0x00,0x00,0x10,0xf0,
181:	0x05,0x00,0x00,0x10,0x40,0x04,0x00,0x00,0x10,0x40,0x04,0x6c,
182:	0xdb,0x10,0x40,0x04,0x00,0x00,0x10,0x40,0x04,0x04
183:	0x47,0x04,0x04,0x10,0x51,0x08,0x04,0x00,0x10,0x50,0x08,0x04,
184:	0x04,0x10,0x91,0x07,0x04,0x04,0x10,0x11,0x00,0x04,0xe0,0x11,
185:	0xf0,0xff,0x07,0x14,0x02,0x01,0x00,0x00,0x14,0x02,0x01,0x00,
186:	0x00,0xe0,0x01,0x00,0x00,0x00,0x04,0x00,0x01,0x00,0x00
187:	0x6d,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x0
188:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
189:	0xe0,0x81,0x01,0x20,0x00,0x30,0x81,0x01,0x30,0x00,0x70,0xb8,
190:	0x39,0x77,0x00,0xe0,0xac,0xad,0x35,0x00,0xc0,0xbd,0xbd,0x31,
191:	0x00,0x90,0x8d,0x8d,0x35,0x00,0xf0,0xb8,0x39,0x67,0x00,0x00,
192:	0x00,0x00,0x00,0x00};
193:	icon_static(select_icon, select_bits, 36, 32);
194:	
195:	
196:	<pre>static unsigned char save_bits[] = {</pre>
197:	0x80,0x07,0x00,0x00,0x00,0xc0,0xe4,0x6e,0x0e,0x00,0xc0,0xb3,
198:	0x6d,0x1b,0x00,0x00,0xe7,0x39,0x1f,0x00,0x40,0xb6,0x39,0x03,
199:	0x00,0xc0,0xf3,0x13,0x1e,0x00,0x00,0x00,0x00,0x00,0x00,0x00
200:	0x00,0x00,0x00,0x00,0x00,0x00,0xfc,0x1f,0x00,0x00,0x00,0x06,
201:	0x32,0x00,0x00,0x00,0x05,0x55,0x00,0x00,0x0
202:	0x00,0x06,0xf9,0xcf,0x00,0x00,0x0e,0x01,0xc0,0x00,0xe0,0x1f,
203:	0x01,0xc0,0x00,0xe0,0x3f,0x01,0xc0,0x00,0xe0,0x1f,0x01,0xc0,

204: 0x00,0x00,0x0e,0xf1,0xc7,0x00,0x00,0x06,0x15,0xd4,0x00,0x00, 205: 0x00,0xd5,0xd5,0x00,0x00,0x00,0x11,0xc4,0x00,0x00,0x00,0xfe, 206: 0xff,0x00,0x00,0x00,0xfc,0x7f,0x00,0x00,0x00,0x00,0x00,0x00, 207: 0x00,0x38,0x0c,0x07,0x00,0xe0,0x31,0x00,0x06,0x00,0x30,0x31, 208: 0x8e,0xe7,0x00,0xf0,0x30,0xcc,0xb6,0x01,0xc0,0x31,0xcc,0xf6, 209: 0x01,0x90,0x31,0xcc,0x36,0x00,0xf0,0xfc,0xbf,0xef,0x01,0x00, 210: $0 \times 00, 0 \times 00, 0 \times 00, 0 \times 00$; 211: icon static(save icon, save bits, 36, 32); 212: 213: static unsigned char load bits[] = { 214: 0x00,0x00,0x00,0x38,0x00,0xc0,0x03,0x00,0x30,0x00,0x80,0xc1, 215: 0x71,0x3c,0x00,0x80,0x61,0xdb,0x36,0x00,0x80,0x61,0xf3,0x36, 216: 0x00,0x80,0x6d,0xdb,0x36,0x00,0xc0,0xcf,0xf9,0x7d,0x00,0x00, 217: 0x00,0x00,0x00,0x00,0xc0,0xff,0x01,0x00,0x00,0x60,0x20,0x03, 218: 219: 0x90,0xff,0x0c,0x03,0x00,0x10,0x00,0x0c,0x07,0x00,0x10,0x00, 220: 0xcc,0x0f,0x00,0x10,0x00,0xcc,0x1f,0x00,0x10,0x00,0xcc,0x0f, 221: 0x00,0x10,0x7f,0x0c,0x07,0x00,0x50,0x41,0x0d,0x03,0x00,0x50, 222: 0x5d,0x0d,0x00,0x00,0x10,0x41,0x0c,0x00,0x00,0xe0,0xff,0x0f, 223: 224: 0x00,0x38,0x0c,0x07,0x00,0xe0,0x31,0x00,0x06,0x00,0x30,0x31, 225: 0x8e,0xe7,0x00,0xf0,0x30,0xcc,0xb6,0x01,0xc0,0x31,0xcc,0xf6, 226: 0x01,0x90,0x31,0xcc,0x36,0x00,0xf0,0xfc,0xbf,0xef,0x01,0x00, 227: $0 \times 00, 0 \times 00, 0 \times 00, 0 \times 00$; 228: icon static(load icon, load bits, 36, 32); 229: 230: static unsigned char export bits[] = { 231: 232: 0x00,0x30,0x00,0xd8,0x76,0xb7,0xf9,0x00,0xd8,0x5c,0x66,0x33, 233: 0x00,0x78,0x0c,0x66,0x33,0x00,0x18,0x0c,0x66,0xb3,0x01,0x3c, 234: 235: 236: 237: 0x10,0x00,0x00,0xfc,0x17,0xd0,0xff,0x00,0x00,0x10,0x10,0x00, 238: 0x00,0x00,0x10,0x10,0x00,0x00,0x00,0x10,0x10,0x00,0x00,0x00, 239: 0x10,0x10,0x00,0x00,0x00,0x10,0x10,0x00,0x00,0x00,0xe0,0x0f, 240: 0x00,0x00,0x00,0x00,0x00,0x80,0x01,0x1f,0x00,0x00,0x80,0x01, 241: 0xd3,0xfd,0x38,0xfb,0x07,0x8f,0xb7,0x6d,0xae,0x01,0x03,0xb3, 242: 0x6d,0x86,0x01,0x9b,0xb7,0x6d,0x86,0x0d,0xdf,0xf6,0x38,0x0f, 243: 244: $0 \times 00, 0 \times 00, 0 \times 00, 0 \times 00$; 245: icon static(export icon, export bits, 36, 32); 246: 247: static GxIconData gxDrawIcons[] = { 248: { &line icon, gx line, "Draw an elastic line..." }, 249: { &pen_icon, gx_pencil, "Draw a freestyle line..." }, 250: { &arc_icon, "Draw a circle..." gx_arc, }, "Draw a square or rectangle... 251: { &box icon, gx box, }, 252: gx_arrow, "Draw an arrow..." { &arr_icon, }, 253: "Draw dynamic text..." { &text_icon, gx_text, }, 254: /*----*/ 255: /* this list MUST be NULL terminated */

Listing 13.5 continued

```
256:
       /*----
                 */
       { NULL },
257:
258:
       };
259:
260:
       static GxIconData gxCntrlIcons[] = {
       { &copy icon, gx copy, "Copy selected object..."
262:
                                                                   },
       { &delete icon, gx delete, "Delete selected object..."
263:
                                                                   },
      { &select_icon, gx_select, "Select multiple objects..."
264:
                                                                   },
      { &save_icon, gx_save, "Save current drawing..."
265:
                                                                   },
       { &load_icon, gx_load, "Load saved drawing..."
266:
                                                                   },
       { &export_icon, gx_export, "Save drawing as GIF or PostScript..."
                                                                   },
267:
       /*----*/
268:
269:
       /* this list MUST be NULL terminated */
270:
       /*----*/
271:
       { NULL },
272:
       };
273:
274:
       /* prototypes */
275:
       extern void create icons (Widget, GxIconDataPtr,
276:
                                  void (*)(Widget, XtPointer, XtPointer ));
       extern Pixmap create_pixmap ( Widget, XbmDataPtr
277:
                                                      );
278:
       #endif /* GX ICONS H INC */
279:
280:
       /**
281:
        ** end of gxIcons.h
        */
282:
```

Following the ifndef preprocessor directive to prevent multiple inclusions of this header file and the include directive for including gxProtos.h for the forward declarations of the draw and management functions, the gxIcons.h file shown in Listing 13.5 defines two important structures for managing icon creation.

Lines 11–17 show the definition for the XbmData structure used to store the bit array data, width, and height values for the bitmap created with the X Client bitmap.

Lines 19–27 in Listing 13.5 hold the definition for the GxIconData structure. The GxIconData structure nests a reference to an XbmData structure and includes fields for the callback function to be assigned to the icon and the context-sensitive help message, which is displayed when the mouse cursor enters the icon.

Following the structure definitions, the gxIcons.h header file shown Listing 13.5 defines a macro for declaring and initializing the XbmData references from the bitmap data found in lines 32–245:

29: #define icon_static(name, bits, width, height) \
30: static XbmData name = { bits, width, height }

The bitmap data is divided into two groups. Lines 32–139 define the object drawing icons and lines 141–245 define the object management icons. The bitmap data used

to define the different icons, as mentioned, is created using the X Client bitmap (or equivalent). After the desired icon is created using a utility capable of saving data in XBM format, the contents of the saved file is inserted into the gxIcons.h header for visibility in the project.

→ Chapter 6, in the section "Creating Buttons," on page 159, discusses in greater detail the generation of bitmaps for use by an application.

After the many icons defined for use in the Graphics Editor user interface, lines 147–258 show the GxIconData array definition for gxDrawIcons. This array defines the icons, callbacks, and context-sensitive help messages for the object-drawing buttons. Lines 260–272 show a similar definition for the gxCntrlIcons used to define the object management buttons.

These arrays are passed independently by create_buttons in Listing 13.4 to the create_icons function for parsing and conversion into command widgets that will act as holders for the different icons:

```
128: create_icons( butnPanel, gxDrawIcons, draw_manager );
```

and

```
146: create_icons( butnPanel, gxCntrlIcons, ctrl_manager );
```

The create_icons function defines a loop and is executed as long as there is a valid element of the GxIconData pointer iconData.

Notice in Listing 13.5 that the GxIconData arrays are NULL terminated. The create_icons function loops until it finds the end of the array as indicated by the presence of the NULL:

```
80: while( iconData->info != NULL ) {
```

As long as there is a valid array element, a Pixmap is created from the XbmData field of the GxIconData structure called info:

```
82: pix = create_pixmap( parent, iconData->info );
```

This Pixmap is then used as the background Pixmap of the command widget whose parent is the button panel created by create_buttons and passed as the first parameter of the create_icons function.

Event handlers are added to this button to manage the context-sensitive help string—displayed when the button gains focus (mouse cursor enters button's window) and to clear the string when the focus is lost.

Added to the button, as well, is a callback function dictated by the type of button that is being created. The buttons from the gxDrawIcons array are assigned the draw_manager callback function and the gxCntrlIcons are given the cntrl_manager function.

The draw_manager and cntrl_manager functions are prototyped in gxProtos.h and passed as the third parameter to the create_icons function.

The call data specified to the XtAddCallback function is very important:

96: XtAddCallback(btn, XtNcallback, callback, →(XtPointer)iconData->func);

The func field of the GxIconData corresponds to the specific action that will be invoked within the draw_manager or cntrl_manager functions.

Look again at the declaration of the gxDrawIcons and gxCntrlIcons arrays gxIcons.h shown in Listing 13.4. The second field of the elements being initialized determines the specific action assigned to the icon.

Functions such as gx_line, gx_pencil, gx_arc, and so forth are assigned to the gxDrawIcons array elements. The gxCntrlIcons elements have functions such as gx_copy, gx_delete, and gx_select assigned to them.

All the functions that are found in the GxIconData array elements declared in gxIcons.h are prototyped in gxProtos.h. Their declarations will be introduced shortly as will the draw_manager and cntrl_manager functions.

First, however, we end the analysis of Listing 13.4 by noting the creation of the exit button on lines 148–156:

148: exitB = XtVaCreateManagedWidget(" Exit "

This is the first button that has been created to use a label type of string. All the previous command widgets had a background pixmap assigned, which excludes the use of text in the button.

The label assigned to a command button is its instance name by default. Notice that the instance name has spaces nested in it to extend the size of the button to approximately equal the menu button panes created in the create_buttons function.

Although valid syntax, a danger exists in using spaces in widget instance names: the difficulty in including the value in an external resource file such as .Xdefaults.

Returning to Listing 13.1, a few final requirements must be met in the function main defined in gxMain.c.

Specifically, the form and canvas widget created unmanaged are explicitly managed:

38: XtManageChild(canvas); 39: XtManageChild(form);

All X applications require the realization of the toplevel widget: recursively mapping the windows of all its descendents to the screen with a call to XtRealizeWidget: 41: XtRealizeWidget(toplevel);

The last step of the function main is to enter an infinite loop in which the application will continually monitor its event queue:

43: XtAppMainLoop(appContext);

As the X Server sends events to the application, the Graphics Editor will remove them from the queue and dispatch them to the widgets that have registered a request to receive them.





Because the XtAppMainLoop function contains an infinite loop, this line of the function will *never* be reached.

Having completed structuring the application, there are only a few more housekeeping issues to discuss.

Listing 13.6 introduces the functions defined in the gxCntrlIcons array elements declared in gxIcons.h.

Listing 13.6 The gxGx.c File

```
/**
1:
       ** 2D Graphical Editor (2d-gx)
2:
       **
3:
4:
       ** gxGx.c
5:
       */
6:
      #include <stdio.h>
7:
8:
      #include "gxGraphics.h"
      #include "gxProtos.h"
9:
10:
       #include <X11/Xaw/Label.h>
11:
12:
13:
       static void (*draw_mgr_func)( XEvent * ) = NULL;
14:
       /*
15:
16:
        * gx_exit
        */
17:
```

Listing 13.6 continued

```
18:
       void gx_exit( Widget w, XtPointer cd, XtPointer cbs )
19:
       {
20:
         exit(0);
21:
       }
22:
23:
       /*
24:
        * gx_copy
       */
25:
26:
       void gx_copy( void )
27:
       {
28:
         printf( "gx_copy\n" );
29:
       }
30:
31:
       /*
32:
        * gx_delete
33:
       */
34:
       void gx_delete( void )
35:
       {
         printf( "gx_delete\n" );
36:
37:
       }
38:
39:
       /*
40:
       * gx_select
41:
       */
42:
       void gx_select( void )
43:
       {
44:
       printf( "gx_select\n" );
45:
       }
46:
47:
       /*
48:
       * gx_save
       */
49:
50:
       void gx_save( void )
51:
       {
52:
       printf( "gx_save\n" );
53:
       }
54:
55:
       /*
       * gx_load
56:
        */
57:
58:
       void gx_load( void )
59:
       {
60:
         printf( "gx_load\n" );
61:
       }
62:
63:
       /*
       * gx_export
64:
       */
65:
66:
       void gx_export( void )
67:
       {
```

```
68:
         printf( "gx export\n" );
69:
       }
70:
71:
       /*
72:
        * setStatus
73:
        */
74:
       void setStatus( char *message )
75:
       {
76:
         XtVaSetValues( GxStatusBar, XtNlabel, message, NULL );
77:
       }
78:
79:
       /*
        * draw_manager
80:
        */
81:
82:
       void draw manager( Widget w, XtPointer cd, XtPointer cbs )
83:
       {
84:
         void (*draw_func)( XEvent * ) = (void (*)(XEvent *))cd;
85:
86:
         if( draw_func != NULL ) (*draw_func)( NULL );
87:
         draw_mgr_func = draw_func;
88:
       }
89:
       /*
90:
91:
        * ctrl manager
92:
        */
93:
       void ctrl_manager( Widget w, XtPointer cd, XtPointer cbs )
94:
       {
95:
         void (*ctrl func)( void ) = (void(*)(void))cd;
96:
         if( ctrl_func != NULL ) ctrl_func();
97:
       }
98:
99:
       /*
100:
        * drawAreaEventProc
101:
        */
102:
        void drawAreaEventProc( Widget w, XtPointer cd, XEvent *event,
        ⇒Boolean flag )XEvent *event, Boolean flag )
103:
        {
104:
          if( draw_mgr_func != NULL ) (*draw_mgr_func)( event );
105:
        }
106:
        /**
107:
         ** end of gxGx.c
108:
         */
109:
```

Most of the functions defined in the file gxGx.c shown in Listing 13.6 are only stubs at this point, doing little more than printing out a statement that the function has been reached.

The bodies of these functions will evolve over time; however, their presence even in this simple form is absolutely necessary to satisfy the link phase when building this phase of the project.

The functions used in the gxDrawIcon array elements defined gxIcons.h are not contained in a single file. Instead, a file specific to the graphic object type created by the icon is defined in Listings 13.7 through 13.9.

Listing 13.7 The gxLine.c File

```
/**
1:
       ** 2D Graphical Editor (2d-gx)
2:
       * *
3:
       ** gxLine.c
4:
5:
       */
6:
      #include <stdio.h>
7:
      #include "gxGraphics.h"
8:
9:
      void gx_line( XEvent *event )
10:
       {
11:
         printf( "draw a line...\n" );
12:
       }
13:
14:
       void gx_pencil( XEvent *event )
15:
       {
         printf( "draw freestlye\n" );
16:
17:
       }
18:
       void gx_arrow( XEvent *event )
19:
20:
       {
21:
         printf( "draw an arrow\n" );
22:
       }
23:
24:
       void gx_box( XEvent *event )
25:
       {
26:
         printf( "draw a box\n" );
27:
       }
28:
       /**
29:
        ** end of gxLine.c
30:
31:
        */
```

The file gxLine.c shown in Listing 13.7 controls the creation of all point-based objects.

Listing 13.8 The gxArc.c File

```
1:
      /**
2:
       ** 2D Graphical Editor (2d-gx)
3:
       **
       ** gxArc.c
4:
5:
       */
6:
      #include <stdio.h>
7:
8:
      void gx_arc( void )
9:
      {
```

```
10: printf( "draw an arc...\n" );
11: }
12:
13: /**
14: ** end of gxArc.c
15: */
```

The gxArc.c file shown in Listing 13.8 contains only support for the arc graphic object.

```
Listing 13.9 The gxText.c File
```

```
/**
1:
2:
       ** 2D Graphical Editor (2d-gx)
       **
3:
       ** gxText.c
4:
5:
       */
6:
      #include <stdio.h>
7:
8:
      void gx_text( void )
9:
      {
10:
         printf( "draw text...\n" );
11:
       }
12:
13:
       /**
        ** end of gxText.c
14:
        */
15:
```

The gxText.c file shown in Listing 13.9 contains only source code for supporting the vector text or *Hershey font* graphic object.

The functions shown to satisfy the references assigned to the drawing icons in the gxDrawIcons array are, like most of the functions in gxGx.c shown in Listing 13.6, only stubs at this point in the project. These files will grow significantly in complexity during the next several chapters as the actual objects are defined, controlled, saved, and restored.

The next section is a diversion from advancing the project in that there are currently no command-line parameters supported by the application. However, I would be negligent to not demonstrate ways to add flags and parameters to the Graphics Editor program.

Parsing the Command Line

Providing information to an application is often crucial for proper use of the program. Because the application is responsible for extracting any information specified by the user when the program was invoked, this section demonstrates two methods of parsing the command line.

The C programming language provides a function called getopt. Similar to getopt, but significantly more robust, is the function supplied by the X Toolkit Intrinsics called XtVaGetApplicationResources.

The getopt Function

The getopt function is a built-in function provided by the C language and is prototyped in the header file stdlib.h included with most C source code files.

The getopt function is dependent upon several external variables and is destructive to the command line; that is, the command line can be parsed only one time using the getopt function.

A prototype for the getopt function follows:

int getopt(int argc, char *argv, const char *optstring);

The getopt function returns the next option letter found in argv that matches a letter in the optstring value.

The definition of optstring made by the programmer is done to specify the option letters used by the application and therefore enable the getopt function to recognize them.



The letters placed in the definition of the optstring value are synonymous with what has been referred to as the flags passed to an application.

In forming the value for optstring, if a colon follows a letter, the option is expected to have an associated argument.

The external variables used by getopt must be declared to the file actually employing the getopt function. These variables include the following:

```
extern char *optarg;
extern int optind, opterr, optopt;
```

The getopt function places the index of the next argument to be processed in the optind variable. The optind is external and initialized to 1 before the first call to getopt.

When all options have been processed, meaning nothing remains on the command line, getopt returns the constant EOF.

If flags (letters) are specified on the command line that are not contained in the optstring value, getopt prints an error message to the standard error (stderr) and

returns a question mark. However, if the external variable opterr is set to 0 before invoking the getopt function, error messages are disabled and getopt will not report unidentified options found on the command line.

Listing 13.10 shows a code fragment for how the getopt function can be used to process command-line arguments.

Listing 13.10 The getopt Function

```
1:
      #include <stdlib.h>
2:
      #include <stdio.h>
3:
      main (int argc, char **argv)
4:
      {
5:
         int c;
6:
7:
         extern char *optarg;
8:
         extern int optind;
9:
10:
         int errCnt
                        = 0:
10:
         int errCnt
                       = 0;
11:
         int verbose = 0:
12:
         char *infile = NULL;
         char *outfile = NULL;
13:
14:
15:
         while ((c = getopt(argc, argv, "vi:o:")) != EOF) {
16:
           switch (c) {
             case 'v':
17:
18:
               verbose = 1;
19:
               break;
             case 'i':
20:
               infile = optarg;
21:
22:
               printf("infile = %s\n", infile);
23:
               break;
24:
             case 'o':
25:
               outfile = optarg;
26:
               printf("outfile = %s\n", outfile);
27:
               break:
             case '?':
28:
29:
                errCnt++;
30:
           }
31:
         }
         if( errCnt > 0 ) {
32:
33:
           fprintf( stderr,
                    "usage: %s -v, -i <input filename> -o <output filename>\n",
34:
35:
                    argv[0] );
36:
           exit( 2 );
37:
         }
38:
         exit( 0 );
39:
       }
```

The flags identified by the sample code shown in Listing 13.10 are v, i, and o. Because the options i and o are followed by a colon (:) in the optstring value, it is expected that these flags will have an associated argument, which the getopt functions assigns to optarg:

20: case 'i': 21: infile = optarg;



A caveat in the use of the getopt function is that getopt does not enforce the inclusion of arguments for flags that require them.

For example, if the code fragment above was invoked without the user specifying a value for the -i flag, getopt would use the next item placed on the command line, which potentially would be the -o flag.

A more resilient approach to parsing the command line is afforded Intrinsics-based applications as demonstrated in the next section.

The XtVaGetApplicationResources Function

The XtVaGetApplicationResources function enables a much more convenient and safer method of parsing the command line. Note, however, that the function is responsible for much more than just the command line.

In fact, the options specified can also appear in a resource file for the application in the same way that widget attribute resources are defined.

The XtVaGetApplicationResources function has the following prototype:

The first parameter specifies a widget that is used to identify the resource database to search for the values of the resources specified. The widget that is specified is generally the application shell returned from the call to XtVaAppInitialize, as the class name of the application is used to determine the resource database maintained for this application.

The second parameter, base, is an address to the structure that will be used to store the values obtained by the call to XtVaGetApplicationResources. The relationship to the fields of the structure referenced by the second parameter and the resources will be clear shortly.

The next parameter, XtResourceList resources, is an array of XtResource elements.

The XtResource structure is defined as follows in the Intrinsic.h header file:

```
typedef struct _XtResources {
   String resource_name;
   String resource_class;
   String resource_type;
   Cardinal resource_size;
   Cardinal resource_offset;
   String default_type;
   XtPointer default_addr;
} XtResource, *XtResourceList;
```

The field resource_name is the name assigned by the programmer to the resource being created for inclusion on the command line or in a resource file.



By convention, the first letter of a resource name is lowercase, and subsequent words are capitalized, as in cardHeight, useFrontPanel, and so on.

The resource_class enables resources to be affected by a single entry in a resource file by referencing the class name of the resource instead of their individual names. For instance, tabCardFont and bottomBannerFont can both be of the class Font. Using only the class reference in an .Xdefaults file would assign the value to both. The syntax in the resource file to support this would be as follows:

*Font: 8x10

A resource_type identifies the data type of the resource value. Table 13.3 shows many of the common resource types understood by Intrinsics and the related C type.

bie 15.5 nesource classes and types				
X Resource Type	С Туре			
XtRBoolean	Boolean			
XtRColor	XColor			
XtRCursor	XCursor			
XtRDimension	Dimension			
XtRFloat	Float			
XtRFont	Font			
XtRInt	int			
XtRShort	short			
XtRString	String			

Table 13.3 Resource Classes and Types

The resource_size specifies the number of bytes that the field of the structure referenced by base has reserved for this resource's value. The sizeof operator is used to determine this value.

The field resource_offset specifies the offset of the field in the structure referenced by base for storing the value associated with this variable. The XtOffsetOf function is used to determine this value.

The last two fields of the XtResource structure determine the default type and default value for this resource if no assignment is made either on the command line or in a resource file.

Listing 13.11 shows a code sample to properly define a resource array to be passed to the XtVaGetApplicationResource function.

```
Listing 13.11 The XtVaGetApplicationResources Function
```

```
1:
      #include <X11/Intrinsic.h>
2:
      #include <X11/StringDefs.h>
3:
      #include <stdio.h>
4:
      #include <stdlib.h>
5:
6:
7:
      struct _cmdArgs {
8:
         char
                 *inFile;
9:
         char
                 *outFile;
10:
         Boolean verbose;
11:
       };
12:
13:
       #define offset(field) XtOffsetOf(struct cmdArgs, field)
14:
       static XtResource resources[] = {
        { "inFile", "InFile", XtRString, sizeof (char *),
15:
16:
              offset( inFile ), XtRString, (XtPointer)NULL },
17:
        { "outFile", "OutFile", XtRString, sizeof (char *),
18:
19:
              offset( outFile ), XtRString, (XtPointer) NULL },
20:
        { "verbose", "Verbose", XtRBoolean, sizeof (Boolean),
21:
22:
              offset(verbose), XtRString, (XtPointer)"off" },
23:
       };
       #undef offset
24:
25:
26:
       static XrmOptionDescRec optionList[] = {
                           "*inFile",
27:
         { "-inFile",
                                           XrmoptionSepArg, (XPointer) NULL },
28:
         { "-outFile",
                           "*outFile",
                                           XrmoptionSepArg, (XPointer) NULL },
                           "*verbose",
         { "-verbose",
                                          XrmoptionNoArg, (XPointer) "on" },
29:
30:
       };
31:
32:
       void main( int argc, char **argv )
33:
       {
34:
          struct _cmdArgs cmdArguments;
35:
36:
          XtAppContext appC;
37:
          Widget
                       toplevel;
38:
39:
          toplevel = XtVaAppInitialize( &appC, "Demo",
40:
                          optionList, XtNumber(optionList),
41:
                                         &argc, argv,
42:
                                         NULL, NULL );
```

```
43:
44:
           XtVaGetApplicationResources( toplevel,
45:
                                       (XtPointer)&cmdArguments,
46;
                                       resources, XtNumber(resources),
47:
                                       NULL);
48:
49:
50:
         if( cmdArguments.inFile )
           printf( "infile = %s\n", cmdArguments.inFile );
51:
52:
53:
         if( cmdArguments.outFile )
54:
           printf( "outfile = %s\n", cmdArguments.outFile );
55:
       }
```

Following the definition of a structure to contain all the values supported on the command line, the array resources is defined to inform Intrinsics of all the details about the resources understood by the application.

The resource names, classes, sizes, placement in the _cmdArgs structure, and even default values are provided in the XtResource array.

Next, a second array is created for inclusion in the call to XtVaAppInitialize in order to inform of the command-line arguments:

```
26: static XrmOptionDescRec optionList[] = {
```

When the call to XtVaAppIniatalize is made, all the resources known by the application are loaded into the X Resource Database. The second field to the optionList informed the database manager of pertinent resource file entries that would satisfy these resources and the first field instructed the valid command-line parameters.

I■ Note

Resource names do not have to be the same as the field names of the structure which houses them. I've simply done this for clarity in determining the relationships of the resources, structure fields, and command-line options.

Finally, when the call to XtVaGetApplicationResources is made specifying the address of the cmdArguments structure, the values for the resources specified in the array are copied from the resource database to the offsets of each of the fields in the structure.

The same caveat must be applied here as was given to the getopts implementation for command-line processing. If a required argument is omitted following a flag that requires it, there is no enforcement.

Continuing with the progression of the Graphics Editor project, it is important to establish a complete understanding of how the canvas will receive and process the events needed for object creation and management.

Setting Up a Canvas

Focusing attention on the routine create_canvas responsible for creating the drawing area GxDrawArea in Listing 13.4, we must be clear in the understanding of the relationship to the event handlers assigned to the widget

```
    34: XtAddEventHandler(GxDrawArea, PointerMotionMask, False,
    35: (XtEventHandler)drawAreaEventProc, (XtPointer)NULL);
    36: XtAddEventHandler(GxDrawArea, ButtonPressMask|ButtonReleaseMask, False,
    37: (XtEventHandler)drawAreaEventProc, (XtPointer)NULL);
```

and the actual cursor management routines found in Listing 13.6 at lines 82-88

```
82: void draw_manager( Widget w, XtPointer cd, XtPointer cbs )
83: {
84: void (*draw_func)( XEvent * ) = (void (*)(XEvent *))cd;
85:
86: if( draw_func != NULL ) (*draw_func)( NULL );
87: draw_mgr_func = draw_func;
88: }
```

and lines 99-105:

As you might recall, the draw_manager routine was assigned as the command widget callback function for all the gxDrawIcons parsed and created by the create_icons function.

Review the client data passed as the last parameter to the XtAddCallback function in create_icons:

Earlier I pointed out that the iconData reference is an element of the gxDrawIcons array. Looking again at the gxDrawIcons array definition, you see the object creation routines are being passed to the draw_manager callback function as client data when the user selects one of the object's drawing icons.

When the draw_manager function is invoked because a user selects one of these icons, it casts the client data to a function of type void, which expects an XEvent structure pointer as its only parameter:

84: void (*draw_func)(XEvent *) = (void (*)(XEvent *))cd;

If the function pointer is not NULL, the function is invoked with a NULL XEvent pointer reference to ensure that it has been reset from any previous creation:

```
86: if( draw_func != NULL ) (*draw_func)( NULL );
```

Finally, the draw_manager function assigns the function pointer value to a global variable called draw_mgr_func:

```
87: draw_mgr_func = draw_func;
```

All this occurs as a result of a graphic object draw icon being selected and it finishes before the user has even had a chance to move the cursor to the GxDrawArea canvas.

How does the event handler drawAreaEventProc assigned to the GxDrawArea relate to the draw_manager function?

Every time a ButtonPress, ButtonRelease, or PointerMotion event occurs in the window of the GxDrawArea widget, the drawAreaEventProc is invoked.

If the value of the global variable draw_mgr_func is not NULL, the drawAreaEventProc continually invokes the routine passing to it the current event:

104: if(draw_mgr_func != NULL) (*draw_mgr_func)(event);

The object-specific creation and management routines are introduced in Chapter 15, "Common Object Definition." These functions are responsible for performing the correct action based on current event type being sent.

Building the Project

Many files have been introduced in this chapter. All of them must be compiled and linked together to form a functional executable.

Chapter 1 provides an excellent introduction to the UNIX make utility and the make configuration file syntax. In fact, the sample configuration files can be used verbatim to build the source code files introduced here.



If the examples for building the project introduced in Chapter 1 are used *as is*, the proposed project structure must be honored or the vpath values will not be correct.

 \rightarrow See Chapter 1, section "Makefile," page 31, for a complete review of the make utility and its configuration syntax.

Next Steps

After you have successfully compiled and linked the files introduced in this chapter (and ensured that the elements are functional), you are ready to advance to the next chapter.

Chapter 14, "Program Flow," provides a brief discussion of the overall flow and program execution to clarify how functions such as event handlers and callbacks are invoked.

Chapter 14

In this chapter

- Processing Events
- X Event Hooks
- Next Steps

Program Flow

Chapter 13, "Application Structure," focused on structuring the application and creating the graphical user interface. This chapter explains in more depth the nature of event-driven programming and the interaction between the application and the X Window System.

When the Graphics Editor is built and executed from the source code presented in Chapter 13, control is quickly relinquished to the XtAppMainLoop function.

Beyond the creation of the graphical user interface and initialization of global variables for use during program execution, the program does not follow the conventional program flow that you might have experienced in the past.

After the XtAppMainLoop is entered, execution lies somewhere in the Intrinsics library well out of sight, only returning control to the Graphics Editor through one of several *entry points*.

It is not possible to predict absolutely where in the code execution will be at any given moment. Nor is it possible to predict the order in which functions will be entered because program flow at this point depends on several factors, primary of which are the habits of the user.

The user's navigation of the application leads to the generation of events. These events can be in the form of ButtonPress, ButtonRelease, or PointerMotion in the drawing area window or callbacks invoked from the selection of a command widget. Other events of interest to the Graphics Editor are the EnterNotify and LeaveNotify events for updating the context-sensitive help message in the status window.



All the events registered with the widgets that form the user interface account for the multiple entry points into the editor application.

The event-driven behavior of an X-based application directly affects the heuristic method applied to the development of the application. Specifically, the programmer must account for and be prepared for entry into the application from any one of the many points registered with X for responding to user input.

Processing Events

The X events sent by the server to an application in response to user actions are, in fact, data structures. Approximately 33 different event structures are defined by the X Window System, and all of these events are relative to a window. Behaviors such as a mouse pointer motion, entering and leaving windows, and even requests to change the width or height of a window are communicated to the application by an appropriate event structure being filled and passed by the X Server to the client.

Of the 33 events understood by X, most are not communicated to the application unless explicitly requested. As was demonstrated in Chapter 13, section "Setting up a Canvas," page 296, specifying the appropriate event mask for events of interest requests of the X Server that these events are communicated to the application.

The most complex aspect of X Window System programming is the task of creating a function to account for all possible events that can occur in the windows and subsequently invoking the functions registered by the user for the event. This task is commonly known as the *event-loop*.

Fortunately, the X Toolkit Intrinsics, through the library functions XtAppNextEvent and XtDispatchEvent, simplifies the task. An understanding of how the presence of events within a window of the application translates to a function being called by the X Server is important to this discussion.

Several *hooks* are available for the X Window System to communicate events to an application. These mechanisms included callbacks, event handlers, and translations.

EXCURSION

Application Program Hooks

In computer programming, a *hook* is a place (usually accessed through an interface) provided in packaged code that enables a programmer to attach customized functions. This enables a programmer to insert additional capability into the package or library. In the context of the X Window System, these hooks enable a programmer to specify re-entry points to the application based on the occurrence of specific events.

Typically, hooks are provided for a stated purpose and are documented for the programmer. Some writers use *hook* to also refer to the function that is inserted. As discussed, there is no assurance of when events will be communicated to the client application. Nor can it be known which event hook will be invoked at any given time.

Two reasons attribute to the seeming randomness by which events are communicated to an application are the *asynchronous* nature of X, and (as mentioned) the habits of the user as she navigates the application.

EXCURSION

The Asynchronous Nature of the X Window System

The X Server handles events *asynchronously*, meaning that events can occur in any order, and there is not a linear relationship to the requests generated by an application and the events that are returned to satisfy them.

The X Server takes a continuous stream of events from the display hardware and dispatches them accordingly to the appropriate applications (remember, an X Server addresses more than one client at any given time), which then take appropriate actions.

Four types of messages are passing between the client and server:

- *Requests* in which the client asks the server to perform a task or provide information.
- *Replies* from the server that can be immediate if the request by the client is a *roundtrip* request such as XQueryGeometry.
- Events that the X Server can surprise the client with in response to user action.
- Errors reported by the server to reflect an unrecoverable condition.



Asynchronous implies that the X Server sends events whether or not the client is ready to receive them.

Many X1ib functions (requests) made by the application cause the X Server to generate events. Additionally, the user's typing or moving the pointer will generate events. Both requests and events are buffered and the X Server addresses them *asynchronously* to maximize efficient use of the network.

When debugging an X Window application, it is often very convenient to request the X Server behave synchronously so that errors are reported at the moment they occur.

The following function lets you disable or enable synchronous behavior:

```
XSynchronous( Display *, Boolean );
```

The second parameter is expected to be either True or False indicating whether the X Server should start (True) or stop (False) its synchronous behavior.

On POSIX-conformant systems a global variable _Xdebug exists that, if set to a non-zero value before starting the program, will force synchronous behavior.

Note that graphics can update significantly slower (30 times or more) when synchronization is enabled, and it is therefore only used for debugging.

In normal asynchronous operations, X Window clients should be constantly prepared for any event that could be sent to them. For instance, windows within an application can be obscured by other applications and then exposed, requiring the applications to redraw.
X Event Hooks

The following sections address the mechanisms by which X invokes the functions registered as the event hooks for an application.

Widget Callbacks

The X Server generates events as it monitors the display hardware. The manner in which the user manipulates widgets by pushing a button, selecting a list item, choosing a menu item, or even moving the mouse pointer results in the X Server creating a corresponding event.



Graphical, event-driven programming is different from traditional programming in that the program does not dictate the flow of control.

Instead, the users, by their choice of actions, indirectly generate events, which the program might or might not respond to.

One type of response to events is a *callback*. Each callback has an associated reason (event type) for which the programmer can register a procedure. This procedure is then called every time the associated reason or event occurs.

For example, if you registered a callback for a command widget, it would be called each time the button was pressed.

The Intrinsics library does not guarantee an order. This is because both the widget author and the application programmer can modify the entire contents of the callback list.

The widget writer is not required to document when the widget will internally add or remove callbacks from the list associating callback procedures to the widget. Therefore, the functionality contained in a callback procedure should be independent of the functionality contained in other callback functions registered by the programmer.

If a widget does not maintain a callback list for the reason (event) needed by the application programmer, an event handler can be assigned to the widget.

Event Handlers

When a widget is created, it will know how to respond to certain events, such as a window manager's request to change size, color, background, or border size. This knowledge enables a command widget to appear highlighted when the mouse is clicked within its window or a menu to cascade when selected by the user.

However, hooks do not exist for all events understood by a widget. The command widget will not inherently enable a programmer to associate a new procedure for

invocation at occurrence of the EnterNotify event as it does for ButtonPress. (Remember, at the presence of a ButtonPress event the command widget will call the function registered with the XtNcallback resource.)

To overcome this, a programmer would use the XtAddEventHandler to force a widget to call a registered action at the presence of a specific event.

As seen in Chapter 13, command widgets do not enable an action to be associated for the events notifying the widget when the mouse cursor enters or leaves the widget's window. The XtAddEventHandler expands a widget's capabilities by instructing it to look for and act on events that it otherwise would ignore.

The relationship between callbacks and event handlers should be clear. Specifically, callbacks are lists inherent to a widget used to associate procedures to events. Event handlers are a less direct approach that expands the capabilities of the widget.

Translation tables are a final method for linking an application function to the internals of a widget.

Widget Translation Tables

Every widget has a *translation table* that defines how it will respond to particular events. These events can invoke one or more *actions*.

An example of part of the translation table for the command widget is the following:

```
BSelect Press: Arm()
BSelect Click: Activate(), Disarm()
```

BSelect Press corresponds to a mouse press and the translation table associates the action Arm to this translation. The Arm action maps to a function internal to the widget that causes, for instance, the command widget to highlight for appearing pressed.

If the mouse is clicked (pressed and released), Activate and Disarm functions are called. The Activate function internal to the widget looks for functions registered with the widget resource XtNcallback and invokes those it finds. The Disarm function returns the widget to normal coloring by turning off the widget highlighting.

Keyboard events can also be listed in the table to provide facilities such as hot keys, function or numeric select keys, and help facilities.

Examples of keyboard translations include KActivate, which typically refers to the Enter key, or KHelp for the HELP or F1 key. Pertinent actions are associated with these translations.

An advanced feature of the X Window System is the capability to add actions to existing translations or add new translations to a widget's translation table.

This feature, though not exercised in the Graphics Editor application, is employed with the functions XtParseTranslationTable and XtAugmentTranslations.

The function XtParseTranslationTable, as the name implies, parses a translation table established by the application developer, the result of which can be installed with the function XtAugmentTranslations.

The introduction of translation tables provides a complete coverage of the hooks available for an application developer to expand and advance the X Window environment.

Next Steps

The discussion of the application's flow in this chapter should bridge the understanding of the functions you author and their relationship to the processing of X Window events.

With a better understanding of the program flow of the Graphics Editor and its relationship to the X libraries and Server, you are ready to structure the graphic objects that are used to represent entities of the application.

The next chapter will advance the structure of the Graphics Editor project by building the common object component from which all objects in the editor will be based.

Chapter 15

In this chapter

- Line Object Data Structure
- Text Object Data Structure
- Arc Object Data Structure
- Common Object Data Structure
- Next Steps

Common Object Definition

Following the detailed look provided in Chapter 14, "Program Flow," of the Graphics Editor program flow and, specifically, the X Event hooks important to the editor project, you are ready to advance the structure of the Graphics Editor. This chapter presents the common object definition from which all objects supported by the editor are created.



All the data structures introduced in this chapter are targeted for inclusion in the gxGraphics.h header file.

→ Chapter 9, "Object Bounds Checking," page 203, introduced the idea that the objects supported by the Graphics Editor are divided into two families: point-array-based objects and arc objects.

Line Object Data Structure

The line objects presented in Figure 15.1 include the LatexLine (star shape), PolyLine, Box, and Arrow graphic objects.

These objects are all point-array-based: An array of points is used to represent them in the Graphics Editor application.





Notice in Figure 15.1 that the PolyLine object is *selected* as indicated by the scale handles surrounding the object.

The section "Common Object Data Structure," later in this chapter, explains fully the necessity of selecting the objects created in the Graphics Editor application and the data fields that manage an object's activity.

Listing 15.1 provides a structure for representing all the objects shown in Figure 15.1.

```
Listing 15.1 The Graphic Line Object Structure Definition
```

```
1: typedef struct _gxline {
2: XPoint *pts;
3: int num_pts;
4: } GXLine, *GXLinePtr;
```

The GXLine structure is very simple, requiring only the array of points

2: XPoint *pts;

and the number of elements contained in the array:

3: int num_pts;

The XPoint array pts stores the vertices of the line object in the elements of the array. Clearly, the number of vertices corresponds directly to the number of array elements as reflected by num_pts.

This structure definition declares two new data types for use in the Graphics Editor application:

4: } GXLine, *GXLinePtr;

The GxLine data type will refer to an actual occurrence of the structure and GxLinePtr, as the name implies, will point to an occurrence of the structure.

Not nearly as simple is the structure required for representing a vector text object in the Graphics Editor application.

Text Object Data Structure

The vector text object is the most complex object supported by the Graphics Editor application. What makes it so complicated is that the text object comprises many lines.

To illustrate, try the following: Using a piece of scrap paper or the back of your hand, try to draw the letter T (consisting of two lines—a vertical stem and a horizon-tal crossbar) without picking up your pencil and without backtracking over a line that you've already drawn.

The need to lift the pen in order to put the crossbar on the stem should be evident.

To accomplish this programmatically, the Graphics Editor uses an array of lines. As seen in the previous section introducing the Line Object data structure, the line object is itself an array of points. In the end, the text object data structure is an array of arrays where implicitly the routine for drawing the lines that compose a character lifts the pen before drawing the next line.

The concept of lifting a pen, of course, is figurative. The text draw routine accomplishes this by issuing separate draw requests for each line in the array.

Figure 15.2 shows a sample screen of three text objects created by the Graphics Editor application.



The data structure used to represent the Graphics Editor Text is shown in Listing 15.2.

```
Listing 15.2 The Graphic Text Object Structure Definition
```

```
1:
      typedef struct _gxtext {
2:
          int x, y;
                        /* upper-left */
3:
          int dx, dy;
4:
          char *text;
5:
          int len;
6:
          GXFont vpts;
7:
          GXFont font;
8:
          GXFontP fontp;
9:
      } GXText, *GXTextPtr;
```

The GXText data type introduced in Listing 15.2 contains elements for tracking the position in the drawing area of the text string managed by the object

2: int x, y; /* upper-left */

as well as the actual string and the string's length:

4: char *text; 5: int len;

The fields of the structure

3: int dx, dy;

are used to manage the scale feature of the text object and will be discussed in detail in Chapter 24, "Vector Text Object." The remaining three fields

6: GXFont vpts; 7: GXFont font; 8: GXFontP fontp;

are required to manage specifics of the character definitions for the vector font set that is employed for this graphic text object.

The font and fontp fields

7: GXFont font;8: GXFontP fontp;

are effectively read-only and refer to the font assigned to this object.

A significant complication related to the vector text object involves the font or character definitions that govern the appearance of characters drawn by the text object.

The following section introduces vector font concepts and the complicated data management employed to support them.

Understanding Vector Fonts

Any font that is defined by mathematical routines that can reproduce the outlines of each character at any size is a vector font.

Vector fonts, also known as *scalable*, *hershey*, or *outline fonts*, describe the shape of each character but not the character's size.

A character's shape definition consists of the vectors (positions relative to an origin) of the endpoints of the line segments that comprise each *stroke* of the character.

For example, consider the shape definition shown in Listing 15.3 for drawing the letter A using a simple vector font.

```
Listing 15.3 Vector Font Character Definition for the Letter A
```

```
static XPoint seg0 A[] = {
1:
2:
           \{0, -12\}, \{-8, 9\},\
3:
      };
4:
      static XPoint seg1_A[] = {
5:
           \{0, -12\}, \{8, 9\},\
6:
      };
      static XPoint seg2_A[] = {
7:
8:
           \{-5,2\}, \{5,2\},\
9:
      };
10:
      static XPoint *char65[] = {
11:
           seg0_A, seg1_A, seg2_A,
12:
           NULL,
13:
      };
```

The character definition for the letter A shown in Listing 15.3 consists of three line segments. Figure 15.3 illustrates the orientation of the vectors comprising each of the segments.



Representation of the line segments defining the letter A for a simple vector font.



As shown in Figure 15.3, the origin of the graph defining a character in this vector font set is located in the center of the cell. As this origin differs from the origin used by the X graphic primitives, the definition of the text object data structure must account for correctly placing the text by tracking the character string's location.

A review of what you now know about vector fonts is important.

A single character consists of multiple line segments that are composed of multiple points. In other words, an array of points forms a line segment and an array of line segments (array of points) defines a character.

Putting this all together, we can say that an array of arrays of points forms a character. Therefore, an entire font set (that is all character definitions) is an array of characters, which is an array of arrays of points. An entire font set is represented as

```
XPoint **plain_simplex[] = {
          char33,
                   char34,
                            char35,
                                     char36,
 char32,
                                              char37,
 char38,
          char39,
                   char40,
                            char41,
                                     char42,
                                              char43,
 char44,
          char45,
                   char46,
                            char47,
                                     char48,
                                              char49,
          char51,
                  char52,
                            char53,
 char50,
                                     char54,
                                              char55,
 char56, char57,
                   char58,
                            char59,
                                     char60,
                                              char61,
 char62, char63,
                  char64,
                            char65,
                                     char66, char67,
                            char71,
          char69,
                  char70,
 char68,
                                     char72,
                                              char73,
                   char76,
 char74,
          char75,
                            char77,
                                     char78,
                                              char79,
 char80,
          char81,
                   char82,
                            char83,
                                     char84,
                                              char85
          char87,
                   char88,
 char86,
                            char89,
                                     char90,
                                              char91,
 char92,
         char93,
                  char94,
                            char95,
                                     char96,
                                             char97,
```

```
char98, char99, char100, char101, char102, char103,
char104, char105, char106, char107, char108, char109,
char110, char111, char112, char113, char114, char115
char115 char116, char117, char118, char119, char120,
char121, char122, char123, char124, char125, char126,
};
```

The elements of the plain_simplex font set are character definitions. The arrays of line segments defining each of the characters are presented in Chapter 24, "Vector Text Object." Important to our present discussion is the understanding that the font set consists of an array in which each element is itself an array and each of these arrays further contain arrays of points.

The GXFont data type has been defined in the Graphics Editor project to represent a font set.

```
typedef XPoint ***GXFont;
```

The length of the arrays is important to every array nested in the GXFont data type defining a vector font set. To track this important length information, a second array is defined:

typedef int ** GXFontP;

The GXFontP array, however, contains elements of type int where each element corresponds to a length value for the corresponding element in the GXFont data type.

Use of these two arrays is closely linked and follows the convention reflected in the following code snippet:

```
static void GXDrawText( GXTextPtr text, GC gc )
1:
2:
      {
3:
        char *txt = text->text;
4:
        int c, chr, nsegs, num_pts;
5:
6:
        for( c = 0; c < text->len; c++, txt++ ) {
7:
          chr = *txt - ' ';
8:
          nseqs = 0;
9:
10:
          while( text->font[chr][nsegs] != NULL ) {
11:
            num pts = text->fontp[chr][nsegs];
12:
13:
            if( num pts > 0 ) {
              XDrawLines( XtDisplay(GxDrawArea),
14:
15:
                           XtWindow(GxDrawArea), gc,
                           text->vpts[c][nsegs], num pts,
16:
17:
                           CoordModeOrigin );
18:
            }
19:
            nsegs++;
20:
          }
21:
        }
22:
      }
```

This snippet is the actual code used to draw the text string contained in a GXText reference. It is officially introduced in Chapter 24, but is included here to show how a vector font set is employed.

The routine begins by extracting the text string from the GXText structure referenced by text:

```
3: char *txt = text->text;
```

Then for each character in the text string

6: for(c = 0; c < text->len; c++, txt++) {

an index into the font set is calculated by subtracting the decimal value of a space (' ') from the character to be drawn:

7: chr = *txt - ' ';

This works because the characters in the font set are ordered identically to the printable characters found in the ASCII table as shown in Figure 15.4.

Figure 15.4	Dec	Char	Dec	Char	Dec	Char
The ASCII table	32	sp	64	@	96	•
1.0011 1.0001.	33	1	65	A	97	а
	34		66	в	98	b
	35	#	67	c	99	с
	36	\$	68	D	100	d
	37	%	69	E	101	е
	38	&	70	F	102	f
	39	· · ·	71	G	103	g
	40	(72	н	104	h
	41)	73	1	105	i
	42	*	74	J	106	j
	43	+	75	к	107	k
	44	· · ·	76	L L	108	1
	45	-	77	м	109	m
	46	· · ·	78	Ν	110	n
	47	/	79	0	111	0
	48	0	80	Р	112	р
	49	1	81	Q	113	q
	50	2	82	R	114	r
	51	3	83	s	115	s
	52	4	84	т	116	t
	53	5	85	U	117	u
	54	6	86	v	118	v
	55	7	87	w	119	w
	56	8	88	х	120	x
	57	9	89	Y	121	У
	58	:	90	z	122	z
	59	;	91	1	123	{
	60	<	92	1	124	1
	61	=	93	1	125	}
	62	>	94	^	126	~
	63	?	95	-		

Notice in Figure 15.4 that the decimal value of a space is 32. As the first character defined in the vector font set is the space (char32), if a space is to be rendered by the GXDrawText function, it subtracts a space (decimal 32) from a space resulting in the value of chr being 0. An index of zero corresponds to the first element of the array, which is in fact the definition of the space character.

Although a space can't be *drawn*, its definition in the vector font set is important for separating words an appropriate distance when it is used in a text string managed by the graphic text object.

Applying this to the letter A (char65) you see that 'A' – ' ' or 65 - 32 is 33, which corresponds to the 34^{th} element of the array, which is the character definition for char65.

After calculating a value for chr (index of the character definition in the font set), the GXDrawText function can use chr to index into the array defining the number of points for the segments of which this character is composed:

```
11: num_pts = text->fontp[chr][nsegs];
```

With this data, the routine has all the information it needs to draw a single segment of the character.

As we saw with the definition of a single character

```
static XPoint *char65[] = {
    seg0_A, seg1_A, seg2_A,
    NULL,
};
```

Note

the character definitions are NULL terminated. This enables us to loop until all segments of the character have been drawn:

```
10: while( text->font[chr][nsegs] != NULL ) {
```

After drawing a segment, the routine increments to the next one

19: nsegs++;

and again tests for NULL.

This finishes the introduction to vector fonts and simple character definitions. Chapter 25, "Introduction to PostScript," will complete the subject and provide the remaining character definitions for this font.

The following section continues the discussion of the text object data structure.

The GXText Data Structure

Looking closer at the requirements of the text object supported by the graphics editor application, notice that the vpts field found in the GXText data structure in Listing 15.2

6: GXFont vpts;

are the actual points used to draw the object to the canvas. The value of vpts is derived by transposing the character definitions in the font set for each character of the text string to the location referenced by the x and y fields of the GXText structure.

Understanding the GXText data structure and the complex GXFont data type will take some time. These definitions will be reviewed and explained in greater detail in Chapter 24.

The complexity of the GXText data structure is inversely proportional to the GXArc data structure introduced in the next section.

Arc Object Data Structure

The arc objects are the simplest data structure of the graphic objects supported by the Graphics Editor. Examples of the graphic arc object are illustrated in Figure 15.5.



The data types for the structure used to manage the graphic arc objects shown in Figure 15.5 are found in Listing 15.4

Listing 15.4 GXArc Data Type Definition

typedef XArc GXArc;
 typedef XArc *GXArcPtr;

Fortunately, the X Window System provides a structure for representing the Arc objects, and this structure is sufficient for use in the Graphics Editor.

With the specific data requirements for the various graphic objects defined and explained in the preceding sections, we are ready to introduce the common object data structure and demonstrate its relationship to the specific graphic objects.

Common Object Data Structure

All the graphic objects supported by the Graphics Editor application require many pieces of information.

These data fields are not repeated for every object type but are contained in a common object data structure from which individual objects are derived.

Listing 15.5 introduces the common object definition used by all graphic objects in the Graphics Editor.

Listing 15.5 Common Object Data Structure

```
typedef struct gx obj {
1:
2:
        /*
         * attributes
3:
         */
4:
5:
        Pixel fg; /* foreground */
6:
        Pixel bg; /* background */
7:
        Pixmap fs; /* fill style */
8:
               ls; /* line style */
9:
        int
10:
        int
               lw; /* line width */
11:
12:
        Boolean selected;
13:
14:
        XRectangle *handles;
15:
        int
                 num handles;
16:
        void *data;
17:
18:
19:
        void (*draw) ( struct gx obj * );
20:
        void (*erase) ( struct _gx_obj * );
21:
22:
        XRectangle *(*bounds)( struct _gx_obj * );
        Boolean (*find) ( struct _gx_obj *, XEvent * );
23:
24:
```

```
Listing 15.5 Continued
```

```
25:
        void (*select)
                          ( struct _gx_obj * );
26:
        void (*deselect) ( struct _gx_obj * );
27:
                          ( struct gx obj * );
        void (*copy)
28:
        void (*move) ( struct gx obj *, XEvent * );
29:
30:
        void (*scale) ( struct _gx_obj *, XEvent * );
31:
32:
        void (*action) ( struct _gx_obj *, XEvent * );
33:
34:
        /*
35:
         * link to other objects
         */
36:
37:
        struct _gx_obj *next;
38:
39:
      } GXObj, *GXObjPtr;
```

The first several fields maintain the values of the attributes assigned to an individual object:

```
5: Pixel fg; /* foreground */
6: Pixel bg; /* background */
7:
8: Pixmap fs; /* fill style */
9: int ls; /* line style */
10: int lw; /* line width */
```

These fields will be used to create the Graphic Context (GC) used to draw the object.

Following the attributes are fields that manage whether the object is selected

12: Boolean selected;

and drawing handles to reflect an active status:

14: XRectangle *handles; 15: int num_handles;

The selected field is set to True when the object is selected and False otherwise.

When an object is selected (or active), eight handles are drawn to reflect this status and indicate the scale directions available for changing the size of the object. Only when an object is active can it be scaled, moved, copied, or cut from the screen.

The next field in the common object data structure is used to store the objectspecific data structures for the various graphic object types understood by the editor:

17: void *data;

The data field can point to any one of the structures introduced early for representing the specific data requirements of the editor objects.

15

Assigning the data field a reference to a valid GXLinePtr, GXTextPtr, or GXArcPtr data structure uniquely creates the specific graphic type.

A specific relationship exists between the structure reference used in the data field assignment when an object is created and the value of the remaining fields of the common object data structure. These remaining fields are pointers to functions, and when nested in a structure definition they are called *methods*.

The following methods are defined for all objects and, as stated, the value of these methods will be relative the structure type referenced in the data field:

```
19:
       void (*draw) ( struct _gx_obj * );
20:
       void (*erase) ( struct gx obj * );
21:
       XRectangle *(*bounds)( struct _gx_obj * );
22:
23:
       Boolean (*find) ( struct _gx_obj *, XEvent * );
24:
       void (*select) ( struct _gx_obj * );
25:
       void (*deselect) ( struct _gx_obj * );
26:
       void (*copy) ( struct gx obj * );
27:
28:
       void (*move) ( struct _gx_obj *, XEvent * );
29:
30:
       void (*scale) ( struct gx obj *, XEvent * );
```

These methods are explained by their names. The draw and erase methods require a single parameter referencing the object to be drawn or erased.

The bounds method also requires a reference to the object and returns a reference to an XRectangle reflecting the bounds of the object specified.

The find method requires, in addition to an object reference, a reference to an XEvent structure defining the x and y location of the mouse cursor when the object selection was attempted.

The select and deselect methods toggle the selected field to reflect the current status and create or destroy the handles reference as required by the change in status.

The copy action is accessed by the icon on the button panel to the right of the graphical user interface. When an object is active (selected) and the copy button is clicked, the active object is duplicated, with the copy placed next to the original.

Finally, the move and scale methods are implied by the user's navigation of the mouse cursor. Because these actions are modal, meaning they are dependent on the user's wishes, the action field

32: void (*action) (struct _gx_obj *, XEvent *);

tracks the current mode (move or scale) in use.

The last field of the common object data structure

37: struct _gx_obj *next; /* entire list of all objects */

is used to create a *linked list*.

EXCURSION

What Is a Linked List?

A *linked list* is a programming construct that enables *nodes* (data structures) to be linked together by a reference to the next node:

```
node (head) -> node -> node -> node -> NULL
```

The usefulness of a linked list lies in the fact that it can contain an unknown number of nodes. Contrast this to an array that has a finite number of elements determined when the array is either defined or allocated.

It is important when using the linked list construct that the *head*, or beginning of the list, be meticulously maintained. If for any reason a reference to the beginning of the list is lost, the entire list is also lost. As demonstrated with this previous linked list, a NULL pointer marks the last node in the list.

Chapter 16, "Object Manipulation," section "Deleting an Object," page 321, and Chapter 17, "Utilities and Tools," section "Linked List Management," page 346, discuss functions used by the Graphics Editor for adding and removing nodes to the linked list supported by the common object data structure.

The linked list provided for in the common object data structure is used to maintain the objects created in the editor.

All objects created by the Graphics Editor will be appended to the next field of the previously created object with the first object defining the head of the list.

This completes the definition of the vital data structures required by the Graphics Editor application.

Next Steps

An understanding of the data fields and methods defined in the common object data structure will allow us to introduce functions that employ them.

Chapter 16 will demonstrate how the graphic objects created by the editor are treated identically despite the specific type represented by the object. This generic treatment of the graphic objects is accomplished by using the methods and data elements of the common object data structure.

Chapter 16

In this chapter

- Copying an Object
- Deleting an Object
- Refreshing Objects
- Parsing for an Object
- Managing Object Handles
- Mangaging the Status of an Object
- Processing User Navigation of Objects
- Next Steps

Object Manipulation

Chapter 15, "Common Object Definition," led you through the definition of the data structures used to create the common graphic object as well as the unique structures for representing the supported graphic object types in the Graphics Editor.

Using the methods and data fields provided by the common object data structure, the Graphics Editor can build functions capable of managing objects without regard to their type.

This chapter introduces the functions that manipulate objects generically.



Many of the functions presented in this chapter complete the stub (empty) functions used to satisfy necessary function resolutions in Chapter 13, "Application Structure."

All the functions introduced in this chapter are intended for inclusion in the gxGx.c source file of the Graphics Editor project.

Copying an Object

The source code in Listing 16.1 satisfies the function invoked when the Copy icon from the button panel of the Graphics Editor interface is pressed.

```
Listing 16.1 Source Code for the copy Function
```

```
1: void gx_copy( void )
2: {
3: if( gx0bjCurrent ) {
```

Listing 16.1 Continued

```
4: (*gxObjCurrent->copy)( gxObjCurrent );
5: gx_refresh();
6: } else
7: setStatus( "Select an object to copy!" );
8: }
```

The gx_copy function defined in Listing 16.1 checks for the presence of the global variable gx0bjCurrent

3: if(gxObjCurrent) {

which refers to the currently active object in the Graphics Editor application.

If there is a currently active object, the gx_copy function invokes the copy method contained in the common object structure definition, passing the current object to satisfy the function's parameter requirements:

```
4: (*gxObjCurrent->copy)( gxObjCurrent );
```

This method, as explained in Chapter 15, "Common Object Definition," in the section "Common Object Data Structure," page 315, is assigned based on the specific graphic object type contained in the data field of the common object data structure.

EXCURSION

C Is Not C++

If you are familiar with the C++ programming language, you know that a feature of the language is the capability of all objects to make an implicit self-reference with the keyword this.

Unfortunately, this mechanism is not available in the C programming language. For this reason, a consistent parameter requirement for the methods contained in the common object data structure is a reference to the graphic object whose method is being invoked.

Following the invocation of the copy method, all the objects are redrawn with a call to the gx_refresh function.

If there is not an object currently selected, the else clause of the initial test is entered

```
6: } else
7: setStatus( "Select an object to copy!" );
```

and an information message is placed in the status label, indicating the intended order of actions for copying an object.

Located near the Copy icon in the Graphics Editor button panel is the Cut icon. The function to satisfy this icon is discussed in the following section.

Deleting an Object

Previously introduced as a stub function in Chapter 13, the code in Listing 16.2 provides the functionality to remove the currently selected object from the canvas and the linked list of created objects.

```
Listing 16.2 Source Code for the delete Function
```

```
1:
      void gx delete( void )
2:
      {
3:
        GXObjPtr gx objs;
4:
        Boolean found;
5:
6:
        if( gxObjCurrent ) {
7:
8:
          (*gxObjCurrent->deselect)( gxObjCurrent );
9:
          (*gxObjCurrent->erase)( gxObjCurrent );
10:
11:
          if( gxObjCurrent->data ) {
12:
            XtFree((char *)gxObjCurrent->data );
13:
            gxObjCurrent->data = NULL;
14:
          }
15:
16:
          found = False;
17:
          gx_objs = gxObjHeader;
18:
19:
          /*
20:
           * find this object in the list and remove it
           */
21:
22:
          while( gx_objs->next && (!found)) {
23:
24:
            /* we want the reference just before us... */
25:
            found = (gx objs->next == gxObjCurrent );
26:
27:
            /* ...now instead of pointing at us,
28:
             * point to what we point to
29:
             */
            if( found ) {
30:
31:
              gx objs->next = gxObjCurrent->next;
32:
              gxObjCurrent->next = NULL;
33:
            } else {
34:
              gx_objs = gx_objs->next;
35:
            }
36:
          }
37:
38:
          /*
39:
           * if we didn't find ourselves in the list, we are the list!
           */
40:
41:
          if( found == False ) {
42:
            if( gxObjHeader == gxObjCurrent )
43:
              qxObjHeader = qxObjCurrent->next;
```

16

continues

Listing 16.2 Continued

```
else
44.
45:
              setStatus( "Panic: object to delete not found list!" );
46:
          }
47:
          /* do some house cleaning */
48:
          XtFree((char *)gxObjCurrent );
49:
50:
          gxObjCurrent = NULL;
51:
          /* redraw remaining objects (if any) */
52:
53:
          if( gxObjHeader ) gx_refresh();
54:
        } else
55:
          setStatus( "Select an object to delete!" );
56:
      }
```

As with the gx_copy function, the gx_delete routine shown in Listing 16.2 begins by testing for the presence of a currently selected object. In its absence, an information message is placed in the status label, providing a hint to the user of the intended sequence of events for deleting an object.

If, however, an object is currently selected, the body of the if statement is entered where the object's deselect method is invoked to remove the object's handles

8: (*gxObjCurrent->deselect)(gxObjCurrent);

and then the object is erased from the screen by a call to its erase method:

9: (*gxObjCurrent->erase)(gxObjCurrent);

A test for the existence of the object-specific data is performed

11: if(gxObjCurrent->data) {

and if it exists, the memory associated with it is returned to the heap:

12: XtFree((char *)gxObjCurrent->data); 13: gxObjCurrent->data = NULL;

It is then necessary to parse the entire linked list of objects in search of the one being deleted so that it can be removed from the list.

This process is started by assigning a variable to point to the beginning of the list:

17: gx_objs = gxObjHeader;

It is important that the variable gxObjHeader is not used directly because the reference will be advanced through the list and it is crucial that the beginning of the list is maintained; otherwise the entire list will be lost.

The construct of the loop

22: while(gx_objs->next && (!found)) {

reads, "So long as there is another object in the list (gx_objs->next) and the one being deleted has not been found (!found), continue looping."

The variable found is assigned directly the Boolean result of the comparison of the next object in the list and the object being deleted:

25: found = (gx_objs->next == gxObjCurrent);

As mentioned previously, when the objects match, found will interrupt the loop and the next field of the current object will point to the one being deleted.

When a match is found, it is removed from the list by reassigning what the current node's next field points to:

```
30: if( found ) {
31: gx_objs->next = gxObjCurrent->next;
32: gxObjCurrent->next = NULL;
```

As long as the object has not been found, progress continues through the list by moving to the next node:

EXCURSION

Managing Pointers in a Linked List

It is important that you understand how the fragment

31: gx_objs->next = gxObjCurrent->next;

removes gx0bjCurrent from the list of graphic objects known to the Graphics Editor application.

A linked list, introduced in Chapter 16, "Object Manipulation," in the Excursion "An Introduction to Linked Lists," page xx, is a construct that allows a dynamic number of items to be managed by an application.

The next field of the common object data structure enables us to link the graphic objects created in the Graphics Editor together.

Conceptually, we depict this list as

```
gxObjHeader->next->next->next->next->(NULL)
```

meaning the gx0bjHeader, whose value is a reference to the first object created by the application, has as the value of its next field a reference to the second object created, which has as the value of its next field a reference to the third and so forth. The most recently created object will not have a valid reference assigned to its next field, thus marking the end of the list.

When the list is traversed in search of the object being deleted, the value of gxObjCurrent is somewhere in the list.

```
gxObjHeader -> next -> next -> [next] -> (NULL)
```

On the first iteration of the traversal, the variable gx0bjs refers to the beginning of the list.

```
gxObjHeader -> next -> next -> next -> (NULL)

gxObjs
```

With subsequent iterations, the gx0bjs value is advanced to equal the next node.

This continues until the current object is found, at which point the next field *referring* to the current object is assigned the reference *pointed to* by the current object.

gxObjHeader
$$\rightarrow$$
 next \rightarrow next \times [next] \rightarrow (NULL)

Linked lists are covered again in Chapter 17, "Utilities and Tools," in the section "Linked List Management," page 346.

Notice again the test for the object being deleted being found:

25: found = (gx_objs->next == gxObjCurrent);

The test is against the next field of the gx_objs variable. This means that the gxObjHeader value (first object created) is not considered in the body of the loop. This is important because the head of a linked list is always a special case.

If the loop ends without the object being found

41: if(found == False) {

an explicit test is performed against the beginning of the list:

42: if(gxObjHeader == gxObjCurrent)

If the object being sought is indeed the beginning of the list, its removal is accomplished with the line

43: gxObjHeader = gxObjCurrent->next;

and a new linked list head is assigned.

Finally, the object is permanently deleted from memory

49: XtFree((char *)gxObjCurrent);

and the objects still maintained in the editor are refreshed to redraw any portion of them that might have been obscured by the object that was deleted.

53: if(gxObjHeader) gx_refresh();

The following section introduces the gx_refresh function used to redraw the entire list of objects controlled by the Graphics Editor application.

Refreshing Objects

To account for the immediate graphic nature of the X Window System as introduced in Chapter 4, "Windowing Concepts," in the section "Expose," page 122, it is often necessary to refresh the graphic objects displayed on the drawing area canvas. The gx_refresh function introduced in Listing 16.3 accomplishes this task.

Listing 16.3 Source Code for the refresh Function

```
1:
      void gx refresh( void )
2:
      {
3:
        GXObjPtr obj = gxObjHeader;
4:
5:
        while( obj ) {
6:
          (*obj->draw)( obj );
7:
          gx draw handles( obj );
8:
          obj = obj->next;
9:
        }
10:
      }
```

The gx_refresh function starts at the beginning of the linked list

3: GXObjPtr obj = gxObjHeader;

and loops until it finds the end of the list marked by NULL:

```
5: while( obj ) {
```

For every object in the list, the object's draw method is invoked

6: (*obj->draw)(obj);

its handles are updated

7: gx_draw_handles(obj);

and the loop proceeds to the next node in the list:

8: obj = obj->next;

Perhaps one of the simplest functions acting against the common object definition, gx_refresh, is one of the most vital. Without it the objects would never get redrawn to the drawing area window.

Another critical function is the one introduced in the next section that enables us to find an object selected by a ButtonPress event.

Parsing for an Object

The capability of a user to select an object by clicking the mouse cursor on it is a vital function in the Graphics Editor. By selecting a graphic object, the user can scale, move, copy, cut, rotate, and modify the attributes of an object.

The parse_all_objects function introduced in Listing 16.4 will compare the x and y components of an XEvent to all the objects managed by the Graphics Editor to determine whether an object has been selected.

```
Listing 16.4 Source Code for Parsing the Objects
```

```
static void
1:
2:
      parse_all_objects( GXObjPtr obj, XEvent *event,
3:
                          GXObjPtr *gx obj )
4:
      {
5:
        if( obj && (*gx obj == NULL) ) {
6:
          if( (*obj->find)( obj, event ) ) {
7:
            *gx obj = obj;
            setStatus( "Object found..." );
8:
9:
          } else {
10:
            setStatus( "No objects found..." );
11:
12:
            parse_all_objects( obj->next, event, gx_obj );
13:
          }
14:
        }
15:
      }
```

The parse_all_objects function expects three parameters. The first parameter is a reference to a graphic object where the current iteration of the search is to begin.

The second parameter is the XEvent containing the x, y location of the position of the mouse cursor when the mouse button was clicked.

Last is a pointer to a graphic object pointer:

3: GXObjPtr *gx_obj)

This enables the return of the object found by the parse function to the calling function.

This function is recursive, meaning that it calls itself. Therefore, the initial test of parse_all_objects is to ensure that neither the object selected by the event nor the end of the list has been found:

5: if(obj && (*gx_obj == NULL)) {

If there is an object to test and no object has been found, the return value of the object's find method is used to determine whether the event successfully selected the current object:

6: if((*obj->find)(obj, event)) {

A True returned from the find method indicates that the object should be saved in the return field of the parameter list:

7: *gx_obj = obj;

A status is reported and the recursion unfolds, returning the selected object.

The find method returning False leads to another level of recursion being invoked

12: parse_all_objects(obj->next, event, gx_obj);

with the next element of the list passed for testing against the event reference.

At some point, either an object will be deemed selected or the obj->next reference passed to the subsequent level of recursion will be NULL and the test

5: if(obj && (*gx_obj == NULL)) {

will end the cycle.

After the user successfully selects an object, many issues must be managed. The first consideration is indicating the active status to the user. This is accomplished by drawing the object's handles.

Managing Object Handles

When an object is active, handles are drawn surrounding the object to reflect the object's status. The gx_draw_handles code found in Listing 16.5 draws the handles associated with a selected object.

```
Listing 16.5 Source Code for the gx_draw_handles Function
```

```
1: void gx_draw_handles( GXObjPtr obj )
2: {
3: static unsigned char mask_bits[8][HNDL_SIZE] = {
4: {0x7f, 0x3f, 0x1f, 0x1f, 0x3f, 0x73, 0xe1, 0xc0},
5: {0x18, 0x3c, 0x7e, 0xff, 0x18, 0x18, 0x18, 0x00},
```

Listing 16.5 Continued

```
6:
          {0xfe, 0xfc, 0xf8, 0xf8, 0xfc, 0xce, 0x87, 0x03},
7:
          {0x10, 0x30, 0x70, 0xff, 0xff, 0x70, 0x30, 0x10},
8:
          {0x03, 0x87, 0xce, 0xfc, 0xf8, 0xf8, 0xfc, 0xfe},
9:
          {0x00, 0x18, 0x18, 0x18, 0xff, 0x7e, 0x3c, 0x18},
10:
          {0xc0, 0xe1, 0x73, 0x3f, 0x1f, 0x1f, 0x3f, 0x7f},
11:
          {0x08, 0x0c, 0x0e, 0xff, 0xff, 0x0e, 0x0c, 0x08} };
12:
13:
        static Boolean masks_created = False;
14:
        static Pixmap masks[8];
15:
16:
        GC gc;
17:
        int i;
18:
19:
        if( masks created == False ) {
20:
          for( i = 0; i < 8; i++ )
21:
            masks[i] =
              XCreatePixmapFromBitmapData(XtDisplay(GxDrawArea),
22:
23:
                                           XtWindow(GxDrawArea),
24:
                                           (char *)mask bits[i],
25:
                                           HNDL SIZE, HNDL SIZE,
26:
                                            1, 0, 1);
27:
28:
          masks_created = True;
29:
        }
30:
31:
        if( obj && obj->handles && obj->num handles > 0 ) {
32:
          gc = gx allocate gc( obj, False );
          XSetFillStyle( XtDisplay(GxDrawArea), gc, FillSolid );
33:
34:
35:
          if( obj->num handles != 8 ) {
36:
            fprintf( stderr,
37:
             "Don't know how to draw objects with %d handles\n",
38:
                      obj->num handles );
39:
          } else {
40:
            for(i = 0; i < 8; i++) {
41:
              XSetClipOrigin( XtDisplay(GxDrawArea), gc,
                               obj->handles[i].x,
42:
43:
                               obj->handles[i].y );
44:
              XSetClipMask(XtDisplay(GxDrawArea), gc, masks[i]);
45:
46:
              XFillRectangle( XtDisplay(GxDrawArea),
47:
                               XtWindow(GxDrawArea), gc,
48:
                               obj->handles[i].x,
49:
                               obj->handles[i].y,
50:
                               obj->handles[i].width,
                               obj->handles[i].height );
51:
52:
            }
53:
          }
54:
          XtReleaseGC(GxDrawArea, gc);
55:
        }
56:
      }
```

The gx_draw_handles function found in Listing 16.5 begins with the definition of an array of bitmapped character data:

static unsigned char mask bits[8][HNDL SIZE] = { 3: {0x7f, 0x3f, 0x1f, 0x1f, 0x3f, 0x73, 0xe1, 0xc0}, 4: 5: {0x18, 0x3c, 0x7e, 0xff, 0x18, 0x18, 0x18, 0x00}, 6: {0xfe, 0xfc, 0xf8, 0xf8, 0xfc, 0xce, 0x87, 0x03}, 7: {0x10, 0x30, 0x70, 0xff, 0xff, 0x70, 0x30, 0x10}, 8: {0x03, 0x87, 0xce, 0xfc, 0xf8, 0xf8, 0xfc, 0xfe}, 9: {0x00, 0x18, 0x18, 0x18, 0xff, 0x7e, 0x3c, 0x18}, 10: {0xc0, 0xe1, 0x73, 0x3f, 0x1f, 0x1f, 0x3f, 0x7f}, 11: {0x08, 0x0c, 0x0e, 0xff, 0xff, 0x0e, 0x0c, 0x08} };

Each element corresponds to an arrow indicating the direction that the handle will scale the object if this handle is selected. Because all the handles are based on a square, the HNDL_SIZE directive indicates the length of the handle bit data array.

It is defined in the gxGraphics.h header file as

```
#define HNDL_SIZE 8
```

A Pixmap created from each of the elements of bit data defined by the array mask_bits need only be created once. To manage this a static flag

13: static Boolean masks_created = False;

is used to ensure that only with the first call to this function the Pixmaps are created:

21:	masks[i] =
22:	<pre>XCreatePixmapFromBitmapData(XtDisplay(GxDrawArea),</pre>
23:	XtWindow(GxDrawArea),
24:	(char *)mask_bits[i],
25:	HNDL_SIZE, HNDL_SIZE,
26:	1, 0, 1);

Notice the last parameter passed to the XCreatePixmapFromBitmapData is a value of 1. Because this parameter specifies the depth of the Pixmap created, the result is a Pixmap with a depth of 1, which is synonymous with a Bitmap and satisfies the definition of a valid ClipMask.

After the Pixmaps have been created, a test is performed to ensure that the object's handles have been created and in fact the object is active:

```
31: if( obj && obj->handles && obj->num_handles > 0 ) {
```

If there are handles to be drawn, a Graphics Context is created using the gx_allocate_gc utility introduced in Chapter 17, "Utilities and Tools":

32: gc = gx_allocate_gc(obj, False);

16

Because the creation of the Graphics Context is based on the attributes of the object specified as the first parameter of the creation utility, the FillStyle must explicitly be set to FillSolid for these handles to appear correctly:

33: XSetFillStyle(XtDisplay(GxDrawArea), gc, FillSolid);

Next, a test is performed to ensure the correct number of handles has been specified for the object:

```
35: if( obj->num_handles != 8 ) {
```

The test of the num_handles is necessary to prevent a potentially fatal error occurring when less than eight handles are created for the object and the subsequent for loop expects to index eight elements of the handles array:

40: for(i = 0; i < 8; i++) {

One further change to the Graphics Context employed for drawing the handles and the routine is complete. Specifically, the assignment of a ClipMask corresponding to the shape of the arrow Pixmap created from the bitmap data defined by the mask_bits array must be assigned to the gc. Always when using a ClipMask a ClipOrigin must be specified to align the mask with the item being drawn:

41:	XSetClipOrigin(XtDisplay(GxDrawArea), gc,
42:	obj->handles[i].x,
43:	obj->handles[i].y);
44:	<pre>XSetClipMask(XtDisplay(GxDrawArea), gc, masks[i]);</pre>

The origin specified for the arrow Pixmap used as the ClipMask is the same as the location for the position of the handle being drawn.

Finally, the individual handle is drawn using the XFillRectangle graphic primitive:

a),
), gc,
th,
.ght);
e a i

As a result of to the assignment of the ClipMask to the Graphics Context, the rectangle is drawn only where the bits of the arrow Pixmap are set to 1. In other words, the item being drawn is clipped (not drawn) where there is not a bit set in the ClipMask.

Finally, a call to return the Graphics Context to the pool cached by the X Toolkit is called to complete the function:

54: XtReleaseGC(GxDrawArea, gc);

To erase the handles associated with an object, the gx_erase_handles function shown in Listing 16.6 is used.

```
Listing 16.6 Source Code for the gx_erase_handles Function
```

```
void gx erase handles( GXObjPtr obj )
1:
2:
      {
        GC gc;
3:
4:
5:
        if( obj && obj->handles && obj->num handles > 0 ) {
6:
          gc = gx_allocate_gc( obj, True );
7:
8:
          XFillRectangles( XtDisplay(GxDrawArea),
9:
                            XtWindow(GxDrawArea), gc,
10:
                            obj->handles, obj->num handles );
11:
12:
          XtReleaseGC( GxDrawArea, gc );
13:
        }
14:
      }
```

A little simpler than the gx_draw_handles function, gx_erase_handles must only ensure the correct number of handles are assigned to the object

5: if(obj && obj->handles && obj->num_handles > \emptyset) {

and create a Graphics Context using the gx_allocate_gc utility:

6: gc = gx_allocate_gc(obj, True);

Notice a difference in the invocation of the gx_allocate_gc utility for the gx_draw_handles and gx_erase_handles functions.

The second parameter for this function indicates whether the gc should be created with a background tile assigned. A False as passed from the gx_draw_handles function is used to draw objects to the canvas and a True is used to erase them.

→ This function is presented in Chapter 17, "Utilities and Tools," in the section "Graphics Context Tiling," page 348, where the use of a background tile for erasing objects is discussed in detail.

Having looked at the first level of management required for objects activated by the user, namely managing the handles associated with the object, the discussion continues in the next section to managing the object's status.

Managing the Status of an Object

The behavior and treatment of a graphic object is influenced by the object's status. The user's navigation of the drawing area and specifically the explicit selection of objects that the user has created affect this status.

The process of selecting an object begins by the user clicking the mouse cursor while it is positioned over a graphic object. The Graphics Editor application must then apply the ButtonPress event to all the objects currently known to it in order to determine which object was targeted by the user's actions. 16

The source code that is found in Listing 16.7 for the find_graphic function begins the process of applying an event to the objects managed by the editor.

```
Listing 16.7 Source Code for the find graphic Function
```

```
1:
      static void find graphic( XEvent *event )
2:
      {
3:
        GXObjPtr gx obj = NULL;
4:
5:
        if( event->type == ButtonPress ) {
6:
          parse_all_objects( gxObjHeader, event, &gx_obj );
7:
8:
          if( gx_obj ) {
9:
            /* set this object as selected */
10:
            activate_obj( gx_obj );
11:
12:
          } else {
            /* deselect any currently selected */
13:
14.
            deactivate objs();
15:
          }
16:
        }
17:
      }
```

The function ensures that the event under consideration is a ButtonPress event:

5: if(event->type == ButtonPress) {

This is necessary to prevent events such as MotionNotify from erroneously selecting the graphic objects.

If the event type is correct, the function parse_all_objects introduced in Listing 16.4 is called to traverse the linked list of objects, invoking each object's find method in an attempt to discover any object that might have been targeted for selection:

6: parse_all_objects(gxObjHeader, event, &gx_obj);

If an object is returned from the parse_all_objects function, the object is activated:

```
8: if(gx_obj) {
9: /* set this object as selected */
10: activate_obj(gx_obj);
```

The source code for the activate_obj function is found in Listing 16.9.

Otherwise, if the event did not successfully return from any known object's find method, all objects are deactivated:

14: deactivate_objs();s

Listing 16.8 shows the source code for the deactivate_objs function.

Listing 16.8 Source Code for the deactivate objs Function

```
static void deactivate objs( void )
1:
2:
      {
3:
        GXObjPtr obj;
4:
5:
        if( gxObjCurrent ) {
6:
          obj = gxObjCurrent;
7:
8:
          (*obj->deselect)( obj );
9:
        }
10:
      }
```

The deactivate_objs function in Listing 16.8 is absolute. For any object currently active in the application, the deselect method is invoked. When the deactivate_objs function completes, there will be no active objects in the Graphics Editor application and the value of gxObjCurrent will be NULL.

Listing 16.9 Source Code for the activate_obj Function

```
static void activate obj( GXObjPtr obj )
1:
2:
      {
3:
        if( gxObjCurrent ) deactivate objs();
4:
        /* ensure we have an object */
5:
6:
        if( obj ) {
7:
          gxObjCurrent = obj;
8:
          (*obj->select)( obj );
9:
        }
10:
      }
```

The activate_objs function begins by considering that another object can be currently active:

3: if(gxObjCurrent) deactivate_objs();

If this is the case, the deactivate_objs function is invoked to toggle the object's status to inactive.

After ensuring that a valid object reference was passed to the activate_objs function

6: if(obj) {

the gxObjCurrent value is set to equal the newly found object. The object's select method is invoked to create the object's handles and set its status flag selected to True.

The user's selection of a graphic object from the Graphics Editor's canvas begins with the user's navigation of the interface. The following section demonstrates how the user's navigation actions are processed by the Graphics Editor application.

Processing User Navigation of Objects

Recall from Chapter 13, in the section "Setting Up a Canvas," page 296, that when we established the layout for the Graphics Editor's graphical user interface we assigned certain event handlers to the canvas area of the application.

Specifically, an event handler called drawAreaEventProc was assigned for the events PointerMotion, ButtonPress, and ButtonRelease. Listing 16.10 shows the source code for this event handler and its management of the different event types it receives.

```
Listing 16.10 Source Code for the drawAreaEventProc Function
```

```
void drawAreaEventProc( Widget w, XtPointer cd,
1:
2:
                               XEvent *event, Boolean flag )
3:
      {
4:
        if( draw_mgr_func != NULL ) {
5:
          (*draw_mgr_func)( event );
6:
        } else {
7:
          process_event( event );
        }
8:
9.
      }
```

The function is quite simple in that it only has to determine which branch of the application the event is targeted for.

If one of the drawing functions gx_line, gx_pencil, gx_box, gx_arrow, gx_arc, or gx_text is currently active, its value will be assigned to the function pointer draw_mgr_func as described in Chapter 13, in the section "Laying Out the User Interface," page 265.

An active drawing function gets preferential treatment

4: if(draw_mgr_func != NULL) {

and the event received by the drawAreaEventProc is sent to this active function:

5: (*draw_mgr_func)(event);

Otherwise, the process_event function that is found in Listing 16.11 is called to screen the event for the intended user action.

```
Listing 16.11 Source Code for the process_event Function
```

```
1: static void process_event( XEvent *xe )
2: {
3: if( gxObjCurrent == NULL ) {
4: if( xe && xe->type == ButtonPress ) {
5: find_graphic( xe );
6: }
```

The process_event function has two responsibilities. First, in the absence of a currently selected object, is to direct the event to the find_graphic function introduced in Listing 16.6:

5: find_graphic(xe);

However, if an object is already selected, the event is targeted for updating the active object by a call to update_obj:

```
9: update_obj( gxObjCurrent, xe );
```

The update_obj function must screen based on event type to determine the nature of object update required. Listing 16.12 shows the source code for the update_obj function.

```
Listing 16.12 Source Code for the update_obj Function
```

```
static void update obj( GXObjPtr obj, XEvent *event )
1:
2:
      {
3:
        switch( event->type ) {
4:
        case ButtonPress:
5:
          buttonpress update( obj, event );
6:
          break:
7:
        case ButtonRelease:
8.
9:
          buttonrelease update( obj, event );
10:
          break;
11:
12:
         case MotionNotify:
           motionnotify update( obj, event );
13:
14:
           break;
15:
        }
16:
      }
```

The update_obj function only has to switch on the event type to determine which update function to call. Listing 16.13 shows the source code for the first case buttonpress_update.

```
Listing 16.13 Source Code for the buttonpress_update Function
```

```
1: static void buttonpress_update( GXObjPtr obj,
2: XEvent *event )
3: {
4: Boolean found = False;
5: GXObjPtr e_obj = NULL;
```

Listing 16.13 Continued

```
indx = 0;
6:
        int
7:
8:
        if( obj->action == NULL ) {
9:
          FixedX = event->xbutton.x;
10:
           FixedY = event->xbutton.y;
11:
           find handle( obj, event, &found, &indx );
12:
13:
           if( found ) {
14:
             obj->action = obj->scale;
15:
16:
17:
             set cursor( SCALE MODE );
             GxActiveHandle = indx;
18:
19:
20:
           } else {
             parse_all_objects( obj, event, &e_obj );
21:
22:
             if( e obj == obj ) {
23:
               obj->action = obj->move;
                set_cursor( MOVE_MODE );
24:
25:
             } else {
26:
                deactivate_objs();
27:
             }
28:
           }
29:
         }
30:
31:
         if( obj->action ) {
32:
           (*obj->deselect)( obj );
33:
           (*obj->action)( obj, event );
34:
         }
35:
       }
```

The buttonpress_update function has the ultimate purpose of correctly assigning an action to the current object.

Two actions are possible, moving or scaling. The determination of the correct action assignment is based on whether the event occurred on one of the handles assigned to the active object or, alternately, whether the event occurred on the object itself.

The buttonpress_update function begins by testing for a current action assigned to the active object:

8: if(obj->action == NULL) {

If a current action is assigned, the body of the if statement is skipped and immediately the action is invoked for the current event:

```
31: if( obj->action ) {
32: (*obj->deselect)( obj );
33: (*obj->action)( obj, event );
34: }
```

The call to deselect the object before invoking the action

32: (*obj->deselect)(obj);

is to ensure that the handles are not displayed while the object is either scaled or moved.

If an object's action field does not have a current function assigned, the buttonpress_update function determines what the value should be.

Before determining the correct action assignment, the global variables

```
9: FixedX = event->xbutton.x;
10: FixedY = event->xbutton.y;
```

are updated with the x, y coordinates of the current event. This will be important during the scale action covered in Chapter 20 in the section "Scaling a Line Object," page 398.

To determine the proper value of the action field for the current object, the buttonpress_update determines whether one of the object's handles was successfully selected by the event being processed:

```
12: find_handle( obj, event, &found, &indx );
```

The source code for the find_handle routine is provided in Listing 16.14.

The find_handle function returns two values in the fields passed as the second and third parameters: a status flag indicating whether a handle was found and, optionally if found reflects that a handle was found, the index of the handle.

The successful selection of the object's handle indicates that the user intended to scale the object. In this case, the object's scale method is used as the value for the action field:

15: obj->action = obj->scale;

Two chores are addressed in preparation for the scale action. The first changes the cursor to reflect the new state of the application:

17: set_cursor(SCALE_MODE);

 \rightarrow This function is provided in Chapter 17, "Utilities and Tools," page 343.

18: GxActiveHandle = indx;

The second chore required in preparation for the scale action is to set the value of indx (reflecting the specific handle selected by the user) to the global variable GxActiveHandle.

This information is necessary to ensure that the object is scaled in a direction appropriate to the direction of the arrow of the selected object's handle.
If a handle was not selected, the body of the else is entered

20: } else {

and the buttonpress_update function parses all known objects to see whether the event landed on an object:

21: parse_all_objects(obj, event, &e_obj);

An event directed to the update branch of the Graphics Editor application that is located directly on the graphic object implies that the user intended for the object to be moved.

If the parse_all_objects function as shown in Listing 16.4 determines that an object is selected by the specified event, the object is returned in the last parameter field of the function call.

If the event selected the same object as the currently active object

22: if(e_obj == obj) {

the user's intent to move the object is confirmed and the action field is updated accordingly:

23: obj->action = obj->move;

The utility to update the applications cursor to reflect the new state of the application is called:

24: set_cursor(MOVE_MODE);

If the object selected by the event does not match the active object, the buttonpress_update function deactivates the current object and ends the update process:

25: } else {
26: deactivate_objs();

Another update function used by the update_obj routine manages the ButtonRelease event type. The source code for the buttonrelease_update function is found in Listing 16.14.

Listing 16.14 Source Code for the buttonrelease_update Function

```
1: static void buttonrelease_update( GXObjPtr obj,
2: XEvent *event )
3: {
4: /*
5: * reset the update function
6: */
```

```
7:
        if( obj->action ) {
          (*obj->action)( obj, event );
8:
9:
          (*obj->action)( obj, NULL );
10:
          obj->action = NULL;
11:
12:
          (*obj->select)( obj );
13:
          gx_refresh();
14:
        }
15:
16:
        set cursor( NORMAL MODE );
17:
      }
```

The task managed by the buttonrelease_update function depends entirely upon a valid assignment to the action field of the object.

If the object has a current action assigned, the action function is invoked with the event being processed:

8: (*obj->action)(obj, event);

By design, the event type is intended to end the iterative actions that affect an object.

Therefore, after the ButtonRelease event is passed to the current action function, the function is invoked a second time with a NULL event to allow the object to perform any necessary cleanup

9: (*obj->action)(obj, NULL);

and the current action is cleared:

```
11: obj->action = NULL;
```

Finally, the object's select method is invoked to redisplay the handles and the screen is refreshed:

```
12: (*obj->select)( obj );
13: gx_refresh();
```

The application's cursor is updated

16: set_cursor(NORMAL_MODE);

reflecting the new state of the application.

The final update function employed by the update_obj routine manages the MotionNotify event type. The source code for the motionnotify_update function is found in Listing 16.15.

16

Listing 16.15 Source Code for the motionnotify update Function

```
1: static void motionnotify_update( GXObjPtr obj,
2: XEvent *event )
3: {
4: if( obj->action ) {
5: (*obj->action)( obj, event );
6: }
7: }
```

As with the buttonrelease_update function, the motionnotify_update routine depends on the action assigned to the current object.

The notification of the mouse in motion is the heart of the iterative processing used to alter the current object. For this reason, if a current action is assigned to the object, the event is passed wholesale.

The final piece to the management of user navigation is the introduction of the find_handle routine used by the buttonpress_update routine found in Listing 16.13.

Listing 16.16 Source Code for the find_handle Function

```
1:
      static void find_handle( GXObjPtr gx_obj, XEvent *xe,
2:
                                Boolean *found, int *indx )
3:
      {
4:
        int i;
5:
        *found = False;
6:
7:
8:
        if( gx_obj && gx_obj->handles && !(*found)) {
9:
          for( i = 0; (i < gx obj->num handles) && !(*found); i++ ) {
            if( ( xe->xbutton.x > gx obj->handles[i].x )
10:
                                                                       &&
                ( xe->xbutton.x < gx_obj->handles[i].x + HNDL_SIZE) &&
11:
                ( xe->xbutton.y > gx obj->handles[i].y )
12:
                                                                       &&
                ( xe->xbutton.y < gx obj->handles[i].y + HNDL SIZE)) {
13:
14:
              *indx = i;
15:
               *found = True;
16:
            }
17:
          }
18:
        }
19:
      }
```

The find_handle function starts by ensuring that handles for this object exist and that a handle has yet to be found:

8: if(gx_obj && gx_obj->handles && !(*found)) {

Then for each handle known to the object, a series of greater-than/less-than tests are performed to see whether the coordinates of the XEvent lie within bounds of the object's handle.

&&

If the event's x component is greater than the starting point of the handle

and the x component is less than the end of the handle

11: (xe->xbutton.x < gx_obj->handles[i].x + HNDL_SIZE) &&

and the event's y component is greater than the start and less than the end of the handle

```
12: ( xe->xbutton.y > gx_obj->handles[i].y ) &&
13: ( xe->xbutton.y < gx_obj->handles[i].y + HNDL_SIZE)) {
```

the handle is considered selected. The found flag is updated to inform the calling function that a handle was found and the index of the satisfying handle is stored in the indx field:

14:	*indx	=	i;
15:	*found	=	True;

If none of the object's handles pass the test, the found flag is returned to the calling function as False, indicating no handle was found and the value of the indx field is immaterial.

This completes the discussion of routines in the Graphics Editor that manage objects based on common object data structure without regard to the specifics of the object's type.

Next Steps

This chapter covered in detail the management routines that act on objects generically, that is, despite their actual type.

In Chapter 17, more utilities and tools are introduced which address areas of the Graphics Editor application necessary for object creation, management, and house-keeping.

Chapter 17

In this chapter

- Common Object Creation
- Creating a Graphics Context
- Using the Cursor as State Indicator
- Next Steps

Utilities and Tools

Chapter 16 presented functions used by the Graphics Editor that employ the common or generic features of the graphic objects.

In this chapter, I introduce and discuss utility routines necessary to manage the graphical user interface as well as behind-the-scenes tool functions.

Common Object Creation

As a software developer, code redundancy should set off sirens in your analytical mind. For that reason, when an object is created in the Graphics Editor, a central gx_create_obj routine is used to create the object and assign the default attributes of the object rather than allowing each of the object-specific creation routines to repeat these necessary steps.

The function also enables the assignment of default methods to prevent an oversight in the object-specific code from resulting in a fatal bus error.

The source code in Listing 17.1 introduces the gx_create_obj function.

```
Listing 17.1 Source Code for the gx_create_obj Function for Inclusion in gxGx.c
```

```
1:
      GXObjPtr gx_create_obj( void )
2:
      {
3:
        GXObjPtr gx_obj = XtNew( GXObj );
4:
        gx obj->fs = None;
5:
6:
        gx_obj->ls = LineSolid;
7:
        gx obj -> lw = 1;
8:
9:
        qx obj->bg = WhitePixelOfScreen(XtScreen(GxDrawArea));
10:
        gx obj->fg = BlackPixelOfScreen(XtScreen(GxDrawArea));
```

Listing 17.1 Continued

```
11:
12:
        gx obj->handles
                            = NULL;
13:
        gx obj -> num handles = 0;
14:
15:
        gx obj->data = NULL;
16:
17:
        gx obj->draw
                         = (void
                                     (*)())null_func;
        gx_obj->erase
18:
                         = (void
                                     (*)())null_func;
        gx obj->find
                         = (Boolean (*)())null_func;
19:
        gx_obj->select = (void
20:
                                    (*)())null_func;
21:
        gx obj->deselect = (void
                                     (*)())null_func;
22:
        gx obj->move = (void
                                    (*)())null func;
23:
        gx_obj->scale = (void
                                     (*)())null_func;
24:
        gx obj->copy
                         = (void
                                     (*)())null func;
25:
26:
        gx obj->action
                         = NULL;
27:
28:
        gx obj->next = NULL;
29:
30:
        /* reset the draw mgr func so
                                        */
        /* future events are applied to */
31:
32:
        /* existing objects
                                         */
33:
        draw mgr func = NULL;
34:
35:
        return gx_obj;
36:
      }
```

The gx_create_obj routine begins by allocating the memory required for the new object:

3: GXObjPtr gx_obj = XtNew(GXObj);

With memory for the object created, the default attributes are assigned:

```
5: gx_obj->fs = None;
6: gx_obj->ls = LineSolid;
7: gx_obj->lw = 1;
8:
9: gx_obj->bg = WhitePixelOfScreen(XtScreen(GxDrawArea));
10: gx_obj->fg = BlackPixelOfScreen(XtScreen(GxDrawArea));
```

The next part of the gx_create_obj function initializes fields of the common object data structure to a safe state:

```
12: gx_obj->handles = NULL;
13: gx_obj->num_handles = 0;
14:
15: gx_obj->data = NULL;
```

Then a temporary function is assigned to the method fields of the object to prevent an erroneous invocation of an invalid function:

17:	gx_obj->draw	=	(void	<pre>(*)())null_func;</pre>
18:	gx_obj->erase	=	(void	<pre>(*)())null_func;</pre>
19:	gx_obj->find	=	(Boolean	<pre>(*)())null_func;</pre>
20:	gx_obj->select	=	(void	<pre>(*)())null_func;</pre>
21:	gx_obj->deselect	=	(void	<pre>(*)())null_func;</pre>
22:	gx_obj->move	=	(void	<pre>(*)())null_func;</pre>
23:	gx_obj->scale	=	(void	<pre>(*)())null_func;</pre>
24:	gx_obj->copy	=	(void	<pre>(*)())null_func;</pre>

The fail safe null_func routine is shown in Listing 17.1a.

Listing 17.1a Source Code for the null func Routine for Inclusion in gxGx.c

```
1a: static void null_func( void )
2a: {
3a: printf( "Warning: null function called!\n" );
4a: }
```

Because the object is newly created, there is not a current action to assign to it. Therefore, set the action field to reflect this:

26: gx_obj->action = NULL;

It is necessary to initialize the next field of the object structure that links to other objects. This is critical to the proper management of the linked list in which this object will be placed:

28: gx_obj->next = NULL;

The global variable draw_mgr_func is reset to NULL to ensure that subsequent events generated in the drawing area canvas are sent to the object that is created and set as active:

```
33: draw_mgr_func = NULL;
```

Finally, the new object is returned to the calling function:

```
35: return gx_obj;
```

Though the creation routines for the object-specific data structures are not introduced until Chapter 20, "Latex Line Object," you should understand (extrapolate) that the graphic object's specific creation routines employ the gx_create_obj routine.

Further, upon receiving the object created by the gx_create_obj function, the object-specific data is assigned to the data field of the object and the appropriate object's methods are updated to properly manage that data field.

For the editor to retain and continually manage the object, it must be added to the linked list of objects under the editor's control. The following section describes the linked list management function that adds nodes to the editor's linked list.

Linked List Management

→The Excursion "What is a Linked List?" found in Chapter 15, "Common Object Definition," describes a linked list as a programming construct that enables nodes (data structures) to be linked together.

Designating a head node as the first element in the list and continually assigning consecutive node references to the next field of the last node in the list accomplishes this.

The gx_add_obj routine is responsible for finding the end of the list (last node in the list) and assigning the reference to the node being added to the next element of that object structure.

Listing 17.2 shows the source code for gx_add_obj.

```
Listing 17.2 Source Code for the gx_add_obj Function for Inclusion in gxGx.c
```

```
void gx add obj( GXObjPtr obj )
1:
2:
      {
3:
        GXObjPtr gx obj;
4:
5:
        if( gxObjHeader == NULL ) {
6:
          gxObjHeader = obj;
7:
        } else {
8:
          gx_obj = gxObjHeader;
9:
10:
          /* find the end of the object list */
11:
          while( gx obj->next != NULL ) {
             gx_obj = gx_obj->next;
12:
13:
          }
14:
          /*
15:
           * add the new object to the end of our list
16:
           */
17:
18:
          gx obj->next = obj;
19:
        }
20:
      }
```

The gx_add_obj function begins by determining whether a list currently exists by testing the value of the gxObjHeader variable responsible for maintaining the list's head:

5: if(gxObjHeader == NULL) {

If the gxObjHeader variable is NULL, this is the first object created by the Graphics Editor, and it therefore assumes the responsibility of being the head of the list:

6: gxObjHeader = obj;

If a valid head already exists, however, the list must be traversed in search of its end.

To do this, a variable is assigned the value of the list head and a while loop is entered until the next element's value is NULL, as NULL marks the end of a linked list:

```
8: gx_obj = gxObjHeader;
9:
10: /* find the end of the object list */
11: while(gx_obj->next != NULL) {
```

As long as there is a valid element assigned to the next field of the current node, traversal continues

```
12: gx_obj = gx_obj->next;
```

When the end of the linked list is found, the element being added assumes a position as the new end of the list:

18: gx_obj->next = obj;

The capability to create and retain graphic objects is made more meaningful when the Graphics Editor can draw those objects to the canvas.

In the next section I introduce a utility function used by the editor to create a Graphics Context based on the current attribute settings of an object.

Creating a Graphics Context

The gx_allocate_gc function introduced in Listing 17.3 is a cornerstone of the Graphics Editor's functionality and assigns the XtGCValues fields based on the attribute settings of the object specified as the function's first parameter. With a properly constructed XtGCValues structure, a Graphics Context is created to honor the object's settings.

Listing 17.3 Source Code for the gx_allocate_gc Function for Inclusion in gxGx.c

```
1:
      GC gx_allocate_gc( GXObjPtr obj, Boolean tile )
2:
      {
3:
        GC gc;
4:
5:
        XGCValues values;
6:
        XtGCMask mask = 0L;
7:
        values.foreground = obj->fg;
8.
        mask |= GCForeground;
9:
10:
        values.background = obj->bg;
11:
        mask |= GCBackground;
12:
13:
14.
        values.line width = obj->lw;
15:
        mask |= GCLineWidth;
```

Listing 17.3 Continued

```
16:
17:
        values.line style = obj->ls;
18:
        mask |= GCLineStyle;
19:
20:
        if( tile ) {
21:
          values.tile = GxDrawAreaBG;
22:
          mask |= GCTile;
23:
24:
          values.fill style = FillTiled;
25:
          mask |= GCFillStyle;
26:
        }
27:
28:
        values.function = GXcopy;
        mask |= GCFunction;
29:
30:
        gc = XtAllocateGC(GxDrawArea, 0, mask, &values, mask, 0);
31:
32:
        return gc;
33:
      }
```

The gx_allocate_gc function is straightforward in the creation of a Graphics Context except for the use of the second parameter called tile, described in the following section.

Graphics Context Tiling

The value of the tile flag passed as the second parameter to the gx_allocate_gc function is specified by the caller to indicate whether the gc created by this routine is intended for a draw or an erase action.

If the value of tile is specified as True, the Graphics Context is created to *tile* the background into the draw request. The effect of this is that the background of the canvas replaces the foreground pixels of the X graphic primitive request.

The net result of the Graphics Context *tile* behavior is to effectively *erase* the item rendered with this gc from the canvas. I say *effectively* because you've still drawn the item, albeit not using the foreground color, but rather by applying the background to the pixels that would normally have been set to the foreground color value.

For the tile behavior to work correctly, a tile Pixmap must first be created. Therefore, I've added the lines shown in Listing 17.4 to the beginning of the create_canvas routine introduced in Chapter 13, "Application Structure."

```
Listing 17.4 Changes to Be Made to the create_canvas Routine in gxGraphics.c
```

```
1: Widget create_canvas( Widget parent )
```

```
2: {
```

```
3: GxDrawAreaBG =
```

4: XCreatePixmapFromBitmapData(XtDisplay(parent),

```
5:DefaultRootWindow(XtDisplay(parent)),6:grid_bits, grid_width, grid_height,7:BlackPixelOfScreen(XtScreen(parent)),8:WhitePixelOfScreen(XtScreen(parent)),9:DefaultDepthOfScreen(XtScreen(parent)));
```

This code fragment creates a Pixmap using the character bitmap data grid_bits shown in Listing 17.5.

Listing 17.5 Contents of the grid.xbm File

To be syntactically correct and prevent compiler errors associated with the changes shown in Listing 17.4, you must create the grid.xbm file from Listing 17.5 and place this file in the src/include directory.

Further, an include directive must be added to the beginning of gxGraphics.c for the grid.xbm file as follows:

```
#include "gxGraphics.h"
#include "grid.xbm"
```

Finally, the following line must be added to the GLOBAL section of the gxGraphic.h header file:

GLOBAL Pixmap GxDrawAreaBG;

The next section introduces how to alter the appearance of the applications cursor to reflect the status or state of the application.

Using the Cursor as State Indicator

The default cursor for the X Window Environment is the left arrow cursor. This is fine for normal operation but can be changed at will to reflect any of the number of states the application might enter.

The cursors available to an X-based application are found in the header file cursorfonts.h. The code in Listing 17.6 shows the Graphics Editor utility for changing the cursor based on the cursor modes known to the application.

Listing 17.6 Source Code for the set_cursor Function to Be Added to gxGraphics.c

```
void set_cursor( CursorMode mode )
1:
2:
      {
3:
         Cursor new cursor;
4:
5:
        switch( mode ) {
6:
          case LINE_MODE:
7:
            new cursor = gxCrosshair;
8:
            break;
9:
10:
          case PENCIL MODE:
11:
            new_cursor = gxPencil;
12:
            break;
13:
14:
          case EDIT MODE:
15:
            new_cursor = gxEdit;
16:
            break;
17:
18:
          case TEXT_MODE:
19:
            new cursor = gxTextI;
20:
            break;
21:
22:
          case SCALE MODE:
23:
            new_cursor = gxScale;
24:
            break;
25:
26:
          case MOVE MODE:
27:
            new_cursor = gxMove;
28:
            break;
29:
30:
          case NORMAL_MODE:
31:
          default:
32:
            new_cursor = gxNormal;
33:
            break;
34:
        }
35:
        XDefineCursor( XtDisplay(GxDrawArea),
36:
                        XtWindow(GxDrawArea), new_cursor );
37:
        XFlush( XtDisplay(GxDrawArea) );
38:
      }
```

The required parameter for the set_cursor function is of the data type CursorMode. This is a data type unique to the Graphics Editor and is defined as the enumeration shown in Listing 17.7.

Listing 17.7	The CursorMode Enumeration for Inclusion in the gxGraphics.h Header File
1:	typedef enum cursor mode {
2:	NORMAL_MODE = 0,
3:	PENCIL_MODE,
4:	EDIT_MODE,
5:	TEXT_MODE,
6:	MOVE_MODE,
7:	SCALE_MODE,
8:	LINE_MODE
9:	} CursorMode;

The set_cursor function switches on the value of the CursorMode specified to the function and sets the new_cursor value to one of the global variables shown in Listing 17.8.

Listing 17.8 Global Cursor References for Inclusion in the gxGraphics.c Source File

```
    Cursor gxCrosshair;
    Cursor gxPencil;
    Cursor gxEdit;
    Cursor gxNormal;
    Cursor gxTextI;
    Cursor gxMove;
    Cursor gxScale;
```

After the new_cursor value is determined, the cursor's corresponding cursor value is defined for the Window of the GxDrawArea widget

```
35: XDefineCursor( XtDisplay(GxDrawArea),
36: XtWindow(GxDrawArea), new_cursor);
```

and the expose event generated by the XDefineCursor function is flushed from the application's event queue to ensure that the new cursor appears immediately in the drawing area window:

37: XFlush(XtDisplay(GxDrawArea));

The Cursor variables shown in Listing 17.8 require values well in advance of the application employing them, so a new function called initializeGX has been added to the gxGraphics.c source file.

The contents of the initializeGX function are found in Listing 17.9. The function's purpose is to initialize all global variables used by the application in addition to the cursor variables in Listing 17.8.

```
Listing 17.9 Source Code for the initializeGX Function for the gxGraphics.c File
```

```
void initializeGX( void )
1:
2:
      {
3:
        Pixel color;
4:
        Display *dsp = XtDisplay(GxDrawArea);
5:
6:
        XtVaGetValues( GxStatusBar, XtNbackground, &color, NULL );
7:
8:
        gxObjHeader = NULL;
9:
        gxObjCurrent = NULL;
10:
11:
        gxCrosshair = XCreateFontCursor(dsp, XC crosshair);
12:
        gxPencil = XCreateFontCursor(dsp, XC pencil);
13:
        gxEdit
                    = XCreateFontCursor(dsp, XC_cross);
        gxTextI = XCreateFontCursor(dsp, XC xterm);
14:
                   = XCreateFontCursor(dsp, XC dotbox);
15:
        gxScale
16:
        gxMove
                    = XCreateFontCursor(dsp, XC_hand1);
17:
        gxNormal
                   = XCreateFontCursor(dsp, XC top left arrow);
18:
        rubberGC = XCreateGC(XtDisplay(GxDrawArea),
19:
20:
                   DefaultRootWindow(XtDisplay(GxDrawArea)), 0, NULL );
21:
        XSetForeground( XtDisplay(GxDrawArea), rubberGC, color );
22:
23:
        XSetFunction( XtDisplay(GxDrawArea), rubberGC, GXxor );
24:
25:
        XSetWindowBackgroundPixmap(XtDisplay(GxDrawArea),
26:
                                   XtWindow(GxDrawArea), GxDrawAreaBG);
27:
28:
        FixedX = FixedY = OrigX = OrigY = ExntX = ExntY = 0;
29:
      }
```

The function starts by extracting the background color of the status label for use later in the function when the global rubberGC is defined:

6: XtVaGetValues(GxStatusBar, XtNbackground, &color, NULL);

The widget specified for determining the background color of the application is arbitrary.

The function continues by ensuring that the head of the linked list used to manage the objects known to the editor and the pointer to the currently active object are set to NULL:

```
8: gxObjHeader = NULL;
9: gxObjCurrent = NULL;
```

Following this, the cursor values used to reflect the state of the application are created:

11: gxCrosshair = XCreateFontCursor(dsp, XC_crosshair); 12: gxPencil = XCreateFontCursor(dsp, XC_pencil);

13:	gxEdit	=	XCreateFontCursor(dsp,	XC_cross);
14:	gxTextI	=	XCreateFontCursor(dsp,	<pre>XC_xterm);</pre>
15:	gxScale	=	XCreateFontCursor(dsp,	<pre>XC_dotbox);</pre>
16:	gxMove	=	XCreateFontCursor(dsp,	XC_hand1);
17:	gxNormal	=	XCreateFontCursor(dsp,	XC top left arrow)

To provide the compiler with the definition of the cursors created in lines 11–17, the header file cursorfonts.h must be added to the beginning of gxGraphics.c source file:

#include <X11/Xaw/Box.h>
#include <X11/cursorfont.h>

Following the creation of these cursors, they are available for use during execution of the application.

The initializeGX function next creates the global rubberGC used by the object creation routines to provide a rubber-banding effect during object creation and update routines:

```
19: rubberGC = XCreateGC(XtDisplay(GxDrawArea),
20: DefaultRootWindow(XtDisplay(GxDrawArea)), 0, NULL );
```

The color extracted from the status label when the function began is assigned as the foreground color of the rubberGC Graphics Context

```
22: XSetForeground( XtDisplay(GxDrawArea), rubberGC, color );
```

and the function of the Graphics Context is set to the GXxor value, which enables the rubber-banding behavior and ensures that items drawn using this GC do not affect the drawing area permanently:

```
23: XSetFunction( XtDisplay(GxDrawArea), rubberGC, GXxor );
```

Finally, several global variables used to manage the object's scale, move, and rotate functions are initialized:

28: FixedX = FixedY = OrigX = OrigY = ExntX = ExntY = 0;

Although use of these variables is not introduced until Chapter 20, their definitions can be added to the GLOBAL section of gxGraphics.h

```
GLOBAL Widget GxStatusBar;
GLOBAL Widget GxDrawArea;
GLOBAL Pixmap GxDrawAreaBG;
GLOBAL GC rubberGC;
GLOBAL int GxActiveHandle;
GLOBAL GXObjPtr gxObjHeader;
GLOBAL GXObjPtr gxObjCurrent;
```

GLOBAL	int	FixedX,	FixedY;
GLOBAL	int	OrigX,	OrigY;
GLOBAL	int	ExntX,	ExntY;

and they will be ready for use when the functions, which require them, are introduced.

Before completing the introduction of the initializeGX function, it is important that a call to it is added at the appropriate phase of the application startup. Listing 17.10 shows the proper placement for a call to initializeGX function.

Listing 17.10 Changes Necessary to the Function Main in gxMain.c for Invoking the Function initalizeGX

```
1: .
2: .
3: .
4: XtRealizeWidget( toplevel );
5: initializeGX();
6:
7: XtAppMainLoop( appContext );
```

This completes a discussion of tools and utilities used by the Graphics Editor application. These routines work to create the common portion of the editor objects, to add objects to the management list, to create GCs to reflect the attributes of an object (and optionally prepare that GC for erasing), as well as using the program's cursor to reflect the state of the editor.

Next Steps

In the next chapter, I will introduce the concept of file formats in preparation for constructing the Save and Restore functions used by the Graphics Editor.

Chapter 18

In this chapter

- Understanding Files
- Binary File Formatting
- ASCII File Formatting
- Magic Numbers
- Next Steps

File Formats

As the Graphics Editor project advances in capability, it is increasingly more important for users to be able to save and subsequently restore the graphic objects they work so hard to create, place, and modify during a session of the editor application.

This chapter introduces the concepts and principles governing file formatting as a segue to adding the Save and Load functions used by the Graphics Editor.



The most important concept to draw from this chapter is that the programmer who authors the application that will employ a file decides the format of the data placed in that file.

This can seem a hasty absolute as I point out that many applications understand many file formats. However, as I introduce the concepts relating to file formats, consider first that the decision of how to format a file starts with a programmer.

Several considerations guide the decision of how a file is formatted, or how the data targeted for a file is arranged.

For example, one consideration is *size*, dictating whether compression should be used in the file format. If a large amount of information is intended for the file contents and these files are to be transmitted across a network, you should make the file as small as possible.

Another consideration is *speed*. Files of considerable size that are formatted in a plain text or *ASCII* format take longer to parse and load than files formatted using a *binary* method.

These considerations direct the decision of the programmer in determining how the data in a file will be organized.

Note As the save and load features of the Graphics Editor are implemented, be cognizant that only the Graphics Editor application will be able to read the files formatted for a task. By read I mean, of course, create graphic objects from the data contained in the file that can be moved, scaled, rotated, and so forth. However, as its popularity grows to worldwide renown, other graphic editors might want to adopt the file format. This probably won't occur for our application, but I need a humorous way to demonstrate that file formats are generally unique to the application programmer who conceives them. Clearly, the more commonplace and popular the application becomes, the more likely that other applications will adopt some level of support for their file format. But it is not automatic. In other words, not all graphic programs understand multiple graphic file formats, just as not all word processors understand documents formatted by other word processing applications.

For this reason, a unique suffix or file extension distinguishes the file type and the application meant to interpret it.

When all the decisions are made and the file format decided, it cannot be arbitrarily changed. Any changes to the placement of the data within the file or the characteristics of the data will have a direct and significant impact on the application that reads and writes it.

In other words, if a new data field is added to the file, the application's functions that write and read the file must be altered to support the new field.

Furthermore, a decision must be made as to whether the previous file format (before the addition of the data field) will still be supported.

If support for the original format is continued, the application's author must account programmatically for determining which format a file passed to the application is in and the action required to read it.

Emphasizing the impact that even a small change to the format of a file has on the application leads me back to the statement made earlier: The programmer who authors the application that will employ a file decides the format of the data placed in that file.

Acknowledging the impact on the application for which the file is targeted, consider for a moment the significance of a format change if that file format is understood by multiple applications. The following sections will address creating files using both ASCII and binary formatting techniques. First, however, let us be sure of what a file is.

Understanding Files

In data processing, a file is a collection of *records*. For example, all the information one might have on his customers would be ideal for storing in a file.

Each customer record would consist of *fields* for individual data elements. These data elements could include items such as the customer's name, identification number, address, and so forth.

By placing all information within a record in exactly the same location for all fields for each of the customer records, the organization of the file will be uniform and easily manipulated by a computer program.

This example can be instantly paralleled to the data requirements of the Graphics Editor project. Each object created in the editor will have common data fields, which must be included in the data set saved to the file created for the purpose. However, the task is complicated in that there are non-uniform data elements that also must be included in the file. These non-uniform elements are the data-specific fields of the various unique objects supported by the editor.

The last point to be made is the capability to distinguish file formats. Some applications describe files with given formats by assigning them a particular filename suffix. (The filename suffix is also known as a *filename extension*.)

For example, a program or executable file is sometimes given or required to have an .exe suffix. In general, the suffixes tend to be as descriptive of the formats as they can be within the limits of the number of characters allowed for suffixes by the operating system.

Binary File Formatting

Writing a file using a binary file format refers to writing the file using the binary numbering system, which uses only 0 and 1.

This is significant because the conventions used to examine the contents of a file (text editors, word processors, and so forth) cannot understand the file's contents. This is true because *character encoding* (meaning the arrangement of characters) is typically based on 8-bit groupings.

A binary data file, however, will not (necessarily) write 8 bits for each data element it places in the file. Consider the PCX file header format shown in Listing 18.1.

Listing 18.1	PCX File Header	r Format	
Byte Ø	Item S Manufacturer	ize 1	Description/Comments Constant Flag 10 = ZSoft .PCX1 Version 1 Version information: 0 = Version 2.5 2 = Version 2.8 w/palette information 3 = Version 2.8 w/o palette information 5 = Version 3.0
2	Encoding	1	1 = .PCX run length encoding
3	Bits per pixel	1	Number of bits/pixel per plane
4	Window	8	Picture Dimensions (Xmin, Ymin) - (Xmax - Ymax) in pixels, inclusive
12	HRes	2	Horizontal Resolution of creating device
14	VRes	2	Vertical Resolution of creating device
16	Colormap	48	Color palette setting, see text
64	Reserved	1	
65	NPlanes	1	Number of color planes
66	Bytes per Line	2	Number of bytes per scan line per color plane (always even for .PCX files)
68	Palette Info	2	How to interpret palette - 1 = color/BW, 2 = grayscale
70	Filler	58	blank to fill out 128 byte header

What is most interesting about Listing 18.1 is the SIZE column reflecting the size of the data field written for the data entities.

The first three fields written to a .pcx file are the Manufacturer, Encoding, and Bits per pixel fields necessary to interpret the image data contained in the file. Each of these data fields is only 1 bit in size. The fourth field placed in the file is the Window element and is 8 bits in size.

Therefore, if a .pcx file were loaded into a text editor, the first 3 bits (corresponding to the first three data elements) and the first 5 bits of the fourth data element would be interpreted by the editor as a character.

Depending on the values of these elements, the bits can map to a printable character. If not, the text editor will attempt to display a non-printable encoding of these 8 bits.

A discussion of the attempt to encode binary data into character data does not introduce a file format consideration, but rather draws a contrast in data organization between ASCII and binary formats.

What does serve as a consideration when choosing a binary file format is the complexity required to parse the file because of its cryptic format: not that it will necessarily be cryptic to the developer, but rather to those who adopt support of it later. An advantage of the binary file format is that the size of the file is minimized as the absolute required data elements (down to the single bit value) are written to the file. Generally, there is very little padding, as shown in the last field of the .pcx file header.

As a caution when choosing the binary file format, the mechanism used to create the file can be governed by the same limitations as binary object files introduced in Chapter 1, "UNIX for Developers," in the section "Object Files," page 19.

For example, if the common object data structure introduced in Chapter 15, "Common Object Definition," in the section "Common Object Data Structure," page 315, were saved to a file using a *dump* as illustrated here

```
fwrite( gxObj, sizeof( GXObj), 1, fp );
```

the data would depend on the byte ordering and word groupings of the machine architecture that performed the fwrite.

The data could only be successfully read on a machine of the same architecture.

The binary file formatting method is (generally) more efficient both in the size of the files it generates and in the speed at which the files are parsed.

Files saved using ASCII file formatting (as introduced in the next section) are not as efficient but are easily maintained.

ASCII File Formatting

ASCII file formats, as implied by the title, are based on ASCII and use only the printable characters from the ASCII table.

Contrasted to binary file formats for saving application data, ASCII files are inherently larger than necessary because the data is written using 8-bit encoding even when the data requirements are less.

Because the files are based on the ASCII table, text editors and word processors can read plain-text fields.

The following sections discuss two common methods of writing ASCII-based files: tagged and position-specific formatting.

Tagged File Formats

One method of providing extensibility and robustness to your application's data file is to employ a tagged file format.

Within a tagged file format, keywords are grouped closely to a value. Keyword value pairs can be one per line or separated using a token known not to exist in either any of the keyword specifications or possible values that will satisfy the keywords.

An example of a tagged file format is shown in Listing 18.2.

Listing 18.2 Sample Line-Separated Tagged File Format

```
1: FS=None
2: LS=Solid
3: LW=3
4: FG=Red
5: BG=None
```

Optionally, all tag value pairs could appear separated by a token on the same line, as shown in Listing 18.3.

Listing 18.3 Sample Token-Separated Tagged File Format

1: FS=None,LS=Solid,LW=3,FG=Red,BG=None

The flexibility afforded by tagged file formats is the capability to add and remove keys with minimal impact on the owning application.

In the absence of a keyword value pair, the application can assign a default value to the destination field.

Further, if an unrecognized keyword value pair is found in the file, it could simply be ignored, enabling new data elements to be added to the file with no impact on older versions of the application loading the new file format.

The tagged ASCII file format is clearly the slowest to parse and load because of the level of processing that must occur for the keyword value pairs found in the data file.

Not as robust as the tagged ASCII file but offering some level of convenience are the position-specific ASCII file formats introduced in the next section.

Position-Specific File Formats

Position-specific ASCII file formats are the easiest to create programmatically because the developer can create format strings that are used for both the write and read functions of the application.

To implement a position-specific file format, the data elements targeted for the application's data file are formatted using the syntax of the format string used by the C language fprintf function.

Listing 18.4 demonstrates this using the same data fields from previous listings.

Listing 18.4 Writing a Position-Specific ASCII Data File Entry

```
1: fprintf( fp, "%s %s %s %s \n", fs, ls, lw, fg, bg );
```

This same format string is subsequently used to read the data with the Cfscanf function, as shown in Listing 18.5.

```
Listing 18.5 Reading an Entry From a Position-Specific ASCII Data File
```

```
1: fscanf( fp, "%s %s %s %s \n", &fs, &ls, &lw, &fg, &bg );
```

The position-specific mechanism, though simple to implement, does not offer the resiliency or robustness of the tagged file method. A change to the file format introduces a complication that is not trivial to overcome. However, by placing a control tag or header element known as a *magic number* at the start of the file, the application will know the expected data and the file characteristics.

Magic Numbers

The general method of determining the data elements written to an application's data file is through use of a *magic number*, which serves as a control or version number for the application's restore function.

The best illustration of a magic number is its use in the graphic interchange file (GIF) format. This graphic image file uses two slightly different formats.

A string found at the very beginning of the file distinguishes the file format as either GIF87a or GIF89a.

Based on the value of the magic number, an application's load function can account for version differences corresponding to the format of the file being loaded.

The magic number, as illustrated with the strings GIF87a and GIF89, need not be an actual number. Literally, the term refers to a position-specific entry in the data file that can subsequently be used by the program to predict the elements contained in the file. As described with the Tagged Line format that uses a tag line to determine the data fields to follow, a magic number applies to the expectations of the contents of the entire file.

Next Steps

With a clearer understanding of the decisions and options you face when determining the best way to format a data file to meet the needs of the save-and-restore feature of an application, you are ready to face the challenge posed in the next chapter.

Chapter 19, "Save and Restore," satisfies the requirement of adding a save-andrestore capability to the Graphics Editor.

Chapter 19

In this chapter

- File Format Strategy
- Save and Restore Program Hooks
- Common-Object Save and Restore
- Object-Specific Save and Restore
- Next Steps

Save and Restore

The ability for users to continue their work in subsequent sessions of the Graphics Editor marks it as a mature and professional-level application.

Borrowing from the discussion of Binary, ASCII, and ASCII-tagged file formats of Chapter 18, "File Formats," the following sections satisfy the requirement of saving and restoring the graphics objects created by the Graphics Editor.

This chapter introduces a file format strategy and shows how the common and specific data elements of the editor objects will easily comply.

File Format Strategy

Familiar with the concepts of formatting application data files from Chapter 18, it is important that we identify a strategy for the Graphics Editor's generation of files that provides a blend of simplicity, resiliency, and robustness.

As was made clear from the editor object definitions introduced in Chapter 15, "Common Object Definition," not all the data elements that compose an editor object are homogenous. In other words, beyond the common elements controlling the attributes of the objects, each object type has unique data fields.

The Arc object retains the contents of the XArc structure to define data specific to the object:

```
typdef struct XArc {
    int x, y;
    Deminsion width, height;
    int angle1, angle2;
};
```

The Text object employs the GXText structure

```
typedef struct _gxtext {
    int x, y;    /* top-left */
    int dx, dy;
    char *text;
    int len;
    GXFont vpts;
    GXFont font;    /* segment definitions */
    GXFontP fontp;    /* num segs per char */
}
```

```
} GXText, *GXTextPtr;
```

and the standard point-array-based objects LatexLine, PolyLine, Box, and Arrow use the GXLine data structure:

```
typedef struct _gxline {
    XPoint *pts;
    int num_pts;
} GXLine, *GXLinePtr;
```

Because the data requirements for the objects of the Graphics Editor are not identical, the file format strategy must account for the nesting of data for objects of different types.

To accomplish this, a combination of the Position-Specific and Tagged-File Formatting methods is used. The merging of the two techniques enables the restore method of the editor to distinguish between object data for the various object types as well as enabling the editor objects to expand the fields they save in future versions without affecting existing data files.

Listing 19.1 shows the data file generated by procedures introduced later in this chapter when the objects shown in Figure 19.1 are saved. The data file includes an example of the point-array-based Line object, an Arc object, and the Text object.

Figure 19.1 A sample of the objects saved by the editor:

```
Listing 19.1 Graphics Editor Data File Format
```

```
1: OBJ - fg bg ls lw
2: 0 65535 0 1
3: LINE [numpts x y x y ...]
4: 9
5: 30 91
6: 38 84
7: 38 59
8: 25 59
9: 52 28
10: 79 59
11: 65 59
12: 65 84
13: 74 91
14: OBJ - fg bg ls lw
15: 0 65535 0 1
16: ARC [x y width height angle1 angle2]
17: 85 88 56 40 0 23040
18: OBJ - fg bg ls lw
20: 0 65535 0 1
21: TEXT [ str x y ]
22: irene
23: 148 131
```

The pattern of the file is to first save a tagged line indicating the format of the data that follows on the next line. For instance,

1: OBJ - fg bg ls lw

shows that the data to follow will be specific to the common object portion of an object and include the field's foreground, background, line style, and line width.

```
2: 0 65535 0 1
```

The second line is a position-specific formatted line containing the data fields promised by the previous tagged line.

Consider the tagged line produced by the Line object:

```
3: LINE [numpts x y x y ...]
```

This line indicates the form of the Line data that follows the tagged line. Specifically, saving first the number of points contained in the object

```
4: 9
```

followed by the appropriate pairs of points:

5: 30 91 6: 38 84 7: 38 59 8: 25 59

The same format is applied to the Arc object

```
14: OBJ - fg bg ls lw
15: 0 65535 0 1
16: ARC [x y width height angle1 angle2]
17: 85 88 56 40 0 23040
and the Text object:
```

21: TEXT [str x y] 22: irene 23: 148 131

Notice that the fields stored for the Text object are considerably less than those contained in the GXText structure. Only the fields required to reproduce the object when loaded into the editor are placed in the data file. For instance, because the editor supports only one vector text font, the font data need not be placed in the data file. If, however, in the future you expand the editor to enable the user to select from many vector fonts, the font assigned to the saved object must be included in the data file. Fortunately, the data file's format strategy accounts for content expansion.



The strength of combining position-specific and tagged-file format strategies is that the application can parse the tagged line to know how to interpret the positionspecific line.

In the future, when the application is advanced and functionality is added, the tagged line is updated to reflect the additional data field(s) saved to the file. The newer versions of the application will include the evaluation of the tagged line to determine whether an older file that doesn't contain the additional data or a newer file that does is being loaded.

The following sections show the necessary modifications to the Graphics Editor project to support the save and restore functionality using a combined position-specific and tagged-file format.

Save and Restore Program Hooks

The definition of the control menu for the Graphics Editor provides a point of entry for the save and restore functions:

```
static GxIconData gxCntrlIcons[] = {
    .
    .
    .
    { &save_icon, gx_save, "Save current drawing..." },
    { &load_icon, gx_load, "Load saved drawing..." },
    .
    .
    .
    { NULL },
};
```

The functions gx_save and gx_load, however, have heretofore been empty (stub) functions. The next section introduces their contents, but first, we must modify the Common Object Structure to include an object method for saving the object-specific data contained in each graphics object.

Add the following line to the GXObj data structure located in gxGraphics.h header file:

```
void (*move) ( struct _gx_obj *, XEvent * );
void (*scale) ( struct _gx_obj *, XEvent * );
void (*action) ( struct _gx_obj *, XEvent * );
void (*save) ( FILE *, struct _gx_obj * );
```

Further, this object method should be initialized in the function gx_create_obj located in the gxGx.c source file:

```
gx_obj->scale = (void (*)())null_func;
gx_obj->copy = (void (*)())null_func;
gx_obj->save = (void (*)())null_func;
```

With the proper object method defined and initialized, we can now look at the contents of the gx_save and gx_load functions.

Common-Object Save and Restore

When the user clicks the Save icon located in the Control button panel of the Graphics Editor application, the gx_save function is invoked.

This function is responsible for prompting the user for a filename, opening the file, and traversing all objects contained in the application, saving first their common data elements and then invoking the objects' save method to write the object-specific data to the file.

The contents of the gx_save function accomplishing the steps described are shown in Listing 19.2.

```
Listing 19.2 The gx_save Function
```

```
1: void gx save( void )
2:
    {
        FILE *fp;
3:
4:
5:
        char *filename = gxGetFileName();
6:
7:
        setStatus( "Saving objects..." );
8:
        if( filename ) {
9:
            fp = fopen( filename, "w+" );
10:
11:
            if( fp == NULL ) {
              perror( "Failed to open file: " );
12:
13:
            } else {
                gxSaveObjs( fp, gxObjHeader );
14:
15.
                fclose( fp );
16:
17:
           }
18:
        }
        setStatus( "Saving objects... done!" );
19:
20: }
```

The gx_save function begins by invoking a function to prompt the user for a filename:

```
5: char *filename = gxGetFileName();
```

When the filename is returned, a file of the specified name is opened for writing:

9: fp = fopen(filename, "w+");

After ensuring that the file opened successfully in lines 11–13, the function gxSaveObjs is called

14: gxSaveObjs(fp, gxObjHeader);

specifying the beginning of the list of objects and a pointer for the opened data file.

Listing 19.3 shows the gxGetFileName function and Listing 19.4 shows the gxSaveObjs function.

EXCURSION

Error Reporting in the C Programming Language

Notice the perror command, used for the first time in this text in Listing 19.2.

12: perror("Failed to open file: ");

The C programming language provides the perror function to print to stderr the programmer's specified message ("Failed to open file: ") followed by the error reported by the system when the offending call failed. System errors such as failing to open a file are made available to routines in C by merit of a global variable named errno.

The value of errno is set when an error occurs, and its value corresponds to an array of error strings maintained by C. For instance, if you wanted the Graphics Editor to report the error to the status bar, replace the perror call with the following:

```
{
    char msg[128];
    sprintf( msg, "Failed to open file %s : %s\n",
        filename, strerror( errno ) );
    setStatus( msg );
}
```

The strerror function, also provided by C, will find the error string corresponding to the value of errno. In order for these functions to be prototyped for use and errno to be visible in the source file, you must include the error.h header file and extern the errno variable. This is accomplished with the following lines:

```
#include <error.h>
extern in errno;
```

Listing 19.3 Retrieving a Filename from User Input

```
static char *gxGetFileName( void )
1:
2: {
3:
      XtAppContext app;
4:
     XEvent
                   event;
5:
6:
     Widget dialog;
7:
      char *str = NULL;
8:
      dialog = XtVaCreateManagedWidget( "Filename...",
9:
10:
                                         dialogWidgetClass,
11:
                                         GxDrawArea,
12:
                                         XtNwidth, 115,
13:
                                         XtNheight, 70,
                                                   "Enter File:",
14:
                                         XtNlabel,
                                         XtNvalue, "",
15:
16:
                                         NULL );
17:
      XawDialogAddButton( dialog, " Ok ", close_dialog, dialog );
18:
19:
20:
      app = XtWidgetToApplicationContext( GxDrawArea );
21:
     while( XtIsManaged(dialog)) {
22:
23:
        XtAppNextEvent( app, &event );
24:
        XtDispatchEvent( &event );
```

19

Listing 19.3 Continued

```
25:
      }
26:
27:
      str = XawDialogGetValueString( dialog );
28:
      XtDestroyWidget( dialog );
29:
30:
      /*
31:
       * look for 'illegal' characters
32:
       */
33:
      {
34:
        int c, indx = 0;
35:
        char illegal_chars[] = { '\n', 'z' };
36:
37:
        while( (c = (int)str[indx]) != ' \setminus 0' ) {
38:
          if( strchr( illegal chars, c ) != NULL ) {
            str[indx] = '\0';
39:
40:
            break;
41:
          }
42:
          indx++;
43:
        }
44:
45:
        /*
46:
         * remove leading zeros
47:
         */
        while( *str && *str == ' ' ) str++;
48:
49:
      }
50:
51:
      if( str && *str )
52:
        return XtNewString(str);
53:
      else
54:
        return NULL;
55: }
56: static void close_dialog( Widget w, XtPointer cdata,
57:
                                XtPointer cbs )
58: {
59:
      Widget dialog = (Widget)cdata;
60:
61:
      if( dialog ) XtUnmanageChild( dialog );
62: }
```

The gxGetFileName function that is found in Listing 19.3 creates a dialog widget with the single OK button (lines 9–18). An illustration of dialogWidget created by this function is found in Figure 19.2.



Inclusion of the header files X11/Xaw/Dialog.h and X11/Xaw/StringDefs.h as well as a prototype for close_dialog must appear at the beginning of the file where you place the functions introduced in Listing 19.2 in order for the sample to compile without error.



The function then extracts the XtApplicationContext and uses it to intercept events from the XtAppMainLoop by extracting and dispatching events until the user closes the dialog prompting for input:

```
20: app = XtWidgetToApplicationContext( GxDrawArea );
21:
22: while( XtIsManaged(dialog)) {
23: XtAppNextEvent( app, &event );
24: XtDispatchEvent( &event );
25: }
```

A widget is considered managed as long as it is visible on the screen. When the user presses the OK button, the dialog is removed from the screen by the close_dialog function registered as the OK button's callback action

```
18: XawDialogAddButton( dialog, " Ok ", close_dialog, dialog );
```

and the test XtIsManaged returns False, ending the while loop.

The close_dialog function simply "unmanages" the dialogWidget when the button is pressed:

```
61: if( dialog ) XtUnmanageChild( dialog );
```

The value entered into the text field inherent to the dialogWidget is consulted for the user's input

```
27: str = XawDialogGetValueString( dialog );
```

and the widget is destroyed:

```
28: XtDestroyWidget( dialog );
```

The remainder of the gxGetFileName function ensures the integrity of the string for use as a filename. Specifically, a filename cannot contain newlines, spaces, or, solely for the purpose of illustration, lowercase z characters.

When the function is sure of the contents of the string, a copy is returned to the user:

51: if(str && *str)
52: return XtNewString(str);

Otherwise, if the removal of illegal characters has consumed the string, NULL is returned:

53: else 54: return NULL;

The gxSaveObjs routine introduced in Listing 19.4 satisfies the final function introduced in Listing 19.2

14: gxSaveObjs(fp, gxObjHeader);

for accomplishing the saving of objects created by the editor application.

Listing 19.4 Traversing the Editor Objects for Saving

```
static void gxSaveCommon( FILE *fp, GXObjPtr obj )
1:
2:
   {
3:
       fprintf( fp, "OBJ - fg bg ls lw\n" );
4:
       fprintf( fp, " %ld %ld %d %d\n",
5:
                obj->fg, obj->bg, obj->ls, obj->lw );
6:
   }
7:
8: static void gxSaveObjs( FILE *fp, GXObjPtr obj )
9: {
10:
        if( obj ) {
            gxSaveCommon( fp, obj );
11:
12:
            (*obj->save)( fp, obj );
13:
14:
            gxSaveObjs( fp, obj->next );
        }
15:
16: }
```

The gxSaveObjs function in Listing 19.4 is a recursive function called repetitively until the end of the list of objects is found. The two lines

10: if(obj) {

and

14: gxSaveObjs(fp, obj->next);

are both important to the recursive nature of the function. Line 10 ensures that a valid object reference was provided as the end of the list of objects is marked with NULL, whereas line 14 passes the next possible object.

For each object contained in the list, two functions are invoked. The first is to save the common object data fields

11: gxSaveCommon(fp, obj);

and the second is to save the object-specific data:

12: (*obj->save)(fp, obj);

The gxSaveCommon found simply writes the tagged line and the position-specific data line to the file referenced by the file pointer:

3: fprintf(fp, "OBJ - fg bg ls lw\n"); 4: fprintf(fp, " %ld %ld %d %d\n", 5: obj->fg, obj->bg, obj->ls, obj->lw);

The graphic objects' save methods are introduced in Chapters 20–24 with the definition of the object internals.

The following section demonstrates how the data file generated by the save action of the Graphics Editor is restored in the editor.

Object-Specific Save and Restore

When the user clicks the Load icon located in the control button panel of the Graphics Editor application, the gx_load function is invoked.

This function is responsible for prompting the user for a filename, opening the file, traversing the file contents, and creating an object specific to the tags located in the file.

Listing 19.5 shows the contents of the gx_load function.

Listing 19.5 The gx_load Function

```
1: void gx_load( void )
2: {
        FILE *fp;
3:
4:
5:
        char *filename = gxGetFileName();
6:
7:
        setStatus( "Loading objects..." );
8:
        if( filename ) {
          fp = fopen( filename, "r" );
9:
10:
11:
          if( fp == NULL ) {
            perror( "Failed to open file: " );
12:
13:
          } else {
              gxLoadObjs( fp );
14:
15:
              fclose( fp );
16:
          }
17:
        }
18:
        setStatus( "Loading object... done!" );
19: }
```

After prompting for and validating the filename, the file is opened read-only:

8: if(filename) {
9: fp = fopen(filename, "r");

19
If the file is opened successfully, the function gxLoadObjs is called with the file reference:

14: gxLoadObjs(fp);

Listing 19.6 introduces the gxLoadObjs function, which traverses the file referenced by fp and invokes the necessary create functions to restore the objects to the editor.

```
Listing 19.6 Loading Objects into the Editor
```

```
static void gxLoadCommon( FILE *fp, GXObjPtr obj )
1:
2:
    {
3:
       fscanf( fp, " %ld %ld %d %d\n",
4:
                &obj->fg, &obj->bg, &obj->ls, &obj->lw );
5:
   }
6:
7: static void gxLoadObjs( FILE *fp )
8: {
9:
        char objForm[128];
10:
        GXObjPtr obj;
11:
12:
        while( fgets( objForm, 128, fp ) != NULL ) {
13:
            obj = gx create obj();
14:
            gxLoadCommon( fp, obj );
15:
16:
            fgets( objForm, 128, fp );
17:
            switch( *objForm ) {
              case 'A':
18:
19:
                gxArcLoad( fp, obj );
20:
                break;
21:
22:
              case 'L':
23:
                gxLineLoad( fp, obj );
24:
                break;
25:
26:
              case 'T':
27:
                gxTextLoad( fp, obj );
28:
                break;
29:
            }
30:
        }
31: }
```

The gxLoad0bjs function retrieves a line from the file reference by the file pointer fp

12: while(fgets(objForm, 128, fp) != NULL) {

using the C fgets function, which fills the buffer objForm up to the occurrence of a newline character in the file.



Knowing the format of the file, you are aware that this line contains the tagged field line specifying the data elements provided on the following line.

Because this is version 1.0, there is no need to parse the tagged line to determine the fields specified. If, however, in future versions of the application, data fields are added to the save format, this is when the line would be parsed.

Following the extraction of the tagged line, the data for the object-specific values is read:

14: gxLoadCommon(fp, obj);

Notice that in the gxLoadCommon function, the syntax for reading the data written by gxSaveCommon is very similar. Specifically, the fwritef has been renamed fscanf and addresses of the structure elements are provided so that the values read from the file can be assigned to them:

With the common object elements restored, the next line of the file is the tagged line of the object-specific data. By reading this line into the objForm array

16: fgets(objForm, 128, fp);

you can determine which load routine to call because the first character is sufficient for determining the object type:

17: switch(*objForm) {



The syntax *objForm is identical to specifying objForm[0]. Both references equate to the first character of the array.

Based on the value of the first character, one of the methods gxArcLoad, gxLineLoad, or gxTextLoad is invoked to read the data specific to an object of that type.

These functions are introduced in the following chapters when (*finally* you must be saying) the internals of the various editor objects are introduced.

Next Steps

The addition of functionality to support the save and restore capability of the editor is necessary to produce a mature product. Relative to the flow of this text, this chapter completes the evolution of the graphics editor objects and enables us to address the important task of defining each object in full.

The following five chapters in Part V, "Adding Objects to the Editor," provide the definition of internal object methods and functions required to create a functional graphics editor.



Adding Objects to the Editor

Chapter 20

In this chapter

- Creating a Latex Line Object
- Drawing and Erasing a Line Object
- Finding a Line Object
- Selecting and Deselecting a Line Object
- Moving a Line Object
- Scaling a Line Object
- Copying a Line Object
- Saving and Restoring a Line Object
- Next Steps

Latex Line Object

By choosing the icon shown in Figure 20.1, a user begins the Latex Line object creation process.

Figure 20.1



Choose the Latex Line creation icon to begin the Latex Line object creation process.

During the creation of a Latex Line object, points are selected from the canvas to define the vertices of the object. There is no limit to the number of points or the direction of the segments created during this process.

When all the points are defined for the new Latex Line object, the user doubleclicks the mouse cursor to indicate the end of the creation. Upon completing the interactive mode of the Latex Line object creation, a plethora of events occur.

This chapter presents the internal methods and supporting functions defined for the Latex Line object. These routines control all aspects of managing the different features of the object.



The code listings introduced in this chapter are targeted for placement in the gxLine.c source file. Additionally, functions presented in the listings that are not defined static should have a corresponding prototype placed in the gxProtos.h header file.

Creating a Latex Line Object

The definition of the gxDrawIcons array introduced in Chapter 13, "Application Structure"

```
{ &line_icon, (void (*)(void))gx_line,
   "Draw an elastic line..." },
```

assigned the Latex Line icon the gx_line function for invocation when selected by the user. This function is defined in Listing 20.1, and it serves as the entry point to the creation process of the Latex Line object.

```
Listing 20.1 The gx_line Function
```

```
1: void gx_line( XEvent *event )
2:
   {
      static GXLine *rubber line = NULL;
3:
4:
5:
      if( event == NULL ) {
6:
        rubber line = NULL;
7:
      } else {
8:
        /* remove the current rubber line (if there is one) */
9:
        GXRubberLine( rubber line );
10:
11:
        switch( event->type ) {
12:
          case ButtonPress:
13:
            if( rubber line == NULL ) set cursor( LINE MODE );
14:
            /*
15:
             * See if we 'double clicked' & are done selecting points
16:
             */
17:
            if( point_equal_event( rubber_line, event ) == True ) {
18:
19:
20:
              /* erase our temp line */
21:
              XDrawLines( XtDisplay(GxDrawArea), XtWindow(GxDrawArea),
                           rubberGC, rubber line->pts,
22:
23:
                           rubber line->num pts, CoordModeOrigin );
24:
25:
              create_line( NULL, rubber_line );
              gx refresh();
26:
27:
28:
              set cursor( NORMAL MODE );
              rubber line = NULL;
29:
            } else {
30:
31:
              /*
               * Initialize a GXLine struture to manage our creation
32:
               */
33:
34:
              GXRubberLine( rubber_line );
35:
              rubber_line = update_line( event, rubber_line );
            }
36:
          break;
37:
38:
```

```
39:
          case ButtonRelease:
40:
          case MotionNotify:
41:
            /*
42:
             * update the GXLine structure based on the
43:
             * new point and current location of the mouse
44:
             */
45:
            if( rubber line ) {
46:
              /*
47:
                * replace the last point with the current event location
48:
                * IF we have more than one point
               */
49:
              if( rubber line->num pts > 1 ) {
50:
51:
52:
                 rubber line->pts[rubber line->num pts-1].x =
53:
                                              event->xbutton.x;
54:
                 rubber_line->pts[rubber_line->num_pts-1].y =
55:
                                              event->xbutton.y;
56:
57:
              } else {
58:
                 (void)update line( event, rubber line);
59:
              }
60:
              /*
61:
               * redraw the rubberbanding line
62:
               */
63:
              GXRubberLine( rubber line );
64:
            }
65:
          break;
66:
        }
67:
      }
68: }
```

In reviewing the definition of the gx_line function, you instantly see that the function is entirely event driven—meaning, based on the event specified, the gx_line function takes an appropriate action.

The actions for each of the events anticipated can be generalized as follows:

- Upon receiving a ButtonPress event, begin the creation process.
- Subsequent ButtonPress events cause the addition of the event point to an array of points contained in the rubber_line structure.
- A ButtonPress event point that equals the previous point inserted into the rubber_line structure indicates a double-click and ends the creation.
- At the occurrence of either the ButtonRelease or MotionNotify event, the event point is connected to the most recent point added to the point array and drawn using a rubber-banding line to provide the interactive creation of the Latex Line.

The gx_line function begins by testing the validity of the event reference specified as the sole parameter to the function:

```
5: if( event == NULL ) {
6:     rubber_line = NULL;
7: } else {
```

Passing a NULL event pointer to the gx_line function enables the local variable rubber_line to be reset, thereby aborting the current creation mode. This is necessary to eliminate the possibility of the user failing to end one creation process before starting another. When a new creation process is started for any type of object, a NULL event is passed first to ensure the initial state.

When, however, a valid event pointer is specified, the rubber_line variable is passed to the GXRubberLine function, which is responsible for updating the line using the interactive rubber-banding GC as follows:

```
9: GXRubberLine( rubber_line );
```

Then the event type is checked to determine which action body the function should execute:

```
11: switch( event->type ) {
```

If the current event is a ButtonPress event, many things can happen because the action body associated with this event is the busiest. If the rubber_line variable is still NULL, the action body knows that this is the first iteration and sets the application cursor to reflect the new mode:

13: if(rubber_line == NULL) set_cursor(LINE_MODE);

IINote

As you might recall, the cursor control functions were introduced in Chapter 17, "Utilities and Tools," in "Using the Cursor as a State Indicator," page 349.

When a ButtonPress event occurs, the gx_line action for it must determine whether this event's location coincides with the pervious ButtonPress event point passed to the function:

```
18: if( point_equal_event( rubber_line, event ) == True ) {
```

If the point_equal_event function indicates that the points are equal, the creation process ends by erasing the interactive rubber-banding line and calling the create_line function (lines 21-25).

The ButtonPress case then returns the cursor to normal and clears the rubber_line variable:

```
28: set_cursor( NORMAL_MODE );
29: rubber_line = NULL;
```

However, if the point_equal_ function returned False, indicating that the current event point does not equal the previously added point (no double-click), the else body is entered, the interactive line is updated

34: GXRubberLine(rubber_line);

and the current event point is added to the rubber_line->pts array:

```
35: rubber_line = update_line( event, rubber_line );
```

The processing of the ButtonRelease or MotionNotify events is a little simpler:

```
39: case ButtonRelease:
40: case MotionNotify:
```

If a valid rubber_line structure reference exists, replace the last point in the array with the current event point:

52:	<pre>rubber_line->pts[rubber_line->num_pts-1].x =</pre>
53:	event->xbutton.x;
54:	<pre>rubber_line->pts[rubber_line->num_pts-1].y =</pre>
55:	event->xbutton.y;

If the rubber_line->pts array contains only one point, the array must be expanded to accommodate the point addition:

58: (void)update_line(event, rubber_line);

Note that the last point added to the pts array by the ButtonRelease or MotionNotify actions is not committed to the list of vertices that are specified by the user. Instead, this last point simply makes it easier for the GXRubberLine to do its job because it only must update the last segment defined by the array. This segment consists of one point selected by the user (second to the last point of the array) and one point stored temporarily by ButtonRelease or MotionNotify (last point). In other words, one more point is maintained in the rubber_line->pts array than requested by the user. This last point stored by the ButtonRelease and MotionNotify is replaced with the next ButtonPress event.

The last thing the gx_line function is responsible for is ensuring that the interactive rubber_line is again updated to the screen:

63: GXRubberLine(rubber_line);

The GXRubberLine function defined in Listing 20.2 updates segments of the line being created, providing a visual interactive creation for the user by enabling him to see where the next segment endpoint will be placed when the mouse button is pressed.

```
Listing 20.2 The GXRubberLine Function
```

```
static void GXRubberLine( GXLine *line )
1:
2: {
3:
      int indx;
4:
     if( line && line->pts && (line->num pts > 1)) {
5:
        indx = line->num_pts - 1;
6:
        XDrawLine( XtDisplay(GxDrawArea),
7:
                   XtWindow(GxDrawArea), rubberGC,
8:
                   line->pts[indx ].x,
9:
                   line->pts[indx ].y,
10:
                   line->pts[indx-1].x,
11:
                   line->pts[indx-1].y );
12:
      }
13: }
```

The GXRubberLine function begins by ensuring enough points are defined to form a line segment

```
4: if( line && line->pts && (line->num_pts > 1)) {
```

and then it assigns the variable indx the value of one less than the number of points, which is the last point of the array:

```
5: indx = line->num_pts - 1;
```



Remember from our discussions of arrays in Chapter 2, "Programming Constructs," that an array with two elements employs the indices 0 and 1 to access them.

Then the last segment of the line defined by the pts array is drawn using indx (the last point) and indx - 1 (the second to the last point):

6:	XDrawLine(XtDisplay(GxDrawArea),
7:		XtWindow(GxDrawArea), rubberGC,
8:		line->pts[indx].x,
9:		line->pts[indx].y,
10:		line->pts[indx-1].x,
11:		line->pts[indx-1].y);

Look now at the definition of the point_equal_point function defined in Listing 20.3, which is used by the gx_line function to determine whether the user doubleclicked the mouse cursor.

Listing 20.3 The point_equal_point Function

```
1: static Boolean point_equal_event( GXLine *line, XEvent *event )
2: {
3: Boolean pts_equal = False;
4: int xe_x, xe_y;
```

```
5:
6:
      xe_x = event->xbutton.x;
7:
      xe y = event->xbutton.y;
8:
9:
      /*
10:
       * the last point will always be the current motion event
11:
       * so check the one before for redundancy (equates to a
12:
       * double click to end the action )
13:
       */
14:
      if( line && (line->num pts > 2) ) {
15:
        int num = line->num pts - 2;
16:
17:
        if( (abs(line->pts[num].x - xe_x) <= TOLERANCE ) &&
            (abs(line->pts[num].y - xe y) <= TOLERANCE )) {</pre>
18:
19:
20:
         pts equal = True;
21:
       }
22:
      }
23:
      return pts_equal;
24: }
```

The point_equal_point function must ensure that there are at least three points in the pts array before determining whether a double-click occurred:

```
14: if( line && (line->num_pts > 2) ) {
```

This is necessary because the ButtonRelease or MotionNotify and not the user placed the last point of the array. In order for the line creation to be valid, at least two user-specified points must be in the array.

When the correct number of points exists in the array, the last user-defined point is compared to the event point. If the absolute value of their differences is less than the TOLERANCE (3) defined in the gxGraphics.h header file, they are considered equal.



The margin of error allowed by TOLERANCE accounts for the user having Monday morning shakes.

Furthermore, even on a good day, it is difficult to click the exact same point on the screen consecutive times. It is expected that the mouse can move ever so slightly from the subtle movement of hand muscles required to double-click the mouse.

If the points are deemed equal (or close enough), the interactive portion of the creation process ends and an actual line object is created with a call to the create_line function found in Listing 20.4.

```
Listing 20.4 The create_line Function
```

```
static void create line( GXObjPtr obj, GXLine *line )
1:
2:
   {
3:
      GXLinePtr line data;
4:
      GXObjPtr obj = _obj;
5:
6:
      if( obj == NULL ) {
7:
          obj = gx_create_obj();
8:
      }
9:
10:
      line data = (GXLinePtr)XtNew(GXLine);
11:
      memcpy( (char *)line_data, (char *)line, sizeof(GXLine));
12:
      obj->data = line data;
13:
14:
15:
      obj->draw
                   = line draw;
16:
      obj->erase = line erase;
17:
     obj->find
                 = line_find;
18:
     obj->move
                   = line move;
19:
     obj->scale = line_scale;
20:
     obj->copy
                = line_copy;
      obj->select = line select;
21:
22:
      obj->deselect = line_deselect;
23:
24:
      obj->save = line save;
25:
26:
      gx_add_obj( obj );
27: }
```

The parameter list of the create_line function

1: static void create_line(GXObjPtr _obj, GXLine *line)

enables an invocation of the procedure with the caller having already created the common object to contain the line data. This supports the restoring of the object from a saved file as is seen later in the chapter.

During interactive creation, however, the function create_line is called with a NULL value as the first parameter forcing the function to create the common object portion through a call to gx_create_obj introduced in Chapter 17, "Utilities and Tools," section "Common Object Creation," page 343.

```
7: obj = gx_create_obj();
```

A GXLine data structure is then created

```
10: line_data = (GXLinePtr)XtNew(GXLine);
```

and the contents of the second argument to the function is copied into the new structure:

11: memcpy((char *)line_data, (char *)line, sizeof(GXLine));

Finally, the new data structure is assigned as the unique data value for the new object:

13: obj->data = line_data;

Consistent with objects of type Line, the methods for the new object are assigned the line manipulation functions for controlling the object-specific data structure in lines 15–24.

Finally, the create_line function must ensure the newly created object is retained by the application by adding the object to the linked list used to manage editor objects:

```
26: gx_add_obj( obj );
```



To review the gx_add_obj function, refer to Chapter 17, section "Linked List Management," page 346.

The final function for review is the update_line function found in Listing 20.5, which is employed by gx_line to commit user selected points to the rubber_line->pts array.

Listing 20.5 The update_line Function

```
1: static GXLinePtr update line( XEvent *event, GXLine *line )
2: {
3:
      static GXLine xline;
      GXLinePtr xlinePtr = &xline;
4:
5:
6:
     if( line == NULL ) {
7:
        xline.pts = (XPoint *)XtMalloc( sizeof(XPoint) );
8:
       xline.num pts = 0;
9:
      } else {
       xline.pts = (XPoint *)XtRealloc( (char *)xline.pts,
10:
                             sizeof(XPoint) * (xline.num_pts + 1));
11:
12:
      }
13:
14:
      xline.pts[xline.num_pts].x = event->xbutton.x;
15:
      xline.pts[xline.num pts].y = event->xbutton.y;
16:
17:
      xline.num pts++;
18:
19:
      return xlinePtr;
20: }
```

The update_line function must manage the memory associated with the points array, holding the vertices added to the Line object by the user. If the line pointer is NULL, an XtMalloc is called to assign the initial storage space for the pts array

20

7: xline.pts = (XPoint *)XtMalloc(sizeof(XPoint));

and the number of points tracked by num_pts is initialized to 0

```
8: xline.num_pts = 0;
```

otherwise, memory previously assigned to the xline GXLine structure referenced is re-allocated to include room for the point being added:

```
10: xline.pts = (XPoint *)XtRealloc( (char *)xline.pts,
11: sizeof(XPoint) * (xline.num_pts + 1));
```

After sufficient room is obtained to contain the new point, it is added to the array

```
14: xline.pts[xline.num_pts].x = event->xbutton.x;
15: xline.pts[xline.num pts].y = event->xbutton.y;
```

and the total number of points is incremented to reflected the entry:

```
17: xline.num_pts++;
```

As the user interactively creates the Latex Line, points are added to the rubber_line->pts array and the number of points tracked in the num_pts field. At some point the user will double-click the mouse cursor, which is caught by the point_equal_point function, and the Line object will be created by the call to create_line. A created object is no longer drawn using the interactive rubber-banding GC, but by invoking the draw method of the object.

The following section introduces the methods that draw and erase the Line object from the drawing area canvas.

Drawing and Erasing a Line Object

The act of drawing or erasing an object in the Graphics Editor differs only in the treatment of the tile field of the GC created for the request.

Notice in Listing 20.6, lines 18 and 23, that the draw and erase methods differ only in the value passed for the tile flag required as the second parameter to the draw_erase function. (Note as well that the erase method is responsible for removing the object's handles if present on the screen.)

Listing 20.6 Drawing and Erasing a Line Object

```
1: static void draw_erase( GXObjPtr line, Boolean tile )
2: {
3: GC gc;
4: GXLinePtr line_data = (GXLinePtr)line->data;
5:
6: gc = gx_allocate_gc( line, tile );
7:
```

```
8:
      XDrawLines(XtDisplay(GxDrawArea),
9:
                XtWindow(GxDrawArea), gc,
10:
                line data->pts, line data->num pts,
11:
                CoordModeOrigin );
12:
13:
      XtReleaseGC( GxDrawArea, gc );
14: }
15:
16: static void line draw( GXObjPtr line )
17: {
18:
      draw erase( line, False );
19: }
20:
21: static void line erase( GXObjPtr line )
22: {
      draw_erase( line, True );
23:
24: }
```

The draw_erase function begins by extracting the GXLine data structure from the common object's data field

```
4: GXLinePtr line_data = (GXLinePtr)line->data;
```

and creating a GC from the attribute setting of the object:

```
6: gc = gx_allocate_gc( line, tile );
```

Important to the creation of the Graphic Context is the specification of the tile flag to the gx_allocate_gc function introduced in Chapter 17, section "Creating a Graphics Context," page 347.



If you recall, this flag indicated whether the background Pixmap of the GxDrawArea was assigned as the value to the tile field of the GC created.

A tile value assigned to a Graphic Context causes an effective erase action to occur because the pixels from the background Pixmap are placed where otherwise the foreground value of the GC would be placed, making the underlying background the result of the XDrawLines (or any X Graphic Primitive) request.

An appropriately created GC for the current draw or erase action, the draw_erase function can request the object be updated in the canvas window:

```
8: XDrawLines(XtDisplay(GxDrawArea),
9: XtWindow(GxDrawArea), gc,
10: line_data->pts, line_data->num_pts,
11: CapadMadaOpigin );
```

11: CoordModeOrigin);

Because the X Server will attempt to cache the GC for future requests, it is important to specify the complete use of it with a call to XtReleaseGC:

13: XtReleaseGC(GxDrawArea, gc);

With the object visible on the screen, it is now eligible for manipulation by the user. However, before it can be moved, scaled, or deleted, it must be selected by the user.

The next section introduces the Line object's find method used to determine whether an event has successfully located the object on the drawing area.

Finding a Line Object

The Line object's find method, shown in Listing 20.7, considers the object selected if the event point contained in the XEvent structure referenced in the function's second parameter coincides with one of the object's vertices or intersects one of the object's line segments.

Listing 20.7 The line_find Function

```
1: static Boolean line find(GXObjPtr line, XEvent *event)
2: {
3:
      Boolean found = False;
4:
     XPoint p;
5:
6:
      p.x = event->xbutton.x;
7:
      p.y = event->xbutton.y;
8:
9:
10:
       found = point selected( (GXLinePtr)line->data, &p );
11:
       if( found == False )
12:
         found = segment selected((GXLinePtr)line->data,&p);
13:
14:
       return found;
15: }
```

The line_find function extracts the event point from the XEvent structure

```
6: p.x = event->xbutton.x;
7: p.y = event->xbutton.y;
```

and tests first the vertices of the object

10: found = point_selected((GXLinePtr)line->data, &p);

and then the intersection of the point with the segments forming the object:

12: found = segment_selected((GXLinePtr)line->data,&p);

If a point is determined selected by the point_selected function, the work is done. Otherwise, the segment_selected is invoked and the resulting value found is returned to the caller.

Listing 20.8 introduces the segment_selected and point_selected functions in turn.

```
Listing 20.8 Determining Whether a Point or Segment Is Selected
```

```
1:
    static
2:
    Boolean segment selected( GXLinePtr data, XPoint *pt )
3:
    {
4:
      Boolean found = False;
5:
      int i;
6:
7:
      for(i=0; i<data->num pts-1 && found == False; i++) {
8:
        found = near_segment(data->pts[i].x,data->pts[i].y,
9:
                         data->pts[i+1].x, data->pts[i+1].y,
10:
                         pt->x, pt->y );
11:
       }
12:
       return found;
13: }
14:
15:
    static
     Boolean point selected( GXLinePtr line, XPoint *pt )
16:
17:
    {
18:
       int i, x, y, found = False;
19:
20:
       for( i = 0; (i < line->num pts) && !found; i++ ) {
21:
22:
         x = line->pts[i].x;
23:
         y = line->pts[i].y;
24.
         if (abs(pt->x - x) \le TOLERANCE \&\&
25:
26:
             abs(pt - y - y) \leq TOLERANCE ) {
27:
28:
           found = True;
29:
         }
30:
       }
31:
       return found;
32:
     }
```

The segment_selected function, found at the beginning of Listing 20.8, breaks the Line object into point pairs, defining each of the line segments that comprise the object. Each of these segments is passed to the near_segment function introduced in Chapter 10, "Trigonometric and Geometric Functions," in the section "Calculating Point and Line Intersection," page 211.

If for any segment the near_segment function returns True, the for loop ends and the value is returned to indicate that the object has been found.

The point_selected function, found in Listing 20.8, parses all the points defining the vertices (segment endpoints) of the object looking for a point within TOLERANCE of the event point.

If an appropriate point is found, True is returned to the line_find method, indicating that the current Line object has been successfully located by the event. The line_find method serves two purposes in the management of the Line object. One purpose of the line_find method is to enable the user to select the object for manipulation. A second purpose determines whether the manipulation requested by the user is the move action.

The follow section introduces the step stemming from the line_find method resulting in the selection (or implicit deselection) of the Line object.

Selecting and Deselecting a Line Object

If the event passed to the line_find method introduced in the previous section results in the location of an object that is not currently selected, the object is made active by invoking its select method.

The select method for the Line object, as seen in Listing 20.9, ensures that handles are created and drawn for the object.

```
Listing 20.9 Selecting a Line Object
```

```
1: static void line_select( GXObjPtr line )
2: {
3: line_bounding_handles( line );
4: gx_draw_handles( line );
5: }
```



The function gx_draw_handles was introduced in Chapter 16, "Object Manipulation," in the section "Managing Object Handles," page 327.

The process of creating the handles for the Line object begins with a call to line_bounding_handles found in Listing 20.10.

Listing 20.10 Creating Handles for the Line Object

```
1:
   static void line bounding handles( GXObjPtr gx line )
2:
   {
3:
      int i, x1, y1, x2, y2, width, height;
4:
5:
      GXLine *line_data = gx_line->data;
6:
7:
      gx line->handles =
8:
        (XRectangle *)XtMalloc( sizeof(XRectangle) * 8 );
      gx line->num handles = 8;
9:
10.
11:
       if( gx line->handles == NULL ) {
         perror( "Alloc failed for line handles" );
12:
13:
         gx line->num handles = 0;
```

```
14:
         return;
15:
       }
16:
17:
       for( i = 0; i < 8; i++ ) {
18:
         gx_line->handles[i].width = HNDL_SIZE;
19:
         gx_line->handles[i].height = HNDL_SIZE;
20:
       }
21:
22:
       get bounds( line data->pts, line data->num pts,
23:
                   &x1, &y1, &x2, &y2 );
24:
       width = x^2 - x^1;
25:
       height = y^2 - y^1;
26:
27:
       gx line->handles[0].x = x1 - HNDL OFFSET - HNDL SIZE;
28:
       gx line->handles[0].y = y1 - HNDL OFFSET - HNDL SIZE;
29:
30:
       gx_line->handles[1].x = x1 + (width/2) - HNDL_OFFSET;
31:
       gx line->handles[1].y = y1 - HNDL SIZE - HNDL OFFSET;
32:
33:
       gx_line->handles[2].x = x2 + HNDL_OFFSET;
34:
       gx_line->handles[2].y = y1 - HNDL_SIZE - HNDL_OFFSET;
35:
36:
       gx_line->handles[3].x = x2 + HNDL_OFFSET;
37:
       gx_line->handles[3].y = y1+(height/2)-HNDL_OFFSET;
38:
       gx line->handles[4].x = x2 + HNDL OFFSET;
39:
       gx_line->handles[4].y = y2 + HNDL_OFFSET;
40:
41:
       gx line->handles[5].x = x1 + (width/2) - HNDL OFFSET;
42:
       gx line -> handles[5].y = y2 + HNDL OFFSET;
43:
44:
       gx line->handles[6].x = x1 - HNDL OFFSET - HNDL SIZE;
45:
       gx_line->handles[6].y = y2 + HNDL_OFFSET;
46:
47:
       gx line->handles[7].x = x1 - HNDL OFFSET - HNDL SIZE;
48:
       gx_line->handles[7].y = y1+(height/2)-HNDL_OFFSET;
49: }
```

The creation of the Line object handles in line_bounding_handles begins by ensuring that the correct number of elements are created in the handles array and the correct handle count is assigned the num_handles field of the common object structure:

```
7: gx_line->handles =
8: (XRectangle *)XtMalloc( sizeof(XRectangle) * 8 );
9: gx_line->num_handles = 8;
```

After testing for a failure of the allocation routine, the widths of all of the handles for the Line object are assigned:

```
17: for( i = 0; i < 8; i++ ) {
18: gx_line->handles[i].width = HNDL_SIZE;
19: gx_line->handles[i].height = HNDL_SIZE;
20: }
```

20

Then, in keeping with the name of the function, the bounds of the Line object are obtained in order to determine the placement of the handles:

```
22: get_bounds( line_data->pts, line_data->num_pts,
23: &x1, &y1, &x2, &y2 );
```



The line_bounding_handles assigns handles at each corner and on every side of the object to mark the bounding box that is required to contain the object.

For a review of object handles see Chapter 16, section "Managing Object Handles," page 327.



The get_bounds function is borrowed from the Text object and is introduced with the Text object internal methods and functions in Chapter 24, "Vector Text Object."

The get_bounds function finds the minimum and maximum points of the segment endpoints contained in the pts array.

Using these extents, the line_bounding_handles determines the width and the height of the object:

```
24: width = x2 - x1;
25: height = y2 - y1;
```

Last comes the tedious task of placing each of the eight handles at the appropriate location around the object in lines 27–48.

The position of the handles runs clock-wise, starting with handles[0] located in the upper-right corner of the object's bounds.

This discussion addressed the for-instance of the line_find method resulting in the location of an object leading to its selection. If, however, the find methods of the objects drawn on the editor's canvas are invoked with no resulting object found, any previously selected object must be deselected.

The Line object's deselect method introduced in Listing 20.11 shows the steps necessary to remove the handles indicating the active state of the Line object.

Listing 20.11 Deselecting a Line Object

```
1: static void line_deselect( GXObjPtr line )
2: {
3: if( line->handles && line->num_handles > 0 ) {
```

```
4: gx_erase_handles( line );
5:
6: XtFree((char *)line->handles );
7:
8: line->handles = NULL;
9: line->num_handles = 0;
10: }
11: }
```

As you can see when reviewing Listing 20.11, it is easier to deselect a Line object than it is to select one.

The deselect method ensure that handles exist for the Line object and erases them from the screen

```
4: gx_erase_handles( line );
```

frees the memory associated with them

```
6: XtFree((char *)line->handles );
```

and returns the handles and num_handles fields to their initial values:

```
8: line->handles = NULL;
9: line->num_handles = 0;
```

In terms of processing, it can seem expensive to destroy the handles completely when an object is deselected. However, the placement of the handles was very sensitive to the location of the Line object on the screen.

As stated earlier, the second purpose of the line_find method (beyond locating an object for selection) is to determine the manipulation action requested by the user. If the object is in motion (being actively moved by the user), the handle placement will no longer be correct at the end of the action.

Therefore, the destruction of the handles and recreation requires much less overhead to ensure the correctness of the object's handles.

This leads to the next section, which introduces the move method that controls the repositioning of the Line object.

Moving a Line Object

The steps required to relocate a Line object on the canvas are provided by the line_move method seen in Listing 20.12.

Listing 20.12 Moving a Line Object

```
1: static void line_move( GXObjPtr line, XEvent *event )
2: {
3: static int x = 0, y = 0;
```

continues

Listing 20.12 Continued

```
4:
5:
      GXLinePtr line_data = (GXLinePtr)line->data;
6:
      int i;
7:
8:
      if( x && y ) {
9:
        XDrawLines( XtDisplay(GxDrawArea),
10:
                    XtWindow(GxDrawArea), rubberGC,
11:
                    line data->pts, line data->num pts,
12:
                    CoordModeOrigin );
13:
      } else {
        /* our first time through */
14:
15:
        (*line->erase)( line );
16:
17:
        x = event ? event->xbutton.x : 0;
18:
        y = event ? event->xbutton.y : 0;
19:
      }
20:
21:
      if( event ) {
22:
        for( i = 0; i < line_data->num_pts; i++ ) {
23:
          line data->pts[i].x += (event->xbutton.x - x);
24:
          line_data->pts[i].y += (event->xbutton.y - y);
25:
        }
26:
27:
        /*
28:
         * draw rubberband line
         */
29:
30:
        XDrawLines( XtDisplay(GxDrawArea),
31:
                    XtWindow(GxDrawArea), rubberGC,
32:
                    line_data->pts, line_data->num_pts,
33:
                    CoordModeOrigin );
34:
35:
        x = event - xbutton.x;
36:
        y = event - xbutton.y;
      } else {
37:
38:
        x = 0;
39:
        y = 0;
40:
      }
41: }
```



As you might recall, the move and scale actions are assigned and invoked from the process_event function introduced in Chapter 16, section "Processing User Navigation of Objects," page 334.

The line_move function uses the static variables x and y to determine whether this invocation is the first time the function has been called:

8: if(x && y) {

If it is not the first time, lines 9–12 erase the rubber-banding copy of the Line object drawn to reflect the object's new location.

However, if x and y are still 0, the else is entered and the Line object's erase method is called to remove it from the screen so that it can be replaced with the interactive version:

15: (*line->erase)(line);

It is important also for the first iteration to save the points associated with the current event:

```
17: x = event ? event->xbutton.x : 0;
18: y = event ? event->xbutton.y : 0;
```



Notice that the if-then-else syntax of lines 17-18 enables the absence of a valid event reference for resetting x and y values to 0 to cancel the move action.

If there is a valid event reference, all points that compose the Line object are moved the difference of the static x and y values and the current event x and y components:

```
22: for( i = 0; i < line_data->num_pts; i++ ) {
23: line_data->pts[i].x += (event->xbutton.x - x);
24: line_data->pts[i].y += (event->xbutton.y - y);
25: }
```

Assuming that every MotionNotify event successfully reaches the line_move function, the object is relocated one pixel at a time.

The interactive depiction of the object is redrawn to the screen for the new point values and the current event point is saved to the static x and y variables in lines 30-36.

Notice again that lines 22–25 directly apply the distance the cursor has moved during the action to the points contained in the line_data->pts array. This is possible because the move transformation does not risk any rounding error.

Whole pixel values are represented by the x and y components of the event structure and their difference from the static x, y are applied directly to the whole pixel values contained in the object. In other words, it is an entirely integer-based calculation.

As demonstrated in the following section, the scale method of the Line object has the potential of suffering data loss because of rounding errors and requires a slightly more intense management of the action.

Scaling a Line Object

The scale method of an object, as introduced in Chapter 16, in the section "Processing User Navigation of Objects," page 334, is assigned as the active action when the user selects one of the active objects handles.

The line_scale method introduced in Listing 20.13 manages the scale action for the Graphics Editor Line object.

```
Listing 20.13 Scaling a Line Object
```

```
1:
   static void line scale( GXObjPtr line, XEvent *event )
2:
   {
3:
      static GXLinePtr tmp data = NULL;
4:
      GXLinePtr line data = (GXLinePtr)line->data;
5:
6:
      if( tmp_data ) {
7:
        XDrawLines( XtDisplay(GxDrawArea),
8:
                    XtWindow(GxDrawArea), rubberGC,
9:
                    tmp_data->pts, tmp_data->num_pts,
10:
                    CoordModeOrigin );
11:
       } else {
12:
         /* our first time... */
13:
         (*line->erase)( line );
14:
15:
         tmp data = (GXLinePtr)XtNew(GXLine);
16:
         tmp data->num pts = line data->num pts;
17:
         tmp data->pts = (XPoint *)
18:
            XtMalloc( sizeof(XPoint) * tmp_data->num_pts );
19:
20:
         get_bounds( line_data->pts, line_data->num_pts,
21:
                     &OrigX, &OrigY, &ExntX, &ExntY );
22:
       }
23:
24:
       if( event ) {
25:
         memcpy( (char *)tmp_data->pts, (char *)line_data->pts,
26:
                 sizeof(XPoint) * tmp_data->num_pts );
27:
28:
         apply_delta( tmp_data->pts, tmp_data->num_pts,
29:
                      FixedX - event->xbutton.x,
30:
                      FixedY - event->xbutton.y );
31:
32:
          XDrawLines( XtDisplay(GxDrawArea),
33:
                      XtWindow(GxDrawArea), rubberGC,
34:
                      tmp data->pts, tmp data->num pts,
35:
                      CoordModeOrigin );
36:
       } else {
37:
         if( tmp_data ) {
38:
           memcpy( (char *)line data->pts,
39:
                    (char *)tmp_data->pts,
40:
                   sizeof(XPoint) * line_data->num_pts );
```

```
41:
42: XtFree((char *)tmp_data->pts);
43: XtFree((char *)tmp_data);
44:
45: tmp_data = NULL;
46: }
47: }
48: }
```

The line_scale method, concerned with compounding rounding errors prone to the floating point calculations of scaling the points contained in the line_data->pts array, makes a copy of the points each time the function is invoked. Then the delta of a FixedX and FixedY and current event point components are applied to the original line_data->pts. This management of data points ensures that the margin of error introduced by the floating-point calculations is absolutely minimized.



For a review of the calculations performed by the apply_delta function as well as the declaration and assignment of the FixedX and FixedY variables, see Chapter 11, section "Scaling a Line," page 234.

Similar in structure to the line_move function, the line_scale function uses the static tmp_data to determine whether the function is called for the first time

```
6: if( tmp_data ) {
```

as a means of updating the interactive Line object reflecting the user's scale actions.

Otherwise, the Line object is erased

```
13: (*line->erase)( line );
```

and the tmp data structure is created:

```
15: tmp_data = (GXLinePtr)XtNew(GXLine);
16: tmp_data->num_pts = line_data->num_pts;
17: tmp_data->pts = (XPoint *)
18: XtMalloc( sizeof(XPoint) * tmp_data->num_pts );
```

Another responsibility of the line_scale action's first invocation is correctly setting the global variables OrigX, OrigY, ExntX, and ExntY to reflect the origin and extents of the object being scaled:

20: get_bounds(line_data->pts, line_data->num_pts, 21: &OrigX, &OrigY, &ExntX, &ExntY);

These global variables are used by the scale support function called by the apply_delta function introduced in Listing 11.4.

20

Then, so long as a valid event reference is passed to line_scale, the original object points are copied to the temporary structure

```
25: memcpy( (char *)tmp_data->pts, (char *)line_data->pts,
26: sizeof(XPoint) * tmp_data->num_pts );
```

and the delta created by the movement of the mouse cursor is applied to the temporary points:

```
28: apply_delta( tmp_data->pts, tmp_data->num_pts,
29: FixedX - event->xbutton.x,
30: FixedY - event->xbutton.y );
```

At some point, the user will release the mouse cursor, ceasing the move action, and the temporary points replace the Line object's original points:

```
38: memcpy( (char *)line_data->pts,
39: (char *)tmp_data->pts,
40: sizeof(XPoint) * line_data->num_pts );
```

With no further need for the temporary structure, it is removed from memory

```
42: XtFree((char *)tmp_data->pts);
43: XtFree((char *)tmp_data);
```

and reset to NULL preparing for the next invocation:

```
45: tmp_data = NULL;
```

The points now contained in the line_data->pts array are a scaled version of the original points with minimal loss of line integrity because of floating-point rounding errors.

Only two more Line object methods await discovery before completing the addition of this object to the Graphics Editor.

The next section introduces the copy method invoked by selecting the copy control icon from the menu panel.

Copying a Line Object

The capability to replicate objects in the Graphics Editor is necessary in order to have an equal number of control functions as draw functions appear in the button panels. (I did not want the menus to be different lengths.)

Truly one of the easiest methods to support, Listing 20.14 introduces the line_copy function for reproducing the currently active object on the editor's canvas.

```
Listing 20.14 Copying a Line Object
```

```
1: static void line_copy( GXObjPtr line )
2: {
```

```
int i;
3:
4:
      GXLinePtr temp_data;
5:
      GXLinePtr line data = (GXLinePtr)line->data;
6:
7:
      (*line->deselect)( line );
8:
9:
     temp data = (GXLine *)XtNew(GXLine);
10:
     temp data->num pts = line data->num pts;
11:
12:
     temp_data->pts = (XPoint *)
13:
          XtMalloc(sizeof(XPoint) * temp data->num pts );
14:
     for( i = 0; i < temp data->num pts; i++ ) {
15:
        temp data->pts[i].x = line data->pts[i].x + OFFSET;
16:
        temp data->pts[i].y = line data->pts[i].y + OFFSET;
17:
      }
18:
      create_line( NULL, temp_data );
19:
20:
      XtFree((char *)temp data );
21: }
```

Comfortable with the line_scale function's requirement to create a temporary copy of the GXLine structure, the line_move function requires a small step to understanding.

Specifically, after ensuring that the active object's handles are removed from the screen

```
7: (*line->deselect)( line );
```

a temporary GXLine structure is created in lines 9–13 and the original object's points are copied to the new structure, but incremented by OFFSET so the new object does not exactly overlay the original:

```
15: temp_data->pts[i].x = line_data->pts[i].x + OFFSET;
16: temp_data->pts[i].y = line_data->pts[i].y + OFFSET;
```

This temporary GXLine structure is used to create a brand new Line object containing the offset copy of the original object's points:

```
19: create_line( NULL, temp_data );
```

The following section presents the last area of functionality required by the Line object, Saving and Restoring the object-specific data.

Saving and Restoring a Line Object

The capability to save and restore objects contained in the Graphics Editor, as introduced in Chapter 19, provides a mature level of capability to our application.

The line_save method found in Listing 20.15 adheres to the save strategy introduced earlier.

Listing 20.15 Saving a Line to a File Object

```
1: static void line save( FILE *fp, GXObjPtr obj )
2: {
3:
        int i;
4:
        GXLinePtr line = (GXLinePtr)obj->data;
5:
6:
        fprintf( fp, "LINE [numpts x y x y ...]\n" );
        fprintf( fp, "%d\n", line->num_pts );
7:
8:
9:
        for( i = 0; i < line->num pts; i++ ) {
           fprintf( fp, "%d %d\n",
10:
11:
                    line->pts[i].x, line->pts[i].y );
12:
        }
13: }
```

The line_save function extracts the GXLine data structure from the common portion of the object

4: GXLinePtr line = (GXLinePtr)obj->data;

and prepares for saving the data it contains by writing a tagged line reflecting the format of the data to follow

```
6: fprintf( fp, "LINE [numpts x y x y ...]\n" );
```

and the number of points that will be written:

7: fprintf(fp, "%d\n", line->num_pts);

Finally, it writes point pairs one per line to the destination file referenced by fp:

```
9: for( i = 0; i < line->num_pts; i++ ) {
10: fprintf( fp, "%d %d\n",
11: line->pts[i].x, line->pts[i].y );
12: }
```

The gxLineLoad function found in Listing 20.16 shows how the data written by the line_save method is restored to the editor.

Listing 20.16 Restoring a Line from a File Object

```
void gxLineLoad( FILE *fp, GXObjPtr obj )
1:
2: {
3:
        int i;
4:
        GXLine line;
5:
        fscanf( fp, "%d\n", &line.num_pts );
6:
7:
8:
        line.pts = (XPoint *)
9:
           XtMalloc(sizeof(XPoint) * line.num_pts);
10:
11:
        for( i = 0; i < line.num pts; i++ ) {</pre>
```

```
12: fscanf( fp, "%hd %hd\n",
13: &line.pts[i].x, &line.pts[i].y );
14: }
15:
16: create_line( obj, &line );
17: }
```

The gxLineLoad function reverses the steps of the line_save method by retrieving first the number of points saved to the file

6: fscanf(fp, "%d\n", &line.num_pts);

creating sufficient memory to store them

```
8: line.pts = (XPoint *)
9: XtMalloc(sizeof(XPoint) * line.num_pts);
```

and then restoring the point pairs from the file:

```
11: for( i = 0; i < line.num_pts; i++ ) {
12: fscanf( fp, "%hd %hd\n",
13: &line.pts[i].x, &line.pts[i].y );</pre>
```

The introduction of the method and function required to save and restore the Line object completes the introduction of this object into the Graphics Editor.

Next Steps

The Latex Line object introduced in this chapter is only one of four pointarray-based objects that share the GXLine object-specific data structure.

In the next chapter I will introduce the Pencil object, which enables a user to create a free-style line using an unlimited number of points.



In this chapter

- Creating a Pencil Object
- Pencil Object Management
- Next Steps

Pencil Line Object

By choosing the Pencil creation icon seen in Figure 21.1, the user enters the interactive creation mode for a freestyle line.

Figure 21.1



Selecting the Pencil creation icon starts the process for a freestyle line.

The creation of the Pencil object begins with the first ButtonPress and ends with the next.

During the interim MotionNotify events separating the ButtonPress events starting and ending the creation, points are added to a points array defining the object under construction.

However, not *every* point is put into the array. This approach would require larger point array storage with no added value because only points that alter the slope of the implied line segment for the Pencil object being created are important enough for storage.

This as well as other aspects of managing the Pencil object creation are introduced in the sections that follow.



The code listings introduced in this chapter are targeted for placement in the gxLine.c source file. Additionally, functions presented in the listings that are not defined static should have a corresponding prototype placed in the gxProtos.h header file.

Creating a Pencil Object

The gx_pencil function introduced in Listing 21.1 is assigned as the function to invoke when the Pencil creation icon is selected from the menu panel through the definition of the gxDrawIcons array found in Chapter 13:

```
{ &pen_icon, (void (*)(void))gx_pencil,
 "Draw a freestyle line..." },
```

```
Listing 21.1 The gx_pencil Function
```

```
1: void gx pencil( XEvent *event )
2:
   {
      static GXLinePtr rubber_pencil = NULL;
3:
4:
      static int
                        ptCnt
                                       = 0;
5:
6:
      if( event == NULL ) {
7:
        rubber pencil = NULL;
8:
      } else {
9:
       /*
10:
        * remove any current rubber banding...
        */
11:
12:
        if( rubber_pencil ) {
13:
          XDrawLines( XtDisplay(GxDrawArea),
14.
                       XtWindow(GxDrawArea), rubberGC,
15:
                       rubber pencil->pts,
16:
                       rubber pencil->num pts,
17:
                       CoordModeOrigin );
18:
        }
19:
20:
        switch( event->type ) {
21:
          case ButtonPress:
            if( rubber pencil == NULL ) {
22:
23:
              rubber pencil =
24:
                  update_pencil( event, rubber_pencil );
25:
              set cursor( PENCIL MODE );
26:
            } else {
27:
              create_line( NULL, rubber_pencil );
28:
              gx refresh();
29:
30:
              set cursor( NORMAL MODE );
              rubber_pencil = 0;
31:
32:
              ptCnt = 0;
33:
            }
34:
          break;
35:
36:
          case ButtonRelease:
37:
          case MotionNotify:
38:
            /*
39:
             * update the GXLine structure based on the
40:
             * new point and current location of the mouse
             */
41:
42:
            if( rubber_pencil ) {
43:
              (void)update pencil( event, rubber pencil );
```

```
44:
               XDrawLines( XtDisplay(GxDrawArea),
45:
                           XtWindow(GxDrawArea), rubberGC,
46:
                            rubber pencil->pts,
47:
                            rubber pencil->num pts,
48:
                            CoordModeOrigin );
49:
             }
50:
          break;
51:
        }
52:
      }
53: }
```

The gx_pencil function is structured identically to the gx_line function controlling the creation of the Latex Line object, with one exception. Where the gx_line function invoked the update_line function to record the event point in the temporary pts array being constructed by the user, the gx_pencil function employs the update_pencil function introduced in Listing 21.2.

Listing 21.2 The update_pencil Function

```
static GXLinePtr update_pencil( XEvent *event, GXLinePtr pencil )
1:
2: {
3:
                        pen = {NULL, 0};
      static GXLine
4:
      GXLinePtr penPtr = &pen;
5:
6:
      int x = -1, y = -1;
7:
8:
      gx new vertex( event, pencil, &x, &y );
9:
10:
      /* start over if the user is being silly... */
11:
      if( pen.num pts > 1024 ) {
12:
        pen.num pts = 0;
13:
      }
14:
15:
      if( (x > 0) \& (y > 0)) {
16:
        if( !pencil ) {
17:
          pen.pts = (XPoint *)XtMalloc( sizeof(XPoint) );
18:
          pen.num pts = 0;
19:
        } else {
          pen.pts = (XPoint *)
20:
21:
            XtRealloc( (char *)pen.pts,
                        sizeof(XPoint) * (pen.num_pts + 1));
22:
23:
        }
24:
25:
        pen.pts[pen.num pts].x = x;
26:
        pen.pts[pen.num pts].y = y;
27:
28:
        pen.num_pts++;
29:
      }
30:
31:
      return penPtr;
32: }
```

21

The update_pencil function begins by immediately calling the function gx_new_vertex to determine whether the current event point should be added to the pts array being constructed by the Pencil object creation process:

```
8: gx_new_vertex( event, pencil, &x, &y );
```

Consideration of this function will follow in a moment; however, whether a point is added to the array or not, the function continues by ensuring that some ridiculously large number of points had not been requested

resetting to the beginning if necessary.

Following this validation, the function sees whether a valid point was returned from the gx_new_vertex function:

15: if((x > 0) & (y > 0)) {

Because the initial x and y values were set to -1, any value returned from gx_new_vertex indicates the need to process them.

The update_pencil function then determines, based on the value of the GXLine reference pen, whether this is the first time through the function

```
16: if( !pencil ) {
```

to determine which allocation routine to invoke to create space for the point being added. On the first pass through this function, the pen value is NULL and the XtMalloc function is used to gain the initial memory assignment:

```
17: pen.pts = (XPoint *)XtMalloc( sizeof(XPoint) );
18: pen.num_pts = 0;
```

On subsequent invocations the XtRealloc function is used to expand the memory currently assigned the pen structure reference:

```
21: XtRealloc( (char *)pen.pts,
22: sizeof(XPoint) * (pen.num_pts + 1));
```

Finally, the new point is added to the array

```
25: pen.pts[pen.num_pts].x = x;
26: pen.pts[pen.num_pts].y = y;
```

and the running total of the point count incremented accordingly:

```
28: pen.num_pts++;
```

Look again at the gx_new_vertex function introduced in Listing 21.3 and used by update_pencil to determine whether the point contained in the event structure is worthy of being added to the points array for the Pencil object under creation.

```
Listing 21.3 Determine Whether a New Vertex Should Be Stored
```

```
static void gx new vertex( XEvent *xe,
1:
2:
                            GXLinePtr upd, int *x, int *y )
3:
    {
4:
      float a = -1.0, b = -1.0;
5:
6:
      if( upd &&
7:
         (upd->num pts > 0) && (xe->xbutton.x > 0) ) {
8:
9:
        /*
            see if the slope has changed
10:
         *
11:
         */
        a = (float)upd->pts[upd->num pts - 1].y /
12:
            (float)upd->pts[upd->num pts - 1].x;
13:
14:
15:
        b = (float)xe->xbutton.y / (float)xe->xbutton.x;
16:
17:
        if( a != b ) {
18:
          *x = xe->xbutton.x;
19:
          *y = xe->xbutton.y;
20:
        }
21:
      }
22: }
```

First ensuring that there is a valid point for comparison contained in the upd structure passed as the second parameter to the function

```
6: if( upd &&
7: (upd->num_pts > 0) && (xe->xbutton.x > 0) ) {
```

the gx_new_vertex also ensures that the value of the x component of the event point is not 0 because this would result in a fatal application error when used as the denominator of a divide calculation later in the function.

After the validation step of lines 6–7, two slope values are calculated. The first

```
12: a = (float)upd->pts[upd->num_pts - 1].y /
13: (float)upd->pts[upd->num_pts - 1].x;
```

is for the previous point added to the update structure. The second

15: b = (float)xe->xbutton.y / (float)xe->xbutton.x;

is for the current event point.



To review slope calculations for a line, return to Chapter 10, "Trigonometric and Geometric Functions," in the section "Calculating Slope," page 216.
The function then compares the two slope calculations to determine whether they are equal:

```
17: if( a != b ) {
```

Only if they are not equal are they returned to the calling function for addition into the pts array of the object being created:

18: *x = xe->xbutton.x; 19: *y = xe->xbutton.y;

No value is gained by adding points with like slopes to the array, because the XDrawLines function will ensure that points connecting endpoints are drawn to the screen.

This method of evaluating points before adding them to the object minimizes the space required to represent them by not including needless information.

When the ButtonPress event marking the end of the creation process for the Pencil object is received, the create_line function introduced in the previous chapter is invoked to create a Line object:

```
27: create_line( NULL, rubber_pencil );
```

As described in the next section, no other functions are unique to the management of the Pencil object.

Pencil Object Management

The Pencil object, through requiring a creation process unique to the object, ends up being a point-array-based object indistinguishable from the Latex Line object.

The Pencil object exists (along with the Latex Line) as an array of points contained within a GXLine structure. This enables all the methods assigned by the create_line function to adequately manage the Pencil object as it does the Latex Line, introduced in Chapter 20.

Next Steps

Two other point-array-based objects exist that employ the GXLine data structure to represent their object-specific data. These, too, will end up as point-array-based objects using the same methods for management as the Latex Line and Pencil objects. However, like the Pencil object, they require a unique creation process.

The following chapter introduces the creation process for two other pointarray-based objects supported by the Graphics Editor: the Box object and the Arrow object.

Chapter 22

In this chapter

- The Box Object
- The Arrow Object
- Next Steps

Object Templates

The Box and Arrow objects supported by the editor are considered template objects because the interactive process of creating these objects enforces the shape appropriate to the object.

The following sections introduce the Box and Arrow objects. You will notice many similarities between these objects and the Latex Line and Pencil objects introduced in previous chapters.



The code listings introduced in this chapter are targeted for placement in the gxLine.c source file. Additionally, functions presented in the listings that are not defined static should have a corresponding prototype placed in the gxProtos.h header file.

The Box Object

By choosing the icon shown in Figure 22.1, a user begins the Box object creation process.



The Box object creation icon.

During the creation of the Box object, the user moves the mouse cursor to the canvas window and presses and holds the mouse button while dragging to form the box dimensions desired.

Releasing the mouse button ends the creation process, and a Box object consistent with the size specified during the interactive phase of the creation is added to the list of objects known to the editor.

The gx_box function introduced in Listing 22.1 satisfies the assignment made to the gxDrawIcons array in Chapter 13, "Application Structure," for the function to invoke when the gx_box icon is selected:

```
{ &box_icon, (void (*)(void))gx_box,
   "Draw a square or rectangle..." },
```

```
Listing 22.1 The gx_box Function
```

```
void gx_box( XEvent *event )
1:
2:
   {
3:
      static XRectangle *rubber box = NULL;
4:
5:
      if( event == NULL ) {
6:
        rubber box = NULL;
7:
      } else {
8:
        /*
         * remove the current rubberband if there is one
9:
10:
         */
        if( rubber box ) {
11:
12:
          XDrawRectangle(XtDisplay(GxDrawArea),
13:
                          XtWindow(GxDrawArea), rubberGC,
14:
                          rubber box->x, rubber box->y,
15:
                          rubber box->width,
16:
                          rubber box->height );
17:
        }
18:
19:
        switch( event->type ) {
20:
          case ButtonPress:
21:
            rubber_box = update_box( event, NULL );
            set cursor( EDIT MODE );
22:
23:
          break;
24:
          case ButtonRelease:
25:
26:
            if( rubber box ) {
27:
              create_line(NULL, line_from_box(rubber_box));
28:
              gx_refresh();
29:
            }
30:
            set_cursor( NORMAL_MODE );
31:
            rubber_box = NULL;
32:
          break;
33:
34:
          case MotionNotify:
35:
            if( rubber box ) {
36:
               (void)update_box( event, rubber_box );
37:
              XDrawRectangle(XtDisplay(GxDrawArea),
                              XtWindow(GxDrawArea),
38:
39:
                              rubberGC,
```

```
40: rubber_box->x, rubber_box->y,
41: rubber_box->width,
42: rubber_box->height);
43: }
44: break;
45: }
46: }
47: }
```

The gx_box function differs from the creation routine gx_line and gx_pencil only in the form of the temporary XRectangle rubber_box data structure employed, the update applied to it

36: (void)update_box(event, rubber_box);

and the processing of the XRectangle to convert it to a GXLine structure:

27: create_line(NULL, line_from_box(rubber_box));

The update_box function is introduced in Listing 22.2 and the line_from_box conversion routine is introduced in Listing 22.3.

Listing 22.2 Updating the Temporary Box Points

```
static
1:
2: XRectangle *update_box( XEvent *event, XRectangle *upd )
3:
   {
      static int fix x = 0, fix y = 0;
4:
5:
      static XRectangle box;
6:
7:
      XRectangle *boxPtr = &box;
8:
9:
      if( upd == NULL ) {
10:
        fix x = event->xbutton.x;
11:
        fix y = event->xbutton.y;
12:
      }
13:
14:
      box.x = min( fix_x, event->xbutton.x );
      box.y = min( fix_y, event->xbutton.y );
15:
16:
      box.width = max( fix_x, event->xbutton.x )-box.x;
17:
18:
      box.height = max( fix_y, event->xbutton.y )-box.y;
19:
20:
      return boxPtr;
21: }
```

The challenge of the update_box function is to determine when the user flips the box over one of the x- or y-axis during creation.

To understand this, first consider that the upper-left corner of the box is placed at the ButtonPress event location indicating the start of the box creation, and the lower corner of the box is placed at the ButtonRelease event location ending the creation. When the event location marking the end of the creation process is less than the event location starting the creation, the object has flipped. This is important to catch because we don't want to specify a negative width or height value for the rectangle defining the Box object. It is the responsibility of the update_box function to account for a flipped condition.

The update_box function begins by determining whether this is the first call to this function for the current creation as indicated by the value of upd being NULL:

```
9: if( upd == NULL ) {
10: fix_x = event->xbutton.x;
11: fix_y = event->xbutton.y;
12: }
```

At the first invocation for a creation, the static variables fix_x and fix_y must be assigned the corresponding components of the event point.

This fixed point will enable the function to prevent subsequent events resulting in the box being flipped over either the x- or the y-axis.

The XRectangle reference by box is then assigned an x, y location that is the smaller of the values fix_x, fix_y, and the corresponding event point components:

```
14: box.x = min( fix_x, event->xbutton.x );
15: box.y = min( fix_y, event->xbutton.y );
```

From the location assignment of box, the width and height are set as the larger of fix_x, fix_y, and the corresponding event point components less whatever x, y values were assigned to box:

```
17: box.width = max( fix_x, event->xbutton.x )-box.x;
18: box.height = max( fix_y, event->xbutton.y )-box.y;
```

Following the successful update of the interactive box defining the wishes of the user, the creation process will end when she releases the mouse button. When the creation ends, the XRectangle used during the creation process must be converted to a GXLine structure for a point-array-based object to result from this process.

Listing 22.3 introduces the line_from_box function for converting between the two structures.

Listing 22.3 Converting a Box Object to a Line Object

```
1: static GXLinePtr line_from_box( XRectangle *box )
2: {
3: static GXLine line;
4:
5: line.pts = (XPoint *)XtMalloc( sizeof(XPoint) * 5);
6: line.num_pts = 5;
7:
```

```
8:
      line.pts[0].x = box->x;
      line.pts[0].y = box->y;
9:
10:
11:
      line.pts[1].x = box->x + box->width;
12:
      line.pts[1].y = box->y;
13:
14:
      line.pts[2].x = box->x + box->width;
15:
      line.pts[2].y = box->y + box->height;
16:
17:
      line.pts[3].x = box->x;
18:
      line.pts[3].y = box->y + box->height;
19:
20:
      line.pts[4].x = box->x;
      line.pts[4].y = box->y;
21:
22:
23:
      return &line;
24: }
```

The steps required to convert an XRectangle to a GXLine structure are very simple and only require that we ensure the resulting GXLine object is closed by repeating the first point as the last point.

The line_from_box function creates space enough for five points in the GXLine pts array

```
5: line.pts = (XPoint *)XtMalloc( sizeof(XPoint) * 5);
6: line.num_pts = 5;
```

and then it assigns each corner of the rectangle to a corresponding element of the pts array in lines 8–18, repeating the first point in the last element:

```
20: line.pts[4].x = box->x;
21: line.pts[4].y = box->y;
```

Without this last step, the resulting line object would appear as a sideways U with the open end facing to the left.

With a GXLine structure representing the box created by the user, the create_line function is called and management continues for this object as it does for both the Latex Line and Pencil objects.

Considerations similar to those for the Box object exist for the Arrow object introduced in the next section.

The Arrow Object

By choosing the Arrow creation icon shown in Figure 22.2 from the Graphics Editor menu panel, a user is able to begin the creation process for the Arrow object.



icon.

The Arrow object, like the Box object, is considered a template object because the Arrow shape is enforced by the creation process. Also similar to the Box object is the fact that the ButtonPress event begins the creation process and the ButtonRelease event ends it.

The event point associated with the ButtonPress determines the upper-left corner of the object being created and the ButtonRelease event point marks the lower-right corner of the object.

Listing 22.4 introduces the gx_arrow function assigned in the gxDrawIcons array in Chapter 13.

```
Listing 22.4 The gx_arrow Function
```

```
1: void gx_arrow( XEvent *event )
2:
   {
3:
      static XRectangle *rubber_box = NULL;
4:
      GXLinePtr arrow;
5:
6:
      if( event == NULL ) {
7:
        rubber box = NULL;
8:
      } else {
9:
        if( rubber box ) {
          arrow = arrow_from_box( rubber_box );
10:
11:
          XDrawLines(XtDisplay(GxDrawArea),
12:
                      XtWindow(GxDrawArea), rubberGC,
13:
                      arrow->pts, arrow->num pts,
14:
                      CoordModeOrigin );
15:
        }
16:
17:
        switch( event->type ) {
18:
          case ButtonPress:
19:
            rubber box = update box( event, NULL );
20:
            set cursor( EDIT MODE );
21:
22:
            break;
23:
24:
          case ButtonRelease:
25:
            if( rubber box ) {
              create line(NULL,arrow from box(rubber box));
26:
27:
              gx_refresh();
            }
28:
            set_cursor( NORMAL_MODE );
29:
            rubber_box = NULL;
30:
31:
32:
          break;
33:
```

```
34:
          case MotionNotify:
35:
            if( rubber_box ) {
36:
               (void)update box( event, rubber box );
37:
               arrow = arrow_from_box( rubber_box );
38:
39:
              XDrawLines(XtDisplay(GxDrawArea),
40:
                          XtWindow(GxDrawArea), rubberGC,
41:
                          arrow->pts, arrow->num pts,
42:
                          CoordModeOrigin );
43:
44:
            }
45:
          break;
46:
        }
47:
      }
48: }
```

The gx_arrow function should be immediately familiar because it follows the same structure as the gx_box (and its predecessors).

The only difference, in fact, between the gx_box function and the gx_arrow function is the arrow_from_box function used to convert from the XRectangle structure to the GXLine structure in the following lines:

26:	<pre>create_line(NULL,arrow_from_box(rubber_box));</pre>
38:	arrow = arrow_from_box(rubber_box);

The arrow_from_box function is defined in Listing 22.5.

Listing 22.5 Creating an Arrow from a Box

```
1: static GXLinePtr arrow from box( XRectangle *box )
2: {
3:
      static GXLine arrow;
4:
      static XPoint pts[10];
5:
6:
      pts[0].x = box ->x + (box ->width/10);
7:
      pts[0].y = box->y + box->height;
8:
9:
      pts[1].x = box -> x + (box -> width / 4);
10:
      pts[1].y = box->y + box->height - (box->height / 8);
11:
12:
      pts[2].x = box -> x + (box -> width / 4);
13:
      pts[2].y = box->y + (box->height / 2);
14:
15:
      pts[3].x = box->x;
16:
      pts[3].y = box->y + (box->height / 2);
17:
18:
      pts[4].x = box -> x + (box -> width / 2);
19:
      pts[4].y = box->y;
20:
21:
      pts[5].x = box -> x + box -> width;
22:
      pts[5].y = box->y + (box->height / 2);
```

```
23:
24:
      pts[6].x = box - >x + ((box - >width * 3) / 4);
25:
      pts[6].y = box->y + (box->height / 2);
26:
27:
      pts[7].x = box->x + ((box->width * 3) / 4);
28:
      pts[7].y = box->y + box->height - (box->height / 8);
29:
30:
      pts[8].x = box -> x + box -> width - (box -> width / 10);
31:
      pts[8].y = box->y + box->height;
32:
33:
      arrow.pts
                    = pts;
34:
      arrow.num pts = 9;
35:
36:
      return &arrow;
37: }
```

The arrow_from_box function defines enough room to hold 10 XPoint structures:

```
4: static XPoint pts[10];
```

The arrangement of points in the array is illustrated in Figure 22.3.

Figure 22.3 An illustration of the arrangement of points in an Arrow.



Lines 6–31 meticulously assign the point values based on the proportions and ratios consistent with point positions shown in Figure 22.3.



Nothing is truly scientific about dividing a box into a series of points to achieve the shape of an arrow.

I accomplished it through trial and error until the results were aesthetically pleasing.

It is important that ratios of the box be used to create the points defining the arrow, however, in order to enable the aspect ratio (width to height proportion) to be honored in the resulting GXLine structure.

The point array is then assigned as the pts value of the GXLine structure along with the correct number of points

```
33: arrow.pts = pts;
34: arrow.num_pts = 9;
```

and the address of the GXLine structure arrow is returned for use in either drawing the interactive object being navigated by the user or in the creating the final pointarray-based object:

36: return &arrow;

This completes the introduction of the Box and Arrow template objects supported by the Graphics Editor.

Next Steps

With the Box and Arrow objects comfortably at rest in the Graphics Editor project, you have successfully added a total of four objects. Because they are all pointarray-based objects they are able to borrow heavily from a core structure.

In Chapter 23, the Arc object is added to the list of objects supported by the editor. Because the representation of an arc is not similar to the GXLine objects seen so far, there will be differences to point out. However, the basic event-driven creation, moving, and scaling processes introduced for the Latex Line and inherited by the Pencil, Box, and Arrow objects will remain the same.

Chapter 23

In this chapter

- Creating an Arc Object
- Drawing and Erasing an Arc Object
- Finding an Arc Object
- Selecting and Deselecting an Arc Object
- Moving an Arc Object
- Scaling an Arc Object
- Copying an Arc Object
- Saving and Restoring an Arc Object
- Next Steps

Arc Object

By choosing the Arc creation icon seen in Figure 23.1, the user begins the process of creating an Arc object.

Figure 23.1



Choosing the Arc creation icon begins the process of creating an Arc object.

The interactive process of creating an Arc object is similar to creating the Box or Arrow objects discussed in Chapter 22, "Object Templates."

Specifically, a ButtonPress event begins the creation process and a ButtonRelease event ends it. The event point corresponding to the ButtonPress assigns the location of the upper-left corner of the bounding box that is to contain the Arc object, whereas the event point for the ButtonRelease determines the lower-right corner of this box.

This chapter presents the internal methods and supporting functions defined for the Arc object. These routines control all aspects of managing the different features of the object.



The code listings introduced in this chapter are targeted for placement in the gxArc.c source file. Additionally, functions presented in the listings that are not defined static should have a corresponding prototype placed in the gxProtos.h header file.

Creating an Arc Object

The definition of the gxDrawIcons array introduced in Chapter 13

```
{ &arc_icon, (void (*)(void))gx_arc,
   "Draw a circle..." },
```

assigned the Arc icon the gx_arc function for invocation when selected by the user. This function is defined in Listing 23.1 and serves as the entry point to the creation process of the Arc object.

```
Listing 23.1 The gx_arc Function
```

```
1: void gx_arc( XEvent *event )
2: {
3:
      static XArc *rubber arc = NULL;
4:
5:
      if( event == NULL ) {
6:
        rubber arc = NULL;
7:
      } else {
        if( rubber arc &&
8:
9:
            rubber arc->width > 0 &&
10:
            rubber arc->height > 0) {
11:
          XDrawArc( XtDisplay(GxDrawArea),
                   XtWindow(GxDrawArea), rubberGC,
12:
13:
                   rubber_arc->x, rubber_arc->y,
14:
                   rubber_arc->width, rubber_arc->height,
                    rubber arc->angle1, rubber arc->angle2);
15:
16:
        }
17:
18:
        switch( event->type ) {
19:
          case ButtonPress:
20:
            /*
             * initailize an XArc structure to retain
21:
             * the arc info during rubber banding
22:
23:
             */
24:
            rubber arc = update arc( event, NULL );
25:
            set_cursor( EDIT_MODE );
26:
          break;
27:
        case ButtonRelease:
28:
29:
          /*
30:
           * create the arc, and mark it selected as
31:
           * the current object
32:
           */
33:
          if( rubber arc &&
34:
              rubber arc->width && rubber arc->height ) {
35:
            create arc( NULL, rubber arc );
36:
            gx refresh();
37:
          }
38:
          set_cursor( NORMAL_MODE );
40:
```

```
41:
          break;
42:
        case MotionNotify:
43:
          /*
44:
           * update the XArc structure based on the
45:
           * new location of the mouse pointer
46:
           */
47:
          if( rubber arc ) {
48:
            (void)update_arc( event, rubber_arc );
49:
50:
            /*
51:
             * redraw the rubberbanding arc
             */
52:
53:
            if( rubber arc->width && rubber arc->height ) {
54:
              XDrawArc( XtDisplay(GxDrawArea),
                    XtWindow(GxDrawArea), rubberGC,
55:
56:
                    rubber arc->x, rubber arc->y,
                    rubber_arc->width, rubber_arc->height,
57:
                    rubber arc->angle1, rubber arc->angle2);
58:
59:
            }
60:
          }
61:
          break;
39:
          rubber arc = NULL;
62:
        }
63:
      }
64: }
```

Structured similarly to the creation functions reviewed in previous chapters, the gx_arc function is driven by the events it receives informing of user navigation.

The function begins by testing for a valid event reference passed by the caller:

```
5: if( event == NULL ) {
6: rubber_arc = NULL;
```

Remember that a NULL event enables the function to reset and cancel any pending creation process.

If a valid event has been sent, the gx_arc function tests the presence of a valid Arc reference stored in the rubber_arc variable:

```
8: if( rubber_arc &&
9: rubber_arc->width > 0 &&
10: rubber_arc->height > 0) {
```

If a valid Arc definition exists, the function erases the previously drawn interactive Arc object pending the update of the rubber_arc structure:

11:	XDrawArc(XtDisplay(GxDrawArea),
12:	<pre>XtWindow(GxDrawArea), rubberGC,</pre>
13:	rubber_arc->x, rubber_arc->y,
14:	rubber_arc->width, rubber_arc->height,
15:	rubber_arc->angle1, rubber_arc->angle2);

The gx_arc function then switches, based on the type of event sent, and takes the appropriate action:

```
18: switch( event->type ) {
```

When the user presses the mouse button, the creation process begins and the rubber_arc structure is assigned its initial values:

24: rubber_arc = update_arc(event, NULL);

The movement of the cursor further updates the rubber_arc structure to reflect the user's navigation:

48: (void)update_arc(event, rubber_arc);

However, when the mouse button is released, an actual Arc object is created:

35: create_arc(NULL, rubber_arc);

Because the structure of the gx_arc function has been thoroughly established with the introduction of similar functions previously, I won't go into great detail. Interesting, however, are the functions unique to the creation of the Arc object, specifically update_arc, introduced in Listing 23.2, and create_arc found in Listing 23.3.

Listing 23.2 The update_arc Function

```
1: static XArc *update_arc( XEvent *event, XArc *upd )
2: {
      static int fix x = 0, fix y = 0;
3:
      static XArc arc;
4:
5:
     XArc *arcPtr = &arc;
6:
7:
      int x1, y1, x2, y2, rx, ry;
8:
9:
      if( upd == NULL ) {
10:
        arc.angle1 = 0*64;
11:
        arc.angle2 = 360*64;
12:
13:
       fix_x = event->xbutton.x;
14:
        fix y = event->xbutton.y;
15:
      }
16:
17:
      rx = event->xbutton.x - fix x;
18:
     ry = event->xbutton.y - fix y;
19:
20:
     x1 = fix_x + rx;
     x^2 = fix x - rx;
21:
22:
     y1 = fix y + ry;
23:
     y^2 = fix y - ry;
24:
```

```
25: arc.x = min(x1, x2);
26: arc.y = min(y1, y2);
27:
28: arc.width = max(x1, x2) - arc.x;
29: arc.height = max(y1, y2) - arc.y;
30:
31: return arcPtr;
32: }
```

The update_arc function has similar responsibilities to the update_box function introduced in Chapter 22. The update_arc function must manage when the distance the cursor has traveled since the start of the creation could result in negative values for the width and height fields of the XArc structure.

Starting with a test for a NULL upd reference to mark the beginning of the creation process

9: if(upd == NULL) {

the update_arc function makes the necessary initializations for processing subsequent events correctly:

```
10: arc.angle1 = 0*64;
11: arc.angle2 = 360*64;
13: fix_x = event->xbutton.x;
14: fix_y = event->xbutton.y;
```

Thinking ahead to consecutive calls to the update_arc function during the creation process, the deltas (or radii) for the x and y-axis are calculated from the fixed points and the current event point:

```
17: rx = event->xbutton.x - fix_x;
18: ry = event->xbutton.y - fix_y;
```

From the x and y-axis radius values rx and ry, the point defining the corners of the bounding box can be found:

```
20: x1 = fix_x + rx;
21: x2 = fix_x - rx;
22: y1 = fix_y + ry;
23: y2 = fix_y - ry;
```

Using these points, you assign the smaller of the pairs as the x and y origin in the XArc structure

```
25: arc.x = min(x1, x2);
26: arc.y = min(y1, y2);
```

and the greater of them as the horizontal and vertical extent:

28: arc.width = max(x1, x2) - arc.x; 29: arc.height = max(y1, y2) - arc.y;

When the interactive mode of the Arc creation yields the object desired by the user, she releases the mouse button and ends the creation process. When this happens, the actual Arc object is created by a call to the create_arc function introduced in Listing 23.3.

Listing 23.3 The create_arc Function

```
static void create_arc( GXObjPtr _obj, XArc *arc )
1:
2:
   {
3:
      XArc *arc_data;
4:
      GXObjPtr obj = _obj;
5:
6:
      if( obj == NULL ) {
7:
          obj = gx_create_obj();
8:
      }
9:
10:
      arc_data = (XArc *)XtNew( XArc );
11:
      obj->data = (void *)arc data;
12:
13:
      arc data->width = arc->width;
      arc_data->height = arc->height;
14:
15:
16:
      arc_data -> x = arc -> x;
      arc_data->y = arc->y;
17:
18:
19:
      arc_data->angle1 = arc->angle1;
20:
      arc data->angle2 = arc->angle2;
21:
22:
     obj->draw = arc draw;
23:
     obj->erase = arc erase;
     obj->find = arc find;
24:
25:
     obj->move = arc move;
26: obj->scale = arc_scale;
27:
     obj->copy = arc copy;
28:
29:
      obj->save = arc_save;
30:
31:
      obj->select = arc_select;
32:
      obj->deselect = arc_deselect;
33:
34:
      gx_add_obj( obj );
35: }
```

As with the create_line function exhausted in earlier chapters, the create_arc function accounts for being called with the common object portion already created in order to support the restore function introduced later in this chapter.

Barring the presence of a GXObj reference, the create_arc function must request one:

7: obj = gx_create_obj();

The function creates a new reference to an XArc structure and assigns it to the data field of the GXObj:

10: arc_data = (XArc *)XtNew(XArc); 11: obj->data = (void *)arc_data;

The values of the XArc defined by the user during creation are transferred to the new reference:

```
13: arc_data->width = arc->width;
14: arc_data->height = arc->height;
and
16: arc_data->x = arc->x;
17: arc_data->y = arc->y;
and
19: arc_data->angle1 = arc->angle1;
20: arc_data->angle2 = arc->angle2;
```

Finally, the function is ready to assign the methods to the GXObj that will properly manage and control the Arc data portion of this object (lines 22–32).

The last thing the create_arc function does is ensure that the newly created object is appended to the list of objects currently managed by the editor application:

34: gx_add_obj(obj);

As was true with the Line objects in Chapters 20–22, for an object to be visible on the canvas it must be drawn. The following section introduces the methods that draw and erase the Arc object.

Drawing and Erasing an Arc Object

Again following the form of the line_draw and line_erase methods assigned to Line objects, except for the arc_erase being responsible for erasing the object's handles, the arc_draw and arc_erase methods differ only in the value passed for the tile flag as seen in Listing 23.4.

Listing 23.4 Drawing and Erasing an Arc Object

```
1: static void draw_erase( GXObjPtr arc, Boolean tile )
2: {
3: GC gc;
4: XArc *arc_data;
5:
6: gc = gx_allocate_gc( arc, tile );
7: arc data = (XArc *)arc->data;
```

Listing 23.4 Continued

```
8:
9:
      XDrawArc( XtDisplay(GxDrawArea),
10:
                XtWindow(GxDrawArea), gc,
11:
                arc data->x, arc data->y,
                arc data->width, arc data->height,
12:
13:
                arc data->angle1, arc data->angle2 );
14.
      XtReleaseGC( GxDrawArea, gc );
15:
16: }
17:
18: static void arc draw( GXObjPtr obj )
19: {
20:
      draw erase( obj, False );
21: }
22:
23: static void arc erase( GXObjPtr obj )
24: {
25:
      gx erase handles( obj );
26:
      draw erase( obj, True );
27: }
```

True for any of the objects added to the editor thus far, if the object is visible it can be selected by the user. The following section introduces the method for determining whether an event successfully selected an Arc object.

Finding an Arc Object

The work for finding an Arc object was done in Chapter 10, "Trigonometric and Geometric Functions." The arc_find method was introduced in Listing 10.2 in the section "Calculating Point and Arc Intersections," page 218, which explains the mathematics required to perform the necessary calculations.

If successfully located by the event passed to the arc_find method, the object must be updated as selected. The following section provides the select and deselect methods assigned to the Arc object.

Selecting and Deselecting an Arc Object

The select method of the arc_object introduced in Listing 23.5 is responsible for creating and displaying the bounding handles associated with the object.

Listing 23.5 Selecting an Arc Object

```
1: static void arc_select( GXObjPtr arc )
2: {
3: arc_bounding_handles( arc );
4: gx_draw_handles( arc );
5: }
```

The arc_select method is a simple process until we introduce the arc_bounding_handles function in Listing 23.6, which creates the handles assigned to the Arc object.

Listing 23.6 Creating Arc Handles

```
static void arc bounding handles( GXObjPtr gx arc )
1:
2:
   {
     int i;
3:
4:
      XArc *arc = gx arc->data;
5:
6:
      gx arc->handles = (XRectangle *)XtMalloc(sizeof(XRectangle) * 8);
7:
      gx arc->num handles = 8;
8:
9:
      if( gx_arc->handles == NULL ) {
10:
        fprintf(stderr, "Alloc failed for arc handles" );
11:
        gx arc -> num handles = 0;
12:
        return;
13:
      }
14:
15:
      for(i = 0; i < 8; i++) {
16:
        gx arc->handles[i].x =
17:
        gx_arc->handles[i].y = 0;
18:
19:
        gx arc->handles[i].width = HNDL SIZE;
20:
        gx arc->handles[i].height = HNDL SIZE;
21:
      }
22:
23:
      gx arc->handles[0].x = arc->x - HNDL SIZE - HNDL OFFSET;
24:
      gx_arc->handles[0].y = arc->y - HNDL_SIZE - HNDL_OFFSET;
25:
26:
      gx_arc->handles[1].x = arc->x + (arc->width/2) - HNDL_OFFSET;
27:
      gx_arc->handles[1].y = arc->y - HNDL_SIZE - HNDL_OFFSET;
28:
29:
      gx arc->handles[2].x = arc->x + arc->width + HNDL OFFSET;
30:
      gx_arc->handles[2].y = arc->y - HNDL_SIZE - HNDL_OFFSET;
31:
32:
      gx arc->handles[3].x = arc->x + arc->width + HNDL OFFSET;
33:
      gx_arc->handles[3].y = arc->y +(arc->height/2) - HNDL_OFFSET;
34:
35:
      gx arc->handles[4].x =
36:
                       arc->x + arc->width + HNDL OFFSET;
37:
      gx arc->handles[4].y =
38:
                       arc->y + arc->height + HNDL OFFSET;
39:
40:
      gx arc->handles[5].x =
41:
                       arc->x + (arc->width/2) - HNDL_OFFSET;
42:
      gx_arc->handles[5].y =
43:
                       arc->y + arc->height + HNDL OFFSET;
44:
45:
      gx_arc->handles[6].x = arc->x - HNDL_SIZE - HNDL_OFFSET;
```

Listing 23.6 Continued

The arc_bounding_handles function creates eight handles for positioning at every corner and every side of the object:

```
6: gx_arc->handles = (XRectangle *)XtMalloc(sizeof(XRectangle) * 8);
7: gx_arc->num_handles = 8;
```

Following steps to ensure that the allocation succeeds for the handles, a for loop is entered to initialize the width and height fields:

```
19: gx_arc->handles[i].width = HNDL_SIZE;
20: gx arc->handles[i].height = HNDL SIZE;
```

Finally, a process identical to assigning the positions of the Line object handles introduced in Chapter 20, "Latex Line Object," in the section "Selecting and Deselecting a Line Object," page 392, is used to place the arc bounding handles. One difference, however, is that the XArc structure stored in the object inherently provides the bounds of the object, eliminating the need to calculate them.

A selected object is a candidate for deselecting; therefore the arc_deselect method is defined in Listing 23.7 to support this requirement.

Listing 23.7 Deselecting an Arc Object

```
static void arc_deselect( GXObjPtr arc )
1:
2:
   {
3:
      if( arc->handles && arc->num handles > 0 ) {
4:
        gx erase handles( arc );
5:
6:
        XtFree((char *)arc->handles );
7:
8:
        arc->handles = NULL;
9:
        arc->num handles = 0;
10:
      }
11: }
```

The deselect method ensures that the handles exist for this object and takes steps to remove them from the screen and release the memory they occupy, resetting the object fields to reflect their defunct state:

8: arc->handles = NULL; 9: arc->num_handles = 0; Because the user has selected an Arc object for the purpose of manipulating it in some way, the following sections present the arc_move and arc_scale functions.

Moving an Arc Object

The steps required to relocate an Arc object on the screen are provided by the arc_move function, seen in Listing 23.8.

```
Listing 23.8 The arc_move Function
```

```
static void arc_move( GXObjPtr arc, XEvent *event )
1:
2: {
3:
      static int x = 0, y = 0;
4:
      XArc *arc data = (XArc *) arc->data;
5:
6:
      if( x && y ) {
7:
        /*
         * erase the rubberband arc
8:
9:
         */
10:
        XDrawArc( XtDisplay(GxDrawArea),
                  XtWindow(GxDrawArea), rubberGC,
11:
12:
                  arc_data->x, arc_data->y,
13:
                  arc_data->width, arc_data->height,
14:
                  arc data->angle1, arc data->angle2 );
15:
16:
      } else {
17:
        /*
18:
         * our first time through - erase the actual arc...
         */
19:
        (*arc->erase)( arc );
20:
21:
        /*
22:
23:
        * ...store the current event location
24:
         */
25:
        x = event ? event->xbutton.x : 0;
26:
        y = event ? event->xbutton.y : 0;
27:
      }
28:
29:
      if( event ) {
30:
        /*
31:
         * get the x,y delta
         */
32:
33:
        arc_data->x += (event->xbutton.x - x);
        arc data->y += (event->xbutton.y - y);
34:
35:
36:
        /*
         * draw a rubberband arc
37:
38:
         */
39:
        XDrawArc( XtDisplay(GxDrawArea),
```

Listing 23.8 Continued

```
40:
                   XtWindow(GxDrawArea), rubberGC,
41:
                   arc_data->x, arc_data->y,
42:
                   arc_data->width, arc_data->height,
43:
                   arc data->angle1, arc data->angle2 );
44:
45:
        x = event - xbutton.x;
46:
        y = event - xbutton.y;
47:
      } else {
48:
        x = 0;
        y = 0;
49:
50:
      }
51: }
```

As you might recall, the move and scale actions are assigned and invoked from the process_event function introduced in Chapter 16, "Object Manipulation," in the section "Processing User Navigation of Objects," page 334.

The arc_move function follows the same form as the line_move introduced in Chapter 20, in the section "Moving a Line Object," page 395. The only difference to note is that by simply updating the x and y values of the XArc structure, the repositioning of the Arc object is accomplished:

```
33: arc_data->x += (event->xbutton.x - x);
34: arc_data->y += (event->xbutton.y - y);
```

The following section introduces the arc_scale method assigned to Arc objects.

Scaling an Arc Object

The scale method of an object, as introduced in Chapter 16, in the section "Processing User Navigation of Objects," page 334, is assigned as the active action when the user selects one of the active objects handles.

The arc_scale function introduced in Listing 23.9 manages the scale action for the Graphics Editor Arc object.

```
Listing 23.9 The arc_scale Function
```

```
static void arc scale( GXObjPtr arc, XEvent *event )
1:
2:
   {
3:
      static XArc *tmp data = NULL;
4:
5:
      if( tmp_data ) {
6:
        /*
         * erase the rubberband arc
7:
8:
         */
9:
        XDrawArc( XtDisplay(GxDrawArea),
10:
                 XtWindow(GxDrawArea), rubberGC,
```

```
11:
                   tmp data->x, tmp data->y,
12:
                   tmp_data->width, tmp_data->height,
13:
                   tmp data->angle1, tmp data->angle2 );
14:
15:
      } else {
16:
        /*
17:
         * our first time through - erase the actual arc
18:
         */
19:
        (*arc->erase)( arc );
20:
21:
        tmp data = (XArc *)XtNew( XArc );
22:
      }
23:
24:
      if( event ) {
25:
        memcpy( (char *)tmp data,
26:
                 (char *)arc->data, sizeof(XArc));
27:
        apply_delta( tmp_data,
28:
                      FixedX - event->xbutton.x,
29:
                      FixedY - event->xbutton.y );
30:
31:
        /*
32:
         * draw a rubberband arc
         */
33:
34:
        XDrawArc( XtDisplay(GxDrawArea),
35:
                  XtWindow(GxDrawArea), rubberGC,
36:
                   tmp_data->x, tmp_data->y,
37:
                   tmp data->width, tmp data->height,
                   tmp_data->angle1, tmp_data->angle2 );
38:
39:
40:
      } else {
41:
        if( tmp data ) {
42:
           memcpy( (char *)arc->data, (char *)tmp_data,
43:
                   sizeof(XArc));
44:
45:
          XtFree( (char *)tmp_data );
46:
          tmp_data = NULL;
47:
        }
48:
      }
49: }
```

 \rightarrow The steps required to scale the Arc object follow those required to scale a Line object as discussed in Chapter 20, in the section "Scaling a Line Object," page 398.



For a review of the calculations preformed by the apply_delta function as well as the declaration and assignment of the FixedX and FixedY variables, see Chapter 11, "Graphic Transformations," in the section "Scaling an Arc," page 243, and Listing 11.7 "The apply_delta Function for Arcs," page 245.

Copying an Arc Object

The arc_copy function, introduced in Listing 23.10, is simple to implement and convenient for users.

Listing 23.10 The arc_copy Function

```
1: static void arc copy( GXObjPtr arc )
2:
   {
3:
     XArc *temp_data;
4:
      (*arc->deselect)(arc);
5:
6:
      temp data = (XArc *)XtNew(XArc);
7:
      memcpy( (char *)temp_data, (char *)arc->data, sizeof(XArc));
8:
9:
     temp data->x += OFFSET;
10: temp data->y += OFFSET;
11:
     create arc( NULL, temp data );
12:
13:
      XtFree( (char *)temp data );
14: }
```

The arc_copy function deselects the active object's handles

4: (*arc->deselect)(arc);

creates memory to hold the XArc data

```
6: temp_data = (XArc *)XtNew(XArc);
```

copies the data associated with the previously active object

```
7: memcpy( (char *)temp_data, (char *)arc->data, sizeof(XArc));
```

increments the location of the copied data by an OFFSET

```
9: temp_data->x += OFFSET;
10: temp data->y += OFFSET;
```

and employs the create_arc function to create a new object

11: create_arc(NULL, temp_data);

followed by clearing the temporary XArc structure:

13: XtFree((char *)temp_data);

The last elements required for satisfying the creation of the Arc object is to add support for saving and restoring the Arc data.

Saving and Restoring an Arc Object

The capability to save and restore objects contained in the Graphics Editor, as introduced in Chapter 19, "Save and Restore," provides a mature level of capability to our application.

The arc_save function found in Listing 23.11 adheres to the save strategy introduced earlier.

```
Listing 23.11 The arc_save Function
```

```
static void arc save( FILE *fp, GXObjPtr arc )
1:
2:
   {
3:
        XArc *arc data = (XArc *)arc->data;
4:
5:
        fprintf( fp, "ARC [x y width height angle1 angle2]\n");
6:
        fprintf( fp, "%d %d %d %d %d %d \n",
7:
8:
                     arc data->x, arc data->y,
9:
                      arc_data->width, arc_data->height,
10:
                      arc data->angle1, arc data->angle2 );
11: \}
```

Extracting the XArc data from the object

3: XArc *arc_data = (XArc *)arc->data;

writing the tagged line to indicate the format of the data to follow

5: fprintf(fp, "ARC [x y width height angle1 angle2]\n");

and finally, writing the data

```
7: fprintf(fp, "%d %d %d %d %d %d \n",
8: arc_data->x, arc_data->y,
9: arc_data->width, arc_data->height,
10: arc_data->angle1, arc_data->angle2);
```

are the steps required to save the data associated with the Arc object.

The steps to restore the data to the editor are introduced in Listing 23.12.

Listing 23.12 The gxArcLoad Function

```
void gxArcLoad( FILE *fp, GXObjPtr obj )
1:
2:
   {
3:
        XArc arc:
4:
5:
        fscanf( fp, "%hd %hd %hd %hd %hd \n",
6:
                     &arc.x, &arc.y, &arc.width,
7:
                     &arc.height, &arc.angle1, &arc.angle2 );
8:
9:
        create_arc( obj, &arc );
10: }
```

The data line is the only thing of import to the gxArcLoad function. Defining a local XArc structure and loading the saved data into the appropriate fields

```
5: fscanf(fp, "%hd %hd %hd %hd %hd %hd \n",
6: &arc.x, &arc.y, &arc.width,
7: &arc.height, &arc.angle1, &arc.angle2);
```

and passing the address of the structure to the create_arc function

9: create_arc(obj, &arc);

retrieves the previously saved data and creates an object from it.

This completes the all aspects of introducing the Arc object to the Graphics Editor.

Next Steps

One object remains for introduction, integration, and investigation: namely, the Text object.

The Text object by far is the most complex of all the objects supported by the Graphics Editor. However, upon successful completion of the following chapter, you will have a functional editor capable of drawing, moving, scaling, saving, and restoring a variety of graphics objects.

Chapter 24

In this chapter

- Creating a Text Object
- Drawing and Erasing a Text Object
- Finding a Text Object
- Selecting and Deselecting a Text Object
- Moving a Text Object
- Scaling a Text Object
- Copying a Text Object
- Saving and Restoring a Text Object
- Next Steps

Vector Text Object

By choosing the Text creation icon shown in Figure 24.1, the user begins the creation process for the Text object.





The Text creation icon begins the Text object creation process.

The creation process for the Text object begins with prompting the user for a text string from which to create the Text object.

With the string successfully entered by the user, a value reflecting the interactive text object follows the movement of the cursor. The user places the text on the canvas window by pressing the mouse button. The resulting ButtonPress event location is used as the origin of the new Text object.

This chapter presents the internal methods and supporting functions defined for the Text object. These routines control all aspects of managing the different features of the object.



The code listings introduced in this chapter are targeted for placement in the gxText.c source file. Additionally, functions presented in the listings that are not defined static should have a corresponding prototype placed in the gxProtos.h header file.

Creating a Text Object

The definition of the gxDrawIcons array introduced in Chapter 13

```
{ &text_icon, (void (*)(void))gx_text,
   "Draw dynamic text..." },
```

assigned the Text icon the gx_text function for invocation when selected by the user. This function is defined in Listing 24.1, and it serves as the entry point to the interactive creation process of the Text object.

```
Listing 24.1 The gx_text Function
```

```
1: void gx_text( XEvent *event )
2: {
3:
      static char *creation text = NULL;
4:
5:
      if( event == NULL ) {
6:
        creation text = NULL;
7:
        place creation text( NULL, NULL );
8:
9:
        /*
         * we have to prompt for a string
10:
11:
         */
12:
        creation text = get creation text();
13:
14:
        if( creation_text != NULL )
          set cursor( TEXT MODE );
15:
16:
17:
      } else {
18:
        /*
19:
20:
         * If we have a string, place it!
         */
21:
22:
        if( creation text ) {
23:
          /* adjust for the hotspot in our cursor */
24:
          event->xbutton.y -= 10;
25:
          event->xbutton.x += 10;
26:
27:
          place_creation_text( event, &creation_text );
28:
        }
29:
      }
30: }
```

At the specification of a NULL event reference, the gx_text function initializes local variables to a known state both in the gx_text function and the place_creation_text function:

```
5: if( event == NULL ) {
6: creation_text = NULL;
7: place_creation_text( NULL, NULL );
```

Following this, the user is prompted for input with a call to the get_creation_text function:

12: creation_text = get_creation_text();

If a string is successfully gained from the user, the cursor mode is updated to reflect the state of the application and

15: set_cursor(TEXT_MODE);

the function finishes for the current iteration. The function is called again when the user moves the cursor into the canvas window and begins to generate MotionNotify events. Because the create_text variable is static, its value is available for the successive calls:

```
22: if( creation_text ) {
```

With the presence of the creation_text string value, the event points are adjusted to account for the hotspot of the cursor

24: event->xbutton.y -= 10; 25: event->xbutton.x += 10;

and then used to locate the interactive text string for the placement:

27: place_creation_text(event, &creation_text);

Listing 24.2 introduces the place_creation_text function used to indicate to the user the future location of the text object based on their navigation.

Listing 24.2 The place_creation_text Function

```
static void
1:
2: place_creation_text( XEvent *event, char **_text )
3: {
4:
      static GXTextPtr rubber text = NULL;
5:
     char *text = NULL;
6:
7:
     if( event == NULL ) {
8:
        rubber text = NULL;
      } else {
9:
10:
        if( _text ) text = *_text;
11:
        if( rubber_text ) {
12:
          GXDrawText( rubber_text, rubberGC );
13:
14:
        }
15:
16:
        switch( event->type ) {
17:
          case ButtonPress:
18:
            if( rubber_text && text ) {
19:
              create_text( NULL, rubber_text );
```

continues

```
Listing 24.2 Continued
```

```
20:
               gx refresh();
21:
22:
               freeGXText( rubber text );
23:
               rubber_text = NULL;
24:
25:
               free( text );
26:
               * text = NULL;
27:
28:
               set_cursor( NORMAL_MODE );
29:
             }
30:
             break;
31:
32:
          case MotionNotify:
33:
            if( text ) {
34:
               if( rubber text == NULL ) {
                 rubber_text = update_gxtext( event,
35:
36:
                                                text, NULL );
               } else {
37:
                 (void)update_gxtext( event,
38:
39:
                                        text, rubber text );
40:
41:
               }
42:
               GXDrawText( rubber_text, rubberGC );
43:
             }
44:
             break;
45:
        }
46:
      }
47: }
```

The place_creation_text function determines its actions based on the current event type.

Notice that the parameters required for the place_creation_text function include not only a reference to the event being processed but also a pointer to the character pointer containing the creation_text:

```
2: place_creation_text( XEvent *event, char **_text )
```

By passing the address to the creation text, its contents can be cleared when the user completes the placement process as indicated by a ButtonPress event.

Beyond validating the presence of valid event and copying the text variable to a local reference in lines 7-10, the place_creation_text function determines whether the rubber_text variable was previously initialized. A valid rubber_text value indicates that the function must erase the current interactive text string:

```
12: if( rubber_text ) {
13: GXDrawText( rubber_text, rubberGC );
14: }
```

The function is now ready to determine the event type and execute the appropriate action body:

```
16: switch( event->type ) {
```

Consider first the MotionNotify action body in lines 32–44.

The action for MotionNotify must invoke the update_gxtext to either establish the initial value of the rubber_text variable (lines 35–36) or to update the variable for subsequent notifications of mouse motion and draw the temporary GXText structure referenced by rubber_text.

The action appropriate for the ButtonPress

17: case ButtonPress:

must ensure that a valid rubber_text value is assigned as well as current text string

18: if(rubber_text && text) {

and invokes the create_text routine to create the Text object:

19: create_text(NULL, rubber_text);

After updating the canvas window with a call to gx_refresh on line 20, the function frees the memory associated with the rubber_text variable used during the interactive creation:

22: freeGXText(rubber_text);

Several functions referenced during the course of this discussion have yet to be introduced. Review first the function to retrieve a string by prompting the user, introduced in Listing 24.3.

Listing 24.3 Prompting the User for a String

```
1: static void
2: close_dialog(Widget w, XtPointer cdata, XtPointer cbs)
3: {
4:
     Widget dialog = (Widget)cdata;
5:
6:
      if( dialog ) XtUnmanageChild( dialog );
7: }
8:
9: static char *get_creation_text( void )
10: {
     XtAppContext app;
11:
12:
    XEvent
                  event;
13:
14: Widget dialog;
15: char *str = NULL;
```

Listing 24.3 Continued

```
16:
17:
      dialog = XtVaCreateManagedWidget( "Text Entry Box",
18:
                                         dialogWidgetClass,
19:
                                         GxDrawArea,
20:
                                         XtNwidth, 115,
21:
                                         XtNheight, 70,
22:
                                         XtNlabel,
23:
                                            "Enter Text:",
                                         XtNvalue, "",
24:
25:
                                         NULL );
26:
27:
      XawDialogAddButton( dialog, " Ok ",
28:
                           close_dialog, dialog );
29:
30:
      app = XtWidgetToApplicationContext( GxDrawArea );
31:
32:
      while( XtIsManaged(dialog)) {
33:
        XtAppNextEvent( app, &event );
34:
        XtDispatchEvent( &event );
35:
      }
36:
37:
      str = XawDialogGetValueString( dialog );
38:
      XtDestroyWidget( dialog );
39:
40:
      /*
41:
       * look for 'illegal' characters
       */
42:
43:
      {
44:
        int c, indx = 0;
45:
        char illegal_chars[] = { '\n', '\r' };
46:
47:
        while( (c = (int)str[indx]) != '\0' ) {
48:
          if( strchr( illegal chars, c ) != NULL )
              str[indx] = ' ';
49:
50:
          indx++;
51:
        }
52:
53:
        /*
54:
         * remove leading zeros
55:
         */
56:
        while( *str && *str == ' ' ) str++;
57:
      }
58:
59:
      if( str && *str )
60:
        return XtNewString(str);
61:
      else
62:
        return NULL;
63: }
```

The get_creation_text function is identical to the function used to retrieve a filename from the user for the gx_save and gx_load functions.

Review the discussion of the GxGetFileName function in Chapter 19, in the section "Common Object Save and Restore," page 367, for how the get_creation_text function works.

Listing 24.4 reviews the GxDrawText function originally introduced in Chapter 15, in the section "Text Object Data Structure," page 307.

```
Listing 24.4 The GXDrawText Function
```

```
1: static void GXDrawText( GXTextPtr text, GC gc )
2: {
3:
      char *txt = text->text;
4:
      int c, chr, nsegs, num_pts;
5:
6:
      for( c = 0; c < text->len; c++, txt++ ) {
7:
        chr = *txt - ' ';
8:
        nsegs = 0;
9:
        while( text->font[chr][nsegs] != NULL ) {
10:
11:
          num_pts = text->fontp[chr][nsegs];
12:
13:
          if( num_pts > 0 ) {
            XDrawLines( XtDisplay(GxDrawArea),
14:
15:
                        XtWindow(GxDrawArea), gc,
16:
                         text->vpts[c][nsegs], num pts,
                        CoordModeOrigin );
17:
18:
          }
19:
          nsegs++;
20:
        }
21:
      }
22: }
```

The GxDrawText function is responsible for traversing the characters comprising the Text object

```
6: for( c = 0; c < text->len; c++, txt++ ) {
7: chr = *txt - ' ';
```

then traversing the segments defining the character

```
10: while( text->font[chr][nsegs] != NULL ) {
```

and for each set of points defining the segment:

11: num_pts = text->fontp[chr][nsegs];

This function also invokes the XDrawLines Graphic Primitive to draw that portion of the character

```
13: if( num_pts > 0 ) {
14: XDrawLines( XtDisplay(GxDrawArea),
15: XtWindow(GxDrawArea), gc,
16: text->vpts[c][nsegs], num_pts,
17: CoordModeOrigin );
18: }
```

before proceeding to the next segment:

19: nsegs++;

The update_gxtext function introduced in Listing 24.5 is a complex function used to create a GXText structure from the character string and event location taken from the parameters passed to the function.

Listing 24.5 The update_gxtext Function

```
static GXTextPtr
1:
2: update gxtext( XEvent *xe, char *str, GXTextPtr upd )
3: {
4:
       GXTextPtr text = NULL;
5:
6:
       if( upd ) {
         reset_pts( upd, xe->xbutton.x, xe->xbutton.y, 0, 0 );
7:
8:
       } else {
9:
         text = create_gxtext( str, xe->xbutton.x,
10:
                               xe->xbutton.v.
11:
                               plain simplex,
12:
                               plain simplex p );
13:
       }
14:
15:
       return text;
16: }
```

The update_gxtext function branches in one of two ways. If a GXText structure is already referenced by upd, the reset_pts is invoked to update the structure for the new location specified by the event point:

7: reset_pts(upd, xe->xbutton.x, xe->xbutton.y, 0, 0);

However, if a GXText structure is not referenced currently by upd, the create_gxtext function is invoked to create one:

9:	<pre>text = create_gxtext(st</pre>	r, xe->xbutton.x,
10:	X	e->xbutton.y,
11:	p	lain_simplex,
12:	p.	<pre>lain_simplex_p);</pre>

Notice the last two parameters passed to the create_gxtext function, plain_simplex and plain_simplex_p.

The plain_simplex array is a vector font set. More literally, it contains the definitions of all characters that can be represented by the font set. The plain_simplex_p is a control array containing the number of points contained in each segment of the corresponding character found in the plain_simplex array.

The definition of these arrays is shown in Listing 24.6; however, to begin understanding it, refer to the Excursion "An Introduction to Vector Fonts," in Chapter 15, in the section "Text Object Data Structure," page 307.

```
Listing 24.6 The vfonts/simplex.h Header File
```

```
#ifndef SIM INC H
1:
2:
    #define SIM_INC_H
3:
4:
    #include "vfonts/vector_chars.h"
5:
6:
   XPoint **plain simplex[] = {
7:
     char699, char714, char717, char733, char719, char2271,
8:
     char734, char731, char721, char722, char2219, char725,
9:
     char711, char724, char710, char720, char700, char701,
10:
     char702, char703, char704, char705, char706, char707,
11:
    char708, char709, char712, char713, char2241, char726,
12:
    char2242, char715, char2273, char501, char502, char503,
13: char504, char505, char506, char507, char508, char509,
14: char510, char511, char512, char513, char514, char515,
15: char516, char517, char518, char519, char520, char521,
16:
     char522, char523, char524, char525, char526, char2223,
     char804, char2224, char2262, char999, char730, char601,
17:
18: char602, char603, char604, char605, char606, char607,
19:
     char608, char609, char610, char611, char612, char613,
20:
     char614, char615, char616, char617, char618, char619,
     char620, char621, char622, char623, char624, char625,
21:
22:
    char626, char2225, char723, char2226, char2246, char718,
23: };
24:
25: int *plain_simplex_p[] = {
26: char p699, char p714, char p717, char p733, char p719,
27: char p2271, char p734, char p731, char p721, char p722,
28: char p2219, char p725, char p711, char p724, char p710,
29: char_p720, char_p700, char_p701, char_p702, char_p703,
30: char_p704, char_p705, char_p706, char_p707, char_p708,
31: char p709, char p712, char p713, char p2241, char p726,
32: char_p2242, char_p715, char_p2273, char_p501, char_p502,
33:
    char p503, char p504, char p505, char p506, char p507,
34: char_p508, char_p509, char_p510, char_p511, char_p512,
35:
     char_p513, char_p514, char_p515, char_p516, char_p517,
     char_p518, char_p519, char_p520, char_p521, char_p522,
36:
37:
     char_p523, char_p524, char_p525, char_p526, char_p2223,
38: char_p804, char_p2224, char_p2262, char_p999, char_p730,
39: char_p601, char_p602, char_p603, char_p604, char_p605,
40:
     char_p606, char_p607, char_p608, char_p609, char_p610,
```
Listing 24.6 Continued

```
41: char_p611,char_p612,char_p613,char_p614,char_p615,
42: char_p616,char_p617,char_p618,char_p619,char_p620,
43: char_p621,char_p622,char_p623,char_p624,char_p625,
44: char_p626,char_p2225,char_p723,char_p2226,char_p2246,
45: char_p718,
46: };
47:
48: #endif /* SIM_INC_H */
```

The plain_simplex array contains the character definitions for all characters that can be represented by this vector font set.



acters, when constructed, were numbered sequentially. Later, as I cleaned up the definitions, removing those not referenced and grouping others into sets, it was necessary to repeat some characters representing simple punctuation and, in the end, the sets had a sort of mismatched appearance.

Appendix C, "Additional Vector Font Sets and vector_chars.h," presents all the vector fonts sets available to the Graphics Editor project as well as the contents of the vector_chars.h file. Because the file contains the definitions of all characters from all font sets (which consist of multiple segment definitions for each character), the file is literally thousands of lines long. Therefore, you probably do not want to type it in. In this case, I encourage to you reference the pertinent files on the CD-ROM accompanying this book.



Chapter 28, "Extending the Graphics Editor," outlines ways to incorporate all available vector font sets into the Graphics Editor application concurrently.

A sample of the character for the exclamation point is presented in Listing 24.7.

```
Listing 24.7 Definition of Exclamation Point (char714)
```

```
static XPoint seg0 714[] = {
1:
2:
         (0, -12\}, \{0, 2\},\
3: };
4:
5: static XPoint seg1_714[] = {
6:
   \{0,7\}, \{-1,8\}, \{0,9\}, \{1,8\}, \{0,7\},
7: };
8:
9: static XPoint *char714[] = {
10: seg0 714, seg1 714,
11: NULL,
12: };
13:
14: static int char p714[] = {
     XtNumber(seg0 714), XtNumber(seg1 714),
15:
16: };
```

A second example is provided in Listing 24.8, which contains the definition of the double quote character.

```
Listing 24.8 Definition of Double Quote (char717)
```

```
static XPoint seg0_717[] = {
1:
2:
      \{-4, -12\}, \{-4, -5\},
3: };
4:
5: static XPoint seg1_717[] = {
6:
   {4,-12}, {4,-5},
7: };
8:
9: static XPoint *char717[] = {
10: seg0_717, seg1_717,
11: NULL,
12: };
13:
14: static int char p717[] = {
15: XtNumber(seg0 717), XtNumber(seg1 717),
16: };
```

The relationship between a font set such as plain_simplex (definition of all characters) and a single character contained in the array (defined by a series of segments), and each of the segments comprising a character (arrays of XPoints), should be clear from these examples.

Further, the plain_simplex_p array serves the purpose of providing the number of points available for each segment defining a character.

With an understanding of the structure of a vector font set well in hand, turn your attention to the create_gxtext function introduced in Listing 24.9.

```
Listing 24.9 The create_gxtext Function
```

```
1: GXTextPtr create gxtext( char *text,
2:
                             int x, int y,
3:
                             GXFont fnt, GXFontP fntp )
4:
   {
5:
     GXTextPtr text_data;
6:
      int c, chr, s, num_pts, nsegs;
7:
8:
     text_data = (void *)XtNew( GXText );
9:
10:
     text data->x = x;
11:
     text_data -> y = y;
12:
13:
     text data->dx = 0;
14:
     text data->dy = 0;
15:
16:
      text data->text = XtNewString( text );
17:
     text_data->len = strlen( text );
18:
19:
     text_data->font = fnt;
20:
     text_data->fontp = fntp;
21:
22:
     text data->vpts = (XPoint ***)
23:
           XtMalloc(sizeof(XPoint **) * text_data->len);
24:
25:
     for( c = 0; c < text data -> len; c++, text++ ) {
26:
        chr = *text - ' ';
27:
28:
        nsegs = 0;
29:
        while( fnt[chr][nsegs] != NULL ) {
30:
         nsegs++;
31:
        }
32:
33:
        text data->vpts[c] = (XPoint **)
34:
         XtMalloc( sizeof( XPoint * ) * nsegs );
35:
36:
        for( s = 0; s < nsegs; s^{++}) {
37:
          num pts = fntp[chr][s];
38:
39:
          text data->vpts[c][s] = (XPoint *)
40:
            XtMalloc( sizeof( XPoint ) * num_pts );
41:
        }
42:
      }
43:
44:
      reset_pts( text_data,
45:
                 text_data->x, text_data->y,
46:
                 text data->dx, text data->dy );
47:
48:
      return text data;
49: }
```

The create_gxtext function begins by allocating a new instance of the GXText structure

```
8: text_data = (void *)XtNew( GXText );
```

followed by initializing the static fields of the GXText structure. The x and y fields track the origin of the Text object and are applied to all points of the font set as they are loaded into the GXText structure. The initial value for these fields will be the coordinates of the event point:

10: text_data->x = x; 11: text_data->y = y;

The dx and dy fields manage the scale factors for the x and y-axis applied to this Text object since creation. Their usefulness will be demonstrated shortly, but upon creation no scale deltas have been requested, so their initial values are 0:

```
13: text_data->dx = 0;
14: text_data->dy = 0;
```

A copy of the character string comprising the new object is stored in the GXText structure along with the string's width:

```
16: text_data->text = XtNewString( text );
17: text_data->len = strlen( text );
```

Finally, to support future expansion of dynamically specifying the font set to apply for individual Text objects, the font particular to this object is assigned to the GXText structure. Also assigned is the control array which tracks the number of points for the segments of the various characters understood by the font set:

```
19: text_data->font = fnt;
20: text_data->fontp = fntp;
```

The function then creates space to store the vector characters for the text string comprising the Text object

```
22: text_data->vpts = (XPoint ***)
23: XtMalloc(sizeof(XPoint **) * text_data->len);
```

and proceeds to traverse the text string one character at a time:

25: for(c = 0; c < text_data->len; c++, text++) {

Because the first character understood by any font set is the space, all characters are subtracted from the decimal value of the space to ensure the index into the font set is correctly aligned for the current character:

26: chr = *text - ' ';

Using this character, the number of segments needed to define it using the font set is determined with a while loop:

```
28: nsegs = 0;
29: while( fnt[chr][nsegs] != NULL ) {
30: nsegs++;
31: }
```

Knowing the required number of segments allows us to allocate sufficient space to store them:

```
33: text_data->vpts[c] = (XPoint **)
34: XtMalloc( sizeof( XPoint * ) * nsegs );
```

Finally, the control array indicating the number of points for each segment is employed to allocate sufficient space to store the points after they are transposed to be relative to the origin:

```
36: for( s = 0; s < nsegs; s++ ) {
37: num_pts = fntp[chr][s];
38:
39: text_data->vpts[c][s] = (XPoint *)
40: XtMalloc( sizeof( XPoint ) * num_pts );
```

With the proper space reserved for the string defining the new GXText structure, the reset_pts function is used to transpose the points defining the segments of the characters in the text string:

```
44: reset_pts( text_data,
45: text_data->x, text_data->y,
46: text_data->dx, text_data->dy );
```

Before presenting the reset_pts function, note the number of allocations performed for each character of the string assigned to the GXText structure.

The GXText structure is a memory-intensive construct because it requires memory allocation for the transformed points within the various segments. The actual number of allocations depends on the number of segments, which is influenced by the complexity of the font.

Therefore it is important that proper management be applied to this memory and that it be returned to the heap when no longer used by the application. For this reason, the freeGXText function found in Listing 24.10 has been defined.

```
Listing 24.10 Freeing the Memory Allocated by the create_gxtext Function
```

```
1: static void freeGXText( GXTextPtr text_data )
2: {
3: int c, chr, nsegs;
3: char *text = text_data->text;
4:
```

```
5:
      for( c = 0; c < text data -> len; c++, text++ ) {
        chr = *text - ' ';
6:
7:
8:
        nsegs = 0;
9:
        while( text_data->font[chr][nsegs] != NULL ) {
10:
11:
          XtFree( (char *)text data->vpts[c][nsegs] );
12:
          nsegs++;
13:
        }
14:
        XtFree( (char *)text data->vpts[c] );
15:
      }
16:
      XtFree( (char *)text data->vpts );
17:
      XtFree( (char *)text data->text );
18: }
```

The freeGXText function effectively works backward to perform frees where the create_gxtext performs allocations.

Look now at the reset_pts function used by the create_gxtext function and defined in Listing 24.11.

Listing 24.11 Resetting Vector Points to Maintain Text Quality

The reset_pts function is responsible for first applying the x, y origin specified in the parameter list to the points defined in the font set and then applying the dx and dy scale factors to the transformed points.

The function reset_font_pts shown in Listing 24.12 accomplishes the task of transforming the points in the font set to the specified origin.

```
Listing 24.12 The reset_font_pts Function
```

```
static void reset font pts( GXTextPtr text,
1:
2:
                                int x, int y )
3: {
4:
     char *txt = text->text;
5:
     int i, c,
                               /* loop counters
                                                       */
                              /* index to chr in font */
6:
         chr,
                              /* num segs for chr
7:
         nsegs,
                                                       */
                                                       */
                              /* num pts for seg
8:
         num pts,
                              /* max extent for a chr */
9:
         maxx,
                             /* where chr begins
                                                      */
10:
         orig x = x,
         c_off = 0;
11:
                             /* offset from prev chr */
```

Listing 24.12 Continued

```
12:
13:
      text -> x = x;
14:
     text->y = y;
15:
16:
      maxx = -9999;
17:
      for( c = 0; c < text->len; c++, txt++ ) {
        chr = *txt - ' ';
18:
19:
20:
        nseqs = 0;
        while( text->font[chr][nsegs] != NULL ) {
21:
22:
          num pts = text->fontp[chr][nsegs];
23:
24:
          for( i = 0; i < num_pts; i++ ) {</pre>
            /*
25:
26:
             * reset vpts from font relative origin of this
27:
             * character applying screen scale
28:
             */
29:
            text->vpts[c][nsegs][i].x =
30:
              (text->font[chr][nsegs][i].x + c_off) +
31:
                                                    orig_x;
32:
            text->vpts[c][nsegs][i].y =
33:
              text->font[chr][nsegs][i].y + y;
34:
35:
            maxx = max( maxx, text->vpts[c][nsegs][i].x );
36:
          }
37:
          nsegs++;
        }
38:
39:
40:
        /*
         * look ahead at width of next char setting orig x
41:
         * to ensure no overlap with the current character
42:
43:
         */
44:
        if( (c + 1) < text->len )
45:
          c_off = next_char_min( text, *(txt + 1) );
46:
47:
        orig_x = SPC(maxx);
48:
      }
49: }
```

Following the declaration of the variables that will control the multiple levels of looping necessary to traverse the characters, segments, and points defining the object, the function resets the x, y origin fields maintained in the GXText structure:

```
13: text->x = x;
14: text->y = y;
```

The mechanism for using each character of the text string contained in the GXText structure to traverse the segments defined by the font set is consistent throughout the support functions used by the Text object. Therefore focus here is on the purpose of the function.

In other words, lines 16–22 are identical to those used previously to access the points which comprise the current character's definition by the font set.

Unique functionality begins when the vpts structure created at the beginning of the create_gxtext function is assigned the transformed points taken from the font set.

```
29: text->vpts[c][nsegs][i].x =
30: (text->font[chr][nsegs][i].x + c_off) +
31: orig_x;
32: text->vpts[c][nsegs][i].y =
33: text->font[chr][nsegs][i].y + y;
```

The transformation applied to the x points contained in the font set

```
text->font[chr][nsegs][i].x
```

consists of applying a character offset c_off and the value of the x component of the origin to determine the value of the corresponding element in the vpts array,

29: text->vpts[c][nsegs][i].x =

which will later be used to actually draw the object.

The transformation for the y component is a little more direct:

```
32: text->vpts[c][nsegs][i].y =
33: text->font[chr][nsegs][i].y + y;
```

The next step taken by the reset_font_pts function is to track the furthest x point for this character:

35: maxx = max(maxx, text->vpts[c][nsegs][i].x);

The maxx value will be important for ensuring that subsequent characters are properly spaced to the right of the current one.

When all the points for all the segments for the current character are transformed and placed in the appropriate vpts element, the while loop ends

38:

ł

and the offset for the next character is calculated:

```
44: if( (c + 1) < text->len )
45: c_off = next_char_min( text, *(txt + 1) );
```

Finally, the origin of the Text object is advanced by the SPC macro

47: orig_x = SPC(maxx);

and the for loop advances to the next character for the text string of this object. The transformation begins for the next character.

The macro definition should be placed at the beginning of the gxText.c file:

#define SPC(w) (w + 3) /* just a wee gap between chars */

The function next_char_min used to calculate the offset of the character to follow is introduced in Listing 24.13.

```
Listing 24.13 The next_char_min Function
```

```
static int next_char_min( GXTextPtr text, char c )
1:
2: {
3:
      int i, minx = 9999, num pts, nsegs = 0, coff = 0;
4:
5:
      int c = c - ' ';
6:
7:
      nsegs = 0;
8:
9:
      while( text->font[(int)_c][nsegs] != NULL ) {
10:
        num_pts = text->fontp[(int)_c][nsegs];
11:
12:
        for( i = 0; i < num pts; i++ ) {</pre>
13:
          minx = min(text->font[(int)_c][nsegs][i].x, minx);
14.
        }
15:
        nsegs++;
      }
16:
17:
18:
      /* approx fixed with of a character */
19:
     /* will be used to represent space */
20:
     if( c == ' ' )
        coff = 12;
21:
22:
      else if (minx < 0)
23:
        coff = abs(minx);
24:
25:
      return coff;
26: }
```

The next_char_min function looks at the points defining a character within the font set and determines the minimum x point for the character cell:

13: minx = min(text->font[(int)_c][nsegs][i].x, minx);

Of course, the points for all segments defining the character are considered

```
15: nsegs++;
```

to determine the amount to advance the transformation of points placed into the vpts array to avoid character overlap.

Two special cases are considered by the next_char_min function. First is the occurrence of a space character

20: if(c == ' ')

because a space should be the same width despite the character following. And, second, the formation of the character definitions centered on origin resulting in a negative minimum value:

22: else if (minx < 0)



Figure 24.2 is a graphical depiction of character definitions being centered on the origin and clearly demonstrates the need to account for a negative minimum value.



The second function employed by the reset_pts function in Listing 24.11 is the apply_scale function

```
6: apply_scale( text, dx, dy );
```

which can be found in Listing 24.14.

```
Listing 24.14 The apply_scale Function
```

```
1:
    static void apply_scale( GXTextPtr text,
                             int dx, int dy )
2:
3:
   {
4:
      char *txt = text->text;
5:
6:
                                 /* for transformation */
      int x, y;
7:
      int i, c, chr, nsegs, num pts; /* for traversing */
8:
      int minx, miny, maxx, maxy, width, height;
9:
10:
      get_extents( text, &minx, &miny, &maxx, &maxy );
11:
12:
      width = maxx - minx;
13:
     height = maxy - miny;
14:
15: x = text - x - minx;
16:
     y = text -> y - miny;
```

Listing 24.14 Continued

```
17:
18:
      for( c = 0; c < text->len; c++, txt++ ) {
        chr = *txt - ' ';
19:
20:
21:
        nsegs = 0;
22:
        while( text->font[chr][nsegs] != NULL ) {
23:
          num pts = text->fontp[chr][nsegs];
24:
          for( i = 0; i < num pts; i++ ) {</pre>
25:
            text->vpts[c][nsegs][i].x += (int)(dx *
26:
                 ((float)(text->vpts[c][nsegs][i].x -
27:
28:
                                minx)/(float)width)) + x;
29:
30:
            text->vpts[c][nsegs][i].y += (int)(dy *
31:
                 ((float)(text->vpts[c][nsegs][i].y -
32:
                               miny)/(float)height)) + y;
33:
          }
34:
          nsegs++;
35:
        }
36:
      }
37: }
```

The apply_scale function is critical for ensuring that the integrity of the Text object is maintained during the many calculations preformed continually on the points contained in the vpts array in the GXText structure.

The role it serves is to enable the user to scale the Text object and, unlike other objects in the editor, have the Text object revert periodically to the original points transformed from the character definitions in the font set (accomplished by the reset_font_pts function). After being reset to the original font points (transformed), the retention of the dx and dy values (scale deltas for the x and y-axis) is reapplied without the user even knowing.

The integrity (quality) of the points defining the Text object displayed on the canvas is critical because many of the line segments within a character definition for the more complex or fancier vector font sets are quite small. An error as slight as one pixel will have a detrimental effect on the appearance of the vector text.

The apply_scale function traverses the characters to access the segment and point arrays defined by the font set, much the same way as other functions introduced already. Important to the purpose of the apply_scale function, however, is the application of the dx and dy values *proportionally* to the x and y values stored in the vpts array:

```
29:
30: text->vpts[c][nsegs][i].y += (int)(dy *
31: ((float)(text->vpts[c][nsegs][i].y -
32: miny)/(float)height)) + y;
```

The term *proportionally* refers to the fact that the application of the scale factor is weighted based on the proximity of the point to the extent of the object.

For this reason, the extents of the object are determined early in the function

10: get_extents(text, &minx, &miny, &maxx, &maxy);

and from the extents the width and height values can be determined.

The get_extents function is found in Listing 24.15.

Listing 24.15 The get_extents Function

```
1:
    static void get extents( GXTextPtr text,
2:
                              int *minx, int *miny,
3:
                              int *maxx, int *maxy )
4:
   {
5:
      char *txt = text->text;
6:
      int c, chr, nsegs, x1, y1, x2, y2;
7:
8:
      *minx = *miny = 9999;
9:
      *maxx = *maxy = -9999;
10:
      for( c = 0; c < text->len; c ++, txt ++ ) {
11:
        chr = *txt - ' ';
12:
13:
14:
        nsegs = 0;
15:
        while( text->font[chr][nsegs] != NULL ) {
16:
17:
          get bounds( text->vpts[c][nsegs],
18:
                      text->fontp[chr][nsegs],
19:
                      &x1, &y1, &x2, &y2 );
20:
21:
          *minx = min(x1, *minx);
22:
          *miny = min( y1, *miny );
          *maxx = max(x2, *maxx);
23:
24:
          *maxy = max( y2, *maxy );
25:
26:
          nsegs++;
27:
        }
28:
      }
29: }
```

The get_extents function is, perhaps, the most straightforward of any function introduced so far.

It, too, traverses the segments defined by the font set to represent each character of the GXText text field to obtain the bounds of the points contained in the segment:

```
17: get_bounds( text->vpts[c][nsegs],
18: text->fontp[chr][nsegs],
19: & x1, &y1, &x2, &y2 );
```

The bounds of each segment are used to keep track of the minimum and maximum points used to represent the object:

```
21: *minx = min( x1, *minx );
22: *miny = min( y1, *miny );
23: *maxx = max( x2, *maxx );
24: *maxy = max( y2, *maxy );
```

These minimums and maximums reflect the extents of the object.

The get_bounds function used by the get_extents function is located in Listing 24.16.

```
Listing 24.16 The get_bounds Function
```

```
1:
    void get_bounds( XPoint *pts, int num_pts,
2:
                      int *x1, int *y1, int *x2, int *y2 )
3: {
4:
      int i;
5:
      *x1 = *y1 = SHRT MAX;
6:
7:
      *x2 = *y2 = 0;
8:
      for( i = 0; i < num_pts; i++ ) {</pre>
9:
10:
        *x1 = min( pts[i].x, *x1 );
11:
        *y1 = min( pts[i].y, *y1 );
12:
13:
        *x2 = max( pts[i].x, *x2 );
14:
        *y2 = max( pts[i].y, *y2 );
15:
      }
16: }
```

The get_bounds function acts on a single array of XPoints and thus is suitable for use by the Line object as well.

For an array of points, the minimum and maximum points contained in the array are returned to the calling function:

```
9: for( i = 0; i < num_pts; i++ ) {
10: *x1 = min( pts[i].x, *x1 );
11: *y1 = min( pts[i].y, *y1 );
13: *x2 = max( pts[i].x, *x2 );
14: *y2 = max( pts[i].y, *y2 );
15: }</pre>
```

We started our winding way to this function by waiting for the user to press the mouse button and position the creation_text, shown in Listing 24.2. When she finally does, the temporary GXText structure referenced by rubber_text is used to create a Text object:

19: create_text(NULL, rubber_text);

The create_text function is defined in Listing 27.17.

Listing 24.17 The create_text Function

```
static void create text( GXObjPtr obj, GXTextPtr text )
1:
2:
   {
3:
      /*
4:
      * create the template object
5:
      */
6:
     GXObjPtr obj = _obj;
7:
8:
     if( obj == NULL ) {
9:
          obj = gx_create_obj();
10:
     }
11:
12:
     obj->data = copy_gxtext( text, 0, 0 );
13:
14:
     obj->draw
                  = text draw;
     obj->erase = text_erase;
15:
16:
     obj->find = text_find;
     obj->move = text move;
17:
     obj->scale = text scale;
18:
19:
20:
21:
     obj->copy = text_copy;
     obj->select = text_select;
22:
23:
     obj->deselect = text_deselect;
24:
25:
     obj->save = text_save;
26:
27:
     gx_add_obj( obj );
28: }
```

The parameter list of the create_text function

1: static void create_text(GXObjPtr _obj, GXTextPtr text)

accounts for an invocation of the procedure with the caller having already created the common object to contain the text data. This supports the restoring of the object from a saved file, as is seen later in the chapter.

During interactive creation, however, the function create_text is called with a NULL value as the first parameter forcing the function to create the common object portion through a call to gx_create_obj.

→This is introduced in Chapter 17, in the section "Common Object Creation," page 343:

```
9: obj = gx_create_obj();
```

The creation routine then copies the GXText structure defining the object

12: obj->data = copy_gxtext(text, 0, 0);

following which the object methods and manipulation functions for controlling the object-specific data structure in lines 14–25 are assigned

```
14:
     obj->draw
                  = text_draw;
15:
     obj->erase = text erase;
    obj->find = text_find;
16:
     obj->move = text move;
17:
     obj->scale
                  = text_scale;
18:
19:
20:
                  = text_copy;
21:
    obj->copy
22:
    obj->select = text select;
23:
     obj->deselect = text deselect;
24:
25:
     obj->save = text save;
```

after which the object is added to the list of objects managed by the editor application:

27: gx_add_obj(obj);

Listing 24.18 shows the contents of the copy_gxtext function used by the create_text function.

```
Listing 24.18 The copy_gxtext Function
```

```
static GXTextPtr copy_gxtext(GXTextPtr text, int off_x, int off_y)
1:
2: {
3:
      GXTextPtr text data;
4:
     int
                i, c, chr, num pts, nsegs;
                *txt;
5:
     char
6:
7:
     text_data =
        create gxtext( text->text, text->x + off_x, text->y + off_y,
8:
9:
                       text->font, text->fontp );
10:
11:
     text data->dx = text->dx;
12:
     text_data->dy = text->dy;
13:
14:
     txt = text_data->text;
     for( c = 0; c < text_data->len; c++, txt++ ) {
15:
       chr = *txt - ' ';
16:
17:
18:
       nsegs = 0;
19:
       while( text data->font[chr][nsegs] != NULL ) {
```

```
20:
21:
          num_pts = text_data->fontp[chr][nsegs];
22:
23:
          for( i = 0; i < num pts; i++ ) {</pre>
24:
25:
            text data->vpts[c][nsegs][i].x =
26:
               text->vpts[c][nsegs][i].x + off x;
27:
            text_data->vpts[c][nsegs][i].y =
28:
               text->vpts[c][nsegs][i].y + off y;
29:
          }
30:
          nsegs++;
31:
        }
32:
      }
33:
34:
      return text data;
35: }
```

The copy_gxtext function begins by creating a new GXText structure with the origin of the Text structure incremented by the offset specified for the x and y components:

```
7: text_data =
8: create_gxtext( text->text, text->x + off_x, text->y + off_y,
9: text->font, text->fontp );
```

The function then traverses all points contained in the vpts array and increments them accordingly as well

25:	text_data->vpts[c][nsegs][i].x =
26:	<pre>text->vpts[c][nsegs][i].x + off_x;</pre>
27:	text_data->vpts[c][nsegs][i].y =
28:	<pre>text->vpts[c][nsegs][i].y + off_y;</pre>

finally returning the copy of the GXText structure with the offset values:

```
34: return text_data;
```

The create_text function specifies **0** for the offset value of the copy created:

12: obj->data = copy_gxtext(text, 0, 0);

However, this same function will be used in the definition of the text_copy method.

A created object is no longer drawn using the interactive rubber-banding GC, but by invoking the draw method of the object.

The following section introduces the methods that draw and erase the Text object from the drawing area canvas.

Drawing and Erasing a Text Object

The act of drawing or erasing an object in the Graphics Editor differs only in the treatment of the tile field of the GC created for the request.

Notice in Listing 24.19, lines 17 and 25, that the draw and erase methods differ only in the value passed for the tile flag required as the second parameter to the draw_erase function. (Note as well that the erase method is responsible for removing the object's handles if present on the screen).

Listing 24.19 Drawing and Erasing a Text Object

```
static void draw erase( GXObjPtr text, Boolean tile )
1:
2:
   {
3:
      GC gc;
4:
5:
      gc = gx_allocate_gc( text, tile );
6:
      GXDrawText( (GXTextPtr)text->data, gc );
7:
8:
      XtReleaseGC( GxDrawArea, gc );
9: }
10:
12: /*
13: * text draw
    */
14:
15: static void text_draw( GXObjPtr obj )
16: {
17:
      draw_erase( obj, False );
18: }
19:
20: /*
    * text_erase
21:
22: */
23: static void text_erase( GXObjPtr obj )
24: {
25:
      draw erase( obj, True );
26: }
```

The draw_erase function begins by creating a Graphics Context for use in the GXDrawText function:

```
5: gc = gx_allocate_gc( text, tile );
```

→ Important to the creation of the Graphics Context is the specification of the tile flag to the gx_allocate_gc function introduced in Chapter 17, in the section "Creating a Graphics Context," page 347.



If you recall, this flag indicated whether the background Pixmap of the GxDrawArea was assigned as the value to the tile field of the GC created.

A tile value assigned to a Graphic Context causes an effective erase action to occur because the pixels from the background Pixmap are placed where otherwise the foreground value of the GC would be placed, making the underlying background the result of the XDrawLines (or any X Graphic Primitive) request.

An appropriately created GC created for the current draw or erase action, the draw_erase function can request that the object be updated in the canvas window:

6: GXDrawText((GXTextPtr)text->data, gc);

Because the X Server will attempt to cache the GC for future requests, it is important to specify our complete use of it with a call to XtReleaseGC:

8: XtReleaseGC(GxDrawArea, gc);

With the object visible on the screen, it is now eligible for manipulation by the user. However, before it can be moved, scaled, or deleted, it must be selected by the user.

The next section introduces the Text object find method used to determine whether an event has successfully located the object on the drawing area.

Finding a Text Object

The text_find method shown in Listing 24.20 borrows greatly from the support functions created for the line_find method in Chapter 20, in the section "Finding a Line Object," page 390.

```
Listing 24.20 The text_find Function
```

```
1: static Boolean text find( GXObjPtr obj, XEvent *event )
2: {
      GXTextPtr text = obj->data;
3:
4:
5:
      char *txt = text->text;
6:
7:
      int i, c, chr, nsegs, num pts, found = False;
8:
9:
      for( c = 0; (c < text -> len) && !found; c++, txt++ ) {
10:
        chr = *txt - ' ';
11:
12:
        nsegs = 0;
        while((text->font[chr][nsegs] != NULL) && !found) {
13:
14:
          num pts = text->fontp[chr][nsegs];
15:
          for(i = 0; (i < (num pts - 1)) && !found; i++) {</pre>
16:
            found =
17:
                near segment( text->vpts[c][nsegs][i ].x,
18:
19:
                               text->vpts[c][nsegs][i ].y,
20:
                               text->vpts[c][nsegs][i+1].x,
21:
                               text->vpts[c][nsegs][i+1].y,
22:
                               event->xbutton.x,
23:
                               event->xbutton.y );
24:
          }
25:
          nsegs++;
```

Listing 24.20 Continued

```
26: }
27: }
28:
29: return found;
30: }
```

The text_find method parses the GXText structure to reduce it to arrays of points compatible with the near_segment function found in Chapter 10, in the section "Calculating Point and Line Intersections," page 211.

If the event point specified to the text_find method intersects any of the many segments composing the Text object, the loops are interrupted and the True value is returned to the calling function.

The text_find method serves two purposes in the management of the Text object. One purpose is to enable the user to select the object for manipulation. A second purpose is to determine whether the manipulation requested by the user is the move action.

The following section introduces the step stemming from the text_find method resulting in the selecting or deselecting of the object.

Selecting and Deselecting a Text Object

If the event passed to the text_find method discussed in the previous section results in the location of an object that is not currently selected, the object is made active by invoking its select method.

The select method for the Text object, as seen in Listing 24.21, ensures that handles are created and drawn for the object.

Listing 24.21 The text_select Function

```
1: static void text_select( GXObjPtr text )
2: {
3: text_bounding_handles( text );
4: gx_draw_handles( text );
5: }
```

→ The function gx_draw_handles was introduced in Chapter 16, section "Managing Object Handles," page 327.

The process of creating the handles for the Text object begins with a call to text_bounding_handles found in Listing 24.22.

```
Listing 24.22 Creating Handles for a Text Object
```

```
static void text bounding handles( GXObjPtr gx text )
1:
2:
   {
3:
      GXTextPtr text = gx text->data;
4:
5:
      int i, minx, miny, maxx, maxy, width, height;
6:
7:
      gx text->handles = (XRectangle *)
8:
                    XtMalloc( sizeof(XRectangle) * 8 );
9:
      gx_text->num_handles = 8;
10:
11:
     if( gx_text->handles == NULL ) {
12:
        perror( "Alloc failed for text handles" );
13:
        gx text->num handles = 0;
14:
        return;
15:
      }
16:
17:
      for( i = 0; i < 8; i++ ) {</pre>
18:
        gx text->handles[i].width = HNDL SIZE;
19:
        gx_text->handles[i].height = HNDL_SIZE;
20:
      }
21:
22:
      get_extents(text, &minx, &miny, &maxx, &maxy);
23:
      width = maxx - minx; height = maxy - miny;
24:
25:
      gx text->handles[0].x =
26:
            minx - HNDL OFFSET - HNDL SIZE;
27:
      gx text->handles[0].y =
28:
            miny - HNDL OFFSET - HNDL SIZE;
29:
30:
      gx text->handles[1].x =
31:
            minx + (width/2) - HNDL OFFSET;
32:
      gx text->handles[1].y =
33:
            miny - HNDL_SIZE - HNDL_OFFSET;
34:
35:
      gx_text->handles[2].x =
36:
            maxx + HNDL_OFFSET;
37:
38:
      gx_text->handles[2].y =
39:
            miny - HNDL_SIZE - HNDL_OFFSET;
40:
41:
      gx text->handles[3].x = maxx + HNDL OFFSET;
42:
      gx text->handles[3].y =
43:
            miny + (height/2) - HNDL_OFFSET;
44:
45:
      gx text->handles[4].x = maxx + HNDL OFFSET;
46:
      gx text->handles[4].y = maxy + HNDL OFFSET;
47:
48:
      gx text->handles[5].x =
49:
            minx + (width/2) - HNDL_OFFSET;
```

Listing 24.22 Continued

```
50:
      gx text->handles[5].y = maxy + HNDL OFFSET;
51:
52:
      qx text->handles[6].x =
53:
            minx - HNDL OFFSET - HNDL SIZE;
54:
      gx text->handles[6].y = maxy + HNDL OFFSET;
55:
56:
      gx_text->handles[7].x =
57:
            minx - HNDL_OFFSET - HNDL_SIZE;
58:
      gx text->handles[7].y =
            miny + (height/2) - HNDL OFFSET;
59:
60: }
```

The creation of the Text object handles in text_bounding_handles begins by ensuring that the correct number of elements are created in the handles array and the correct handle count is assigned the num_handles field of the common object structure:

```
7: gx_text->handles = (XRectangle *)
8: XtMalloc( sizeof(XRectangle) * 8 );
9: gx_text->num_handles = 8;
```

After testing for a failure of the allocation routine, the widths of the handles for the Text object are assigned

```
17: for( i = 0; i < 8; i++ ) {
18: gx_text->handles[i].width = HNDL_SIZE;
19: gx_text->handles[i].height = HNDL_SIZE;
20: }
```

and then the extents of the Text object are obtained in order to determine the placement of the handles:

```
22: get_extents(text, &minx, &miny, &maxx, &maxy);
23: width = maxx - minx; height = maxy - miny;
```

Last comes the tedious task of placing each of the eight handles at the appropriate location around the object in lines 25–59.



The position of the handles runs clockwise, starting with handles[0] located in the upper-right corner of the object's bounds.

The Text object's deselect method introduced in Listing 24.23 shows the steps necessary to remove the handles, indicating the active state of the Text object.

```
Listing 24.23 Deselecting a Text Object
```

```
1: static void text deselect( GXObjPtr text )
2: {
3:
      if( text->handles && text->num handles > 0 ) {
4:
        gx_erase_handles( text );
5:
6:
        XtFree((char *)text->handles );
7:
8:
        text->handles = NULL;
9:
        text->num handles = 0;
10:
      }
11: }
```

The function is simply responsible for erasing the handles

4: gx_erase_handles(text);

freeing the memory associated with them

```
6: XtFree((char *)text->handles );
```

and resetting the values of the variables:

```
8: text->handles = NULL;
9: text->num_handles = 0;
```

The next section provides the functionality of interactively moving a Text object.

Moving a Text Object

The steps required to relocate a Text object on the canvas are provided by the Text_move method seen in Listing 24.24.

```
Listing 24.24 The text move Function
```

```
1: static void text move( GXObjPtr text, XEvent *event )
2: {
3:
      static int x = 0, y = 0;
4:
5:
      if( x && y ) {
6:
        GXDrawText( (GXTextPtr)text->data, rubberGC );
7:
      } else {
8:
        (*text->erase)( text );
9:
10:
        x = event ? event -> xbutton.x : 0;
11:
        y = event ? event->xbutton.y : 0;
12:
      }
13:
14:
      if( event ) {
15:
        apply delta( text->data,
16:
                     event->xbutton.x - x,
17:
                     event->xbutton.y - y );
```

Listing 24.24 Continued

```
18:

19: GXDrawText( text->data, rubberGC );

20: x = event->xbutton.x;

21: y = event->xbutton.y;

22: } else {

23: x = 0;

24: y = 0;

25: }

26: }
```

The text_move function uses the static points x and y to track consecutive calls to the function for a single move operation:

5: if(x && y) {

If x and y are non-zero, the text_move function knows to erase a previously draw rubber-banding copy of the Text object:

```
6: GXDrawText( (GXTextPtr)text->data, rubberGC );
```

Otherwise, the actual Text object is erased from the canvas

```
8: (*text->erase)( text );
```

and the values of x and y are assigned:

```
10: x = event ? event->xbutton.x : 0;
11: y = event ? event->xbutton.y : 0;
```

In the presence of a valid event

```
14: if( event ) {
```

the apply_delta function is used to reposition the interactive Text object

the GXText structure with the updated points is drawn using the rubber-banding GC

19: GXDrawText(text->data, rubberGC);

and the current event point is retained in the static variables:

```
20: x = event->xbutton.x;
21: y = event->xbutton.y;
```

If a NULL event reference is passed to the function, the x and y values are reset to 0 and the current move action is cancelled:

23: x = 0; 24: y = 0; Listing 24.25 shows the definition of the apply_delta function used by the text_move method.

```
Listing 24.25 The apply_delta Function
```

```
static void apply_delta( GXTextPtr text,
1:
2:
                              int dx, int dy )
3: {
4:
      char *txt = text->text;
5:
      int i, c, chr, nsegs, num pts;
6:
7:
      text->x += dx;
8:
      text->y += dy;
9:
10:
      for( c = 0; c < text->len; c++, txt++ ) {
        chr = *txt - ' ';
11:
12:
13:
        nsegs = 0;
14:
        while( text->font[chr][nsegs] != NULL ) {
          num_pts = text->fontp[chr][nsegs];
15:
16:
17:
          for( i = 0; i < num pts; i++ ) {</pre>
18:
19:
            text->vpts[c][nsegs][i].x += dx;
20:
            text->vpts[c][nsegs][i].y += dy;
21:
          }
22:
          nsegs++;
23:
        }
24:
      }
25: }
```

The apply_delta function is a pleasant reprieve from the complex support function introduced in this chapter. By now, you should be comfortable with traversing the GXText structure using the characters of the text field to index into the font set to find the segments and access the points.

The apply_delta simply makes this traversal to access the points composing each character and increments them by the delta values specified for the x and y components accomplishing a move for the Text object.

The following section addresses the steps required to scale the Text object.

Scaling a Text Object

The scale method of an object, as introduced in Chapter 16, in the section "Processing User Navigation of Objects," page 334, is assigned as the active action when the user selects one of the active object's handles.

The text_scale method introduced in Listing 24.26 manages the scale action for the Graphics Editor Text object.

Listing 24.26 The text_scale Function

```
static void text_scale( GXObjPtr text, XEvent *event )
1:
2: {
3:
      static int x = 0, y = 0;
4:
5:
      if(x && y ) {
        GXDrawText( text->data, rubberGC );
6:
7:
8:
      } else {
9:
        (*text->erase)( text );
10:
11:
       x = event ? event->xbutton.x : 0;
12:
        y = event ? event->xbutton.y : 0;
13:
      }
14:
     if( event ) {
15:
        calc apply scale( text->data,
16:
17:
             event->xbutton.x - x, event->xbutton.y - y );
18:
       x = event - xbutton.x;
19:
        y = event->xbutton.y;
20:
21:
        GXDrawText( text->data, rubberGC );
22:
      } else {
23:
        x = 0;
24:
        y = 0;
25:
      }
26: }
```

The text_scale method of the Text object is responsible for managing the event references passed to the function. Identical in structure to scale methods for previously introduced objects, focus on the portion of this function that is unique to the Text object:

```
16: calc_apply_scale( text->data,
17: event->xbutton.x - x, event->xbutton.y - y );
18: x = event->xbutton.x;
19: y = event->xbutton.y;
```

Listing 24.27 shows the calc_apply_scale function used by the text_scale method to accomplish the scale request.

Listing 24.27 The calc_apply_scale Function

```
5:
      case 0:
6:
        apply_scale_top ( text, dx, dy );
7:
        apply scale left( text, dx, dy );
8:
        break;
9:
10:
     case 1:
11:
        apply_scale_top( text, dx, dy );
12:
        break:
13:
14:
      case 2:
15:
        apply scale top ( text, dx, dy );
        apply_scale_right( text, dx, dy );
16:
17:
        break;
18.
19:
      case 3:
20:
        apply scale right( text, dx, dy );
21:
        break;
22:
23:
      case 4:
24:
        apply_scale_right ( text, dx, dy );
25:
        apply_scale_bottom( text, dx, dy );
26:
        break;
27:
28:
      case 5:
29:
        apply scale bottom( text, dx, dy );
30:
        break;
31:
32:
     case 6:
33:
        apply scale bottom( text, dx, dy );
        apply scale left ( text, dx, dy );
34:
        break;
35:
36:
37:
      case 7:
38:
        apply_scale_left( text, dx, dy );
39:
        break;
40:
41:
      default:
42:
        setStatus( "TEXT: The end is nigh!" );
43:
      }
44:
45:
      /* resetting to the original points ensures we don't */
46:
      /* compound rounding errors for points that have
                                                             */
47:
      /* already been scaled
                                                             */
      reset pts( text, text->x, text->y, text->dx, text->dy );
48:
49: }
```

The function switches on the GxActiveHandle value set at the start of the scale action

```
4: switch( GxActiveHandle ) {
```

to determine which direction the scale action should be applied. Then one of the several apply_scale_<direction> functions found in Listing 24.28 is invoked to adjust the x, y, dx, or dy fields of the GXText structure appropriate for the direction the object is being scaled.

Listing 24.28 The Support Functions Scaling a Text Object

```
static void apply scale top( GXTextPtr text,
1:
2:
                                 int dx, int dy )
3:
   {
4:
     text->y += dy;
5:
    text->dy += (dy * -1);
6: }
7: static void apply scale left( GXTextPtr text,
8:
                                 int dx, int dy )
9: {
10: text->x += dx;
11: text->dx += (dx * -1);
12: }
13: static void apply scale right( GXTextPtr text,
14:
                                  int dx, int dy )
15: {
16:
     text->dx += dx;
17: }
18: static void apply scale bottom( GXTextPtr text,
19:
                                    int dx, int dy )
20: {
21:
     text->dy += dy;
22: }
```

Following the adjustment of the necessary fields within the GXText structure, the text_scale function invokes the reset_pts function to do the work:

48: reset_pts(text, text->x, text->y, text->dx, text->dy);

The call to reset_pts ensures that no rounding errors from previous actions have adversely affected the quality of the Text object being displayed.

The critical support functions required by the text_copy feature introduced in the next section have already been seen, but let's go there anyway.

Copying a Text Object

To accomplish the text_copy method seen in Listing 24.29 for duplicating the Text object, you need only apply functions already introduced in this chapter.

```
Listing 24.29 The text_copy Function
```

```
1: static void text_copy( GXObjPtr obj )
2: {
3: GXTextPtr temp_data;
4:
5: (*obj->deselect)( obj );
6:
```

By first deselecting the active Text object, the handles associated with it are removed from the screen

```
5: (*obj->deselect)( obj );
```

following which the copy_gxtext function can be employed, specifying the value of OFFSET to prevent the copy from exactly overlaying the original object:

```
7: temp_data = copy_gxtext( (GXTextPtr)obj->data,
8: 0FFSET, 0FFSET );
```

The temporary GXText structure is then used in a call to create_text

```
9: create_text( NULL, temp_data );
```

and the memory associated with the temporary structure is freed:

```
11: freeGXText(temp_data);
12: XtFree((char *)temp_data);
```

The following section presents the last area of functionality required by the Text object—saving and restoring the object-specific data.

Saving and Restoring a Text Object

The ability to save and restore objects contained in the Graphics Editor, as introduced in Chapter 19, provides a mature level of capability to our application.

The text_save method found in Listing 24.30 adheres to the save strategy introduced earlier.

```
Listing 24.30 The text_save Function
```

```
1: static void text_save( FILE *fp, GXObjPtr obj )
2: {
3: GXTextPtr text = (GXTextPtr)obj->data;
4:
5: fprintf( fp, "TEXT [ str x y ]\n" );
6: fprintf( fp, "%s\n %d %d\n",
7: text->text, text->x, text->y );
8: }
```

The text_save function extracts the GXText structure from the common object structure

```
3: GXTextPtr text = (GXTextPtr)obj->data;
```

writes a simple tag line showing the format of the data to follow

5: fprintf(fp, "TEXT [str x y]\n");

and then writes the data

```
6: fprintf( fp, "%s\n %d %d\n",
7: text->text, text->x, text->y );
```

which consists of simply the location of the object and text string associated with it.

Notice the newline character inserted after the %s format token in the data line. Because it is possible for the user to embed spaces in the text value, we must ensure that the restore function can distinguish completely the text string from the location.

Listing 24.31 shows how the data written by the text_save method is retrieved to the editor.

```
Listing 24.31 The GXTextLoad Function
```

```
void gxTextLoad( FILE *fp, GXObjPtr obj )
1:
2:
   {
3:
        char text[256], *ptr;
4:
5:
        GXTextPtr data;
6:
        int x, y;
7:
8:
        fgets( text, 256, fp );
        if( (ptr = strchr( text, '\n' )) != NULL ) {
9:
            *ptr = '\0';
10:
11:
        }
12:
        fscanf( fp, "%d %d\n", &x, &y );
13:
        data = create_gxtext( text, x, y,
14:
15:
                               plain_simplex, plain_simplex_p );
16:
17:
        create_text( obj, data );
18: }
```

The GxLoadText function reads first the string using the C fgets function reads until it finds the newline character written by the text_save method

8: fgets(text, 256, fp);

and then strips the newline character from the string because it cannot be represented by a vector font set:

```
9: if( (ptr = strchr( text, '\n' )) != NULL ) {
10: *ptr = '\0';
11: }
```

After retrieving the object's location

12: fscanf(fp, "%d %d\n", &x, &y);

the restore function has the necessary components to create a GXText structure

```
14: data = create_gxtext( text, x, y,
15: plain_simplex, plain_simplex_p );
```

which it passes to the create_text function to create the Text object and add it to the list managed by the editor application:

```
17: create_text( obj, data );
```

With the completion of the discussion concerning saving and restoring the Text object, you have completed the addition of all objects supported by the editor application. Congratulations!

Next Steps

Chapter 25, "Introduction to PostScript," will introduce the PostScript programming language as we work our way closer to the next goal of adding print capability to the Graphics Editor application.



Adding a Print Driver

Chapter 25

In this chapter

- PostScript
- Learning PostScript
- PostScript Commands
- PostScript Programming
- Viewing PostScript Files
- Next Steps

Introduction to PostScript

The usefulness of the Graphics Editor is significantly advanced by the addition of print capability. There are several ways to approach the task of printing; therefore, I have selected a simple approach that easily allows for future enhancements.

To accomplish the Graphics Editor print function, the PostScript page description language will be used. PostScript enables us to communicate with a PostScript-compatible printer to effectively describe the page to be printed.

The following sections introduce the PostScript language that is used in future chapters to print the graphics objects drawn by the Graphics Editor.



This chapter covers only a portion of PostScript Level 1 and Level 2, which are the earlier versions of PostScript created by Adobe. The current version is Level 3.

PostScript

PostScript is a programming language that describes the appearance of a printed page. It was developed by Adobe in 1985 and has become an industry standard for printing and imaging.

All major printer manufacturers make printers that contain or can be loaded with PostScript interpreter software. A PostScript file can generally be identified by a ps suffix. Like C language source code files, PostScript files are plain text and can be viewed using your favorite editor.

PostScript is an interpreted, stack-oriented language for describing—in a deviceindependent fashion—the way in which pages can be composed of characters, shapes, and digitized images in black and white, grayscale, or color. Concerning the fields of text and graphics, PostScript is the most widely used printer controller in the industry. It gives computer users total control over text, graphics, color-separations, and halftones.

The page description language recognizes a page the user creates as a unit and converts the elements of the page into control data for the output device. The printing engine receives the control data in its own format and resolution. For the printed page the resolution can be up to 300 dots per inch (dpi).

PostScript language programs are used for communication between a software product such as the Graphics Editor and a printing system, and they enable the user to mix text, graphics, and images from various sources.

The following section introduces the PostScript language by drawing on what you already know about the C programming language.

Learning PostScript

The PostScript page description language serves as an interface between an application program and a printing system (usually a printer). A C program can generate the PostScript language code needed by a printer for rendering the pages described.

Using the C programming language classic "Hello World" example,

```
main()
{
    printf("hello world\n");
}
```

to print the character string hello world to the screen, we will begin our instruction in PostScript.

The translation of this C program into PostScript would look like the following:

```
1: %!
2: /Courier findfont 10 scalefont setfont
3: 0 100 moveto
4: (hello world) show
5: 0 88 moveto
6: showpage
```

At first glance, you don't see similarities between the C program sample and its PostScript translation. Because PostScript is a graphical programming language, you must specify how the character string will be printed. This requires specifying the typeface and point size:

/Courier findfont 10 scalefont setfont

Courier is the font that is specified with the findfont command, and 10 is the point size PostScript is told to scale the font to with the scalefont command. The last keyword, setfont, instructs the PostScript interpreter to make the font active.

You next must specify the origin for placing the string:

0 100 moveto

This instruction moves 0 points to the right and 100 points up from the lower-left corner of the paper sheet.

Next, you designate the line feed by a vertical motion downward from the previous location:

```
0 88 moveto
```

Finally, you instruct the interpreter to display the page that's been described:

showpage

Notice the order of the sample PostScript commands relative to the parameters required by the command. As with the moveto command

0 88 moveto

the values 0 and 88 are the x and y coordinates needed by the command, but they precede moveto. This syntax is necessary because the PostScript language is stack based.

Stacks

A *stack* is a data structure implemented as a last in, first out (LIFO) list. On a stack, the last item added to the structure is the first item removed.

Everything read by PostScript is placed (pushed) on a stack internal to the interpreter. When the interpreter reads a command or instruction requiring arguments, the appropriate number of parameters are popped (removed) from the stack to satisfy the command.

Several stacks are in a PostScript system, but only two are important for this discussion: the *operand stack* and the *dictionary stack*.

The operand stack is a stack with arguments to procedures (or operators, using PostScript vernacular) that are pushed prior to use. The dictionary stack is, as the name implies, for dictionaries, and it provides storage for variables.

Looking again at the moveto command, envision a list where everything read by the interpreter is placed. In the end, the list would appear as follows:
2. moveto 1. 88 0. 0

When the moveto command is reached, the elements placed on the list are removed in the reverse order they were inserted.

The following section introduces the commands needed by the Graphics Editor for printing the objects drawn on the canvas.

PostScript Commands

PostScript is a highly capable language. The following sections discuss many PostScript operators used by the editor application and provide a description for each.

Comments

Inserting comments into a PostScript file is done using the percent (%) character. As with other languages, anything following on the same line as a comment token is ignored by the interpreter.

The special comment %! token used as the first two characters of a PostScript program is seen as a tag marking the file as PostScript code by many systems, including the UNIX 1pr command.

It is a good idea to start every PostScript document with the %! token, because it ensures that every spooler and printer the document can encounter recognizes it as PostScript code.

In addition to the information found in the comment specifiers, PostScript understands numerous commands. The following section introduces PostScript conventions for forming the commands used by the Graphics Editor for printing the canvas window.

PostScript Programming

Programming in PostScript is quite easy. The fundamentals are that you *push* operands onto the operand stack by naming them, and then you invoke the operands to employ them.

The challenge in programming with PostScript is in knowing which operand to use, and when.

Operators to draw and put text on the screen make up the bulk of PostScript operands. A couple of operands, however, are used for maintaining the program itself.

The first of these operators is the def operand, which is responsible for entering a definition into the top-most dictionary on the dictionary stack.



A *dictionary* is a collection of name-value pairs. All named variables are stored in dictionaries. In addition, all available operators are stored in dictionaries along with their code. The dictionary stack is a stack of all currently open dictionaries. When a program refers to some key, the interpreter wanders down the stack looking for the first instance of that key in a dictionary. In this manner, names can be associated with variables and a simple form of managing scope is implemented. Conveniently, dictionaries can be given names and can be stored in other dictionaries.

The top operand on the operand stack is the value, and the operand below the value is the key and is typically a name.

For instance, defining the name x and assigning it a value of 5 in PostScript would be done as follows:

/x 5 def

Notice the use of the forward slash before the x. This ensures that the *name* x, and not any value it might represent, will be pushed onto the operand stack in any defined dictionary stack.

The def operand is also used to define new operators. The value in this case is just a procedure. The following code defines an operator foo, which adds its top-most two operands and multiplies the result with the next operand on the stack:

/foo {add mul} def

Remember that operands, which return results, *push* them onto the stack so they can be used later.

An important point to know when defining procedures is that the elements in a procedure are not evaluated until the procedure is invoked.

That means that in the procedure

```
{1 2 add 3 mul}
```

the actual names add and mul are stored in the procedure array. This is different from an actual array in which the components are evaluated when the array is created.

This delayed evaluation of procedure components has two important effects. First, the definition of an operator used in a procedure is the one that is in effect when the procedure is run, not when it is defined. Second, because each operator must be looked up each time the procedure is invoked, things can be a little slow.

Fortunately, PostScript provides the bind operator to replace each name in a procedure object with its current definition:

/foo {add mul} bind def

If add or mul is redefined after defining foo, foo will have the same behavior as before. Without the use of bind, the behavior of foo would change. Comfortable with the basic syntax forming a PostScript operator, we will look at a summary of some of the commands understood by PostScript.

Table 25.1 shows many of the commands provided with Level 1 PostScript.

	-	
Operator	Example	Description
add	num1 num2 add num3	Returns the addition of the two arguments.
arc	x-coord y-coord r ang1 ang2 arc	Adds an arc to the current path. The arc is generated by sweeping a line segment of length r , and tied at the point (x-coord y-coord), in a counter-clockwise direc- tion from an angle ang1 to an angle ang2. Note: A straight- line segment will connect the current point to the first point of the arc if they are not the same.
begin	dict begin	Pushes the dictionary dict onto the dictionary stack, where it can be used for def and name lookup. This oper- ator enables an operator to set up a dictionary for its own use (for example, for local variables).
bind	procedure1 bind procedure2	Goes through procedure1 and replaces any operator names with their associate operators. Names that do not refer to operators are left alone. Operators within procedure1 that have unrestricted access will have bind called on themselves before they are inserted into the pro- cedure. The new procedure, with operators instead of oper- ator names, is returned on the stack as procedure2. The main effect and use of this operator is to reduce the amount of name lookup done by the interpreter. This speeds up execution and ties down the behavior of operators.
clip	clip	Intersects the current clipping path with the current path and sets the current clipping path to the results. Any part of a path drawn after calling this operator that extends outside this new clipping area will simply not be drawn. If the given path is open, clip will treat it as if it were closed. Also, clip does not destroy the current path when it is finished; it can be used for other activities. It is impor- tant to note that there is no easy way to restore the clip path to a larger size after it has been set. The best way to set the clip path is to wrap it in a gsave and grestore pair.

Table 25.1 PostScript Level 1 Commands

Operator	Example	Description
closepath	closepath	Adds a line segment to the current path from the current point to the first point in the path. This closes the path so it can be filled.
charpath	string bool charpath	Takes the given string and appends the path, which the characters define to the current path. The result can be used as any other path for stroking, filling, or clipping. The Boolean argument informs charpath what to do if the font is not designed to be stroked. If the Boolean is true, the path will be modified to be filled and clipped (but not stroked). If the Boolean is false, the path will be suitable to be stroked (but not filled or clipped).
curveto	x1 y1 x2 y2 x3 y3 curveto	Draws a curve from the current point to the point $(x3, y3)$ using points $(x1, y1)$ and $(x2, y2)$ as control points. The curve is a Bézier cubic curve. In such a curve, the tangent of the curve at the current point will be a line segment running from the current point to $(x1, y1)$, and the tangent at $(x3, y3)$ is the line running from $(x3, y3)$ to $(x2, y2)$.
def	name value def	Associates the name with a value in the dictionary at the top of the dictionary stack. This operator essentially defines names to have values in the dictionary and is used to define variables and operators.
div	num1 num2 div num3	Returns the result of dividing num1 by num2. The result is always a real.
dup	object dup object object	Pushes a second copy of the topmost object on the operand stack. If the object is a reference to an array, string, or similar composite object, only the reference is duplicated; both references will still refer to the same object.
end	end	Pops the topmost dictionary from the dictionary stack. The dictionary below it becomes the new current dictionary.
exch	value1 value2 exch value2 value1	Simply exchanges the top two items on the operand stack. It does not matter what the operands are.
fill	fill	Closes and fills the current path with the current color. Any ink within the path is obliterated. Note that fill blanks out the current path as if it had called newpath. If you want the current path preserved, you should use gsave and grestore to preserve the path.
findfont	name findfont font	Looks for the named font in the font dictionary. If it finds the font, it pushes the font on the stack for later process- ing. It signals an error if the font cannot be found.

Anerator	Fxamnle	Description
Ομεταίοι	схатрь	Description
for	initial increment limit proc for	Executes proc repeatedly. The first time proc is executed, it will be given initial as the top operand. Each time it is executed after that, the top operand will be incremented by increment. This process will continue until the argument exceeds the limit.
grestore	grestore	Sets the current graphics state to the topmost graphics state on the graphics state stack and pops that state off the stack. This operator is usually used in conjunction with gsave.
gsave	gsave	Pushes a copy of the current graphics state onto the graphics state stack. The graphics state consists of (among other things): the current font and the current color.
if	bool proc if	Executes proc if bool is true.
ifelse	bool proc1proc2 ifelse	Executes proc1 if bool is true, and proc2 otherwise.
index	value_n value_ 0 n index value_ n value_0 value_n	Grabs the nth item off the operand stack (item is the one just under the index you push on the stack for the operator) and pushes it on top of the stack.
lineto	x-coord y-coord lineto	Adds a line into the path. The line is from the current point to the point (x-coord y-coord). After the line is added to the path, the current point is set to (x-coord y-coord). It is an error to call lineto without having a current point.
moveto	x-coord y-coord moveto	Moves the current point of the current path to the given point in user space. If a moveto operator immediately fol- lows another moveto operator, the previous one is erased.
mul	value1 value2 mul product	Multiplies the first two operands on the stack and pushes the result back onto the stack. The result is an integer if both operands are integers and the product is not out of range. If the product is too big or one of the operands is a real, the result will be a real.
newpath	newpath	Clears the current path and prepares the system to start a new current path. This operator should be called before starting any new path, although some operators call it implicitly.
рор	value pop	Removes the top-most item from the operand stack.
restore	state restore	Restores the total state of the PostScript system to the state saved in state.

Tab

Operator	Example	Description
rlineto	dx dy rlineto	Adds a line into the path. The line is from the current point to a point found by adding dx to the current x and dy to the current y. After the line is added to the path, the cur- rent point is set to the new point. It is an error to call lineto without having a current point.
rmoveto	dx dy rmoveto	Moves the current point of the current path by adding dx to the current x and dy to the current y.
rotate	angle rotate	Rotates the user space counter-clockwise by angle degrees (negative angles rotate clockwise). The rotation occurs around the current origin.
save	save state	Gathers the complete state of the PostScript system and saves it in state. Possible errors resulting from this call include limitcheck and stackoverflow.
scale	sx sy scale	Scales the user coordinates. Sx in the horizontal direction and sy in the vertical direction will multiply all coordinates. The origin will not be affected by this operation.
scalefont	font size scalefont font	Takes the given font and scales it by the given scale factor. The resulting scaled font is pushed onto the stack. A size of 1 produces the same sized characters as the original font, 0.5 produces half-size characters, and so on.
setfont	font setfont	Sets the current font to be font. This font can be the result of any font creation or modification operator. This font is used in all subsequent character operations like show.
setgray	gray-value setgray	Sets the current intensity of the ink to gray-value, which must be a number from 0 (black) to 1 (white). This will affect all markings stroked or filled onto the page. This applies even to path components created before the call to setgray, as long as they have not yet been stroked.
setlinewidth	width setlinewidth	Sets the width of all lines to be stroked to width, which must be specified in points. A line width of 0 is possible and is interpreted to be a hairline, as thin as can be ren- dered on the given device.
show	string show	Draws the given string onto the page. The current graph- ics state applies, so the current font, font size, gray value, and current transformation matrix all apply. The current point determines the location for the text. The current point will specify the leftmost point of the baseline for the text.

DIE 25.1 Continued					
Operator	Example	Description			
showpage	showpage	Commits the current page to print and ejects the page from the printing device. showpage also prepares a new blank page.			
stroke	stroke	Draws a line along the current path using the current set- tings. This includes the current line thickness, current pen color, current dash pattern, current settings for how lines should be joined, and what kind of caps they should have. These settings are the settings at the time the stroke operator is invoked. A closed path consisting of two or more points at the same location is a degenerate path. A degenerate path will be drawn only if you have set the line caps to round caps. If your line caps are not round caps, or if the path is not closed, the path will not be drawn. If the path is drawn, it will appear as a filled circle center at the point.			
sub	num1 num2 sub num3	Returns the result of subtracting num2 from num1.			
translate	x-coord y-coord translate	Moves the origin to the point (x-coord, y-coord) in the current user space.			

Table 25.1	Continued
------------	-----------

The file description page created using the PostScript commands listed in Table 25.1 can be viewed on your computer screen using a PostScript viewer utility.

Viewing PostScript Files

Although generally used to communicate with a print device, PostScript can be viewed on the computer monitor. This is especially useful when troubleshooting a print driver or learning PostScript language.

Ghostscript, a utility provided with Linux for viewing PostScript files, is invoked with the command gv.

Ghostscript is the name of a set of software that provides an interpreter for the PostScript language. It has the capability to convert PostScript language files to many raster formats, view them on displays, and print them to printers that don't have PostScript language capability built in.

The Ghostscript software is also able to interpret Portable Document Format (PDF) files and has the capability to convert PostScript language files to PDF (with some limitations) and vice versa.

Ghostscript provides a set of C procedures through the Ghostscript library that implement the graphics capabilities that appear as primitive operations in the PostScript language.

If you are not using Linux, or if the Ghostscript software is not installed on your system, you can obtain it from ftp://ftp.cs.wisc.edu/ghost.

Comments Understood by Ghostscript

The comments found in Listing 25.1 are recognized by the Ghostscript interpreter and are useful when forming the prolog of a page description file. There are no required elements when forming a page prolog; however, the more directives you employ the closer your description will be honored.



As described in the text, the prolog is the initial area of the PostScript file. The prolog, when it is used, contains any procedures that are used in the body of the document. These are surrounded by

%BeginProlog definitions go here %EndProlog

In keeping with the BeginProlog and EndProlog directives, the word *prolog* is often used in lieu of what might be grammatically more correct to refer to as *prologue*.

The terms, however, in the context of a PostScript document, are synonymous.

Listing 25.1 Comments Understood by PostScript

```
1: %!PS-Adobe-<real> [EPSF-<real>]
2: %&BoundingBox: <int> <int> <int> (atend)
3: %&CreationDate: <textline>
4: %&Orientation: Portrait|Landscape|(atend)
5: %&Pages: <uint>|(atend)
6: %&PageOrder: Ascend|Descend|Special|(atend)
7: %%Title: <textline>
8: %&DocumentMedia: <text> <real> <real> <real> <text> <text>
9: %&DocumentPageSizes: <text>
10: %&EndComments
11:
12: %&BeginPreview
13: %&EndPreview
14:
```

```
15: %%BeginDefaults
```

```
16: %%PageBoundingBox: <int> <int> <int> <int> |(atend)
```

25

continues

Listing 25.1 Continued

```
17: %%PageOrientation: Portrait|Landscape
18: %%PageMedia: <text>
19: %%EndDefaults
20:
21: %%BeginProlog
22: %%EndProlog
23:
24: %%BeginSetup
25: %%PageBoundingBox: <int> <int> <int> <int> |(atend)
26: %%PageOrientation: Portrait|Landscape
27: %%PaperSize: <text>
28: %%EndSetup
29:
30: %%Page: <text> <uint>
31: %%PageBoundingBox: <int> <int> <int> <int> |(atend)
32: %%PageOrientation: Portrait|Landscape
33: %%PageMedia: <text>
34: %%PaperSize: <text>
35:
36: %%Trailer
37: %%EOF
38:
39: %%BeginDocument: <text> [<real>[<text>]]
40: %%EndDocument
41:
42: %%BeginBinary: <uint>
43: %%EndBinary
44:
45: %BeginData: <uint> [Hex|Binary|ASCII[Bytes|Lines]]
46: %%EndData
```

Many of these comments are used in the page description file created by the Graphics Editor in the following chapters.

Next Steps

The use of color in a PostScript file poses a certain challenge that is complicated by trying to form a single page description file to work when interpreted by both color and black-and-white printers.

The next chapter introduces a PostScript color conversion function that will enable color images to be converted to grayscale during interpretation by black-and-white printers.

Chapter 26

In this chapter

- Determining a Printer's Capability
- Defining Color Images for Black and White Printers
- Next Steps

Color Versus Black and White

Formatting the image data contained in the canvas area of the Graphics Editor as a color image creates an incompatibility with black and white printers. However, the capability to support printing in color or black and white from a single PostScript page description file is possible and is the focus of this chapter.

Determining a Printer's Capability

You must consider whether the destination printer specified for receiving output from the Graphics Editor requires the color values contained in the page description file to be gray-scaled to correctly interpret them.

If you simply try testing for the presence of a color operator such as colorimage, your print system will fail on PostScript Level 2 printers. Many applications make the mistake of testing for the presence of an operator (colorimage, setcmykcolor, and so forth) to determine the printer's capability; however, this is not sufficient. Some Level 1 PostScript black and white devices and all Level 2 devices include the color operators.

Two reasons exist for an application such as the Graphics Editor to care about the color capabilities of a printer. First is to determine whether the printer will accept PostScript language programs using the color extensions (colorimage, setcmykcolor, and so forth). The second reason is to determine whether the printer, as currently configured, will actually produce color output.

If the printer will accept the color extensions, the application should send it colorformatted output, whether or not the printer will actually produce color. In this way, the PostScript language file captures the application's intent, and color will be produced if the file is later diverted to a color printer. The following section describes how to determine whether a printer is capable of PostScript color extensions and how to accommodate those printers that are not.

Defining Color Images for Black and White Printers

The write_ps_color_conv function in Listing 26.1 writes to the page description file referenced by the parameter the PostScript code needed to convert a color image to gray-scale for printers not able to process color extension operands.

```
Listing 26.1 Gray-Scaling Color Raster Data
```

```
static void write ps color conv( FILE *fp )
1:
2: {
3:
     int i = 0;
4:
5:
     static char *ColorConvStr[] = {
                                                            \n",
6:
        "% define colorimage if it is not defined
7:
        "/colorimage where % do we know about colorimage? \n",
8:
          { pop }
                            % yes: pop off the dict returned\n",
9:
          {
                            % no: define one\n
                                                            \n".
10:
            /colortogray { % define an RGB->I function
                                                            \n",
       п
11:
              /robdata exch store
                                     % call input rgbdata
                                                           \n",
       ш
12:
              rgbdata length 3 idiv \n",
                                    \n",
13:
              /npixls exch store
14:
             /rgbindx 0 store
                                    \n",
        ш
15:
             0 1 npixls 1 sub {
                                    \n",
              grays exch
16:
                                    \n",
        ш
              rgbdata rgbindx
                                                 % Red
17:
                                   get 20 mul
                                                            \n",
18:
              rgbdata rgbindx 1 add get 32 mul % Green \n",
19:
              rgbdata rgbindx 2 add get 12 mul % Blue \n",
20:
                add add 64 idiv
                                     \n",
21:
                put \n",
        ш
22:
                /rgbindx rgbindx 3 add store \n",
23:
             } for n",
24:
             grays 0 npixls getinterval \n",
25:
            } bind def \n",
26:
                                                         \n",
27:
            % Utility procedure for colorimage operator. \n",
        ш
28:
            % This procedure takes two procedures off the\n",
        ш
29:
            % stack and merges them into a single one.
                                                         \n",
30:
                                \n",
31:
            /mergeprocs { % def \n",
       ш
32:
              dup length
                                \n",
33:
        ш
              3 -1 roll
                                \n",
       п
34:
                                \n",
              dup
        н
                         \n",
35:
              length
       ш
36:
             dup
                          \n",
37:
              5 1 roll
                          \n",
                         \n",
38:
              3 -1 roll
39:
             add
                          \n",
```

```
40:
               array cvx
                           \n",
41:
                            \n",
               dup
                            \n",
42:
               3 -1 roll
43:
               0 exch
                           \n",
44:
        н
               putinterval \n",
45:
                            \n"
               dup
        ш
46:
               4 2 roll
                           \n",
        п
47:
              putinterval \n".
        ш
48:
             } bind def
                           \n",
49:
                                                      \n",
50:
            /colorimage { % def
                                                       \n",
                         % remove false 3 operands \n",
51.
               pop pop
        ш
52:
               {colortogray} mergeprocs
                                                      \n",
53:
               image
                      \n",
             } bind def \n",
54
        " } ifelse
55:
                         % end of false case\n \n",
56:
        NULL };
57:
58:
      while( ColorConvStr[i] ) {
        fprintf( fp, "%s", ColorConvStr[i++] );
59:
60:
      }
61: }
```

The color conversion function in Listing 26.1 can seem overwhelming if your introduction to PostScript syntax began with Chapter 25, "Introduction to PostScript." However, this function is really not difficult to understand and will serve as an intermediate introduction to the language before proceeding to implement print capability in the Graphics Editor in Chapter 27, "Working with XImages and Colormaps."

The function begins by determining whether the operand colorimage is defined in the existing dictionaries:

7: "/colorimage where

If it is defined, simply pop the returned value from the stack

8: " { pop }

and let normal interpretation of the file continue.

Otherwise, it is necessary to define it. The function defined in absence of the colorimage operand is a procedure to convert the color values of a raster image into weighted gray-scale values:

10: " /colortogray {

Lines 11–15 invoke the rgbdata, define the number of color components, and prepare to substitute the color values with proportionally weighted gray values based on the intensity of the corresponding color component:

17:	н	rgbdata	rgbindx			get	20	mul	%	Red	\n",
18:	II .	rgbdata	rgbindx	1	add	get	32	mul	%	Green	\n",
19:	н	rgbdata	rgbindx	2	add	get	12	mul	%	Blue	\n",

The sum of the values is found and returned to the stack

add add 64 idiv 20: \n", 21: put \n",

before continuing to the next line of data:

23: } for $n^{"}$, 24: grays 0 npixls getinterval \n",

Next, lines 31-48 define mergeprocs, a process which performs some extensive stack manipulation to merge the colorimage and colortogray functions. This is defined for execution in line 52:

п 52: {colortogray} mergeprocs \n",

When the mergeprocs is placed on the stack, it is given the image for processing

п 53: image \n",

and the work is done.

The effect of this PostScript segment is to create a single page description file that will print successfully on both color and black and white printers.

Next Steps

In the next chapter we will look closely at the implementation of the print capability for the Graphics Editor, and see how the color conversion functions introduced here fit into the overall print system.

Chapter 27

In this chapter

- Printing the Canvas
- Creating an XImage
- Creating a PostScript Prolog
- Parsing the X Colormap
- Writing the PostScript Page Definition File
- Directing the Output to a Printer or File
- Next Steps

Working with XImages and Colormaps

The introduction to the PostScript page description language provided by the preceding chapters has prepared you for implementing print capability in the Graphics Editor project.

To accomplish this, it is necessary to create an XImage of the drawing area window and parse the image data to form a digitized raster image understandable by PostScript.

The next section introduces the gx_print function defined as the control action for the print icon of the application's menu panel.

I≣[™] ⊂ Note

The code introduced in this chapter is targeted for the gxGx.c source file. Important too is that non-static function definitions have prototypes for them placed in the gxProtos.h header file. Finally, it will be necessary to ensure that the headers time.h, unistd.h, and sys/stat.h have preprocessor include directives for each of them placed at the beginning of the gxGx.c file.

Printing the Canvas

The print capability of the Graphics Editor prompts the user for the destination of the print action.

As you'll see shortly, the print function tries to determine whether the destination specified is a printer device contained in the /dev directory. If a match to it is not

found, the destination is treated like a file with the PostScript page description for the canvas being written to it.

The gx_print function was previously defined as a stump function to satisfy the link phase of building the application. The actual function definition is presented in Listing 27.1.

```
Listing 27.1 The gx_print Function
```

```
void gx print( void )
1:
2: {
3:
      char * filename = "/tmp/gx-print-data.ps";
4:
5:
      XImage *xi = NULL;
            *fp = NULL;
6:
      FILE
7:
8:
      Dimension
                   width, height;
9:
      int
                   numcols = 0;
10:
11:
             *data;
      byte
12:
      bvte
             Red[MAXCOLORMAPSIZE],
13:
             Green[MAXCOLORMAPSIZE],
14:
             Blue[MAXCOLORMAPSIZE];
15:
      XColor cm[MAXCOLORMAPSIZE];
16:
      fp = fopen( filename, "w+" );
17:
18:
19:
      if( fp ) {
20:
        char *printTo = gxGetFileName();
21:
22:
        get_image( GxDrawArea, &xi, &width, &height );
23:
        write_ps_prolog( fp, width, height );
24:
25:
        if( xi ) {
26:
          data =
27:
            set_color_data(GxDrawArea, xi, width, height,
28:
                            cm, Red, Green, Blue, &numcols);
          write_ps( fp, xi, cm, width, height );
29:
30:
          XtFree( (char *)data );
31:
32:
          XDestroyImage( xi );
33:
34:
          fclose( fp );
35:
        }
36:
        doPrintTo( _filename, printTo );
37:
      }
38: }
```

The gx_print function defines a temporary storage file for constructing the PostScript page definition of the canvas window:

3: char *_filename = "/tmp/gx-print-data.ps";

The function begins by opening the file for writing

17: fp = fopen(_filename, "w+");

and prompting the user for the destination of the print action:

20: char *filename = gxGetFileName();

The gx_print function then creates an XImage from the canvas window and writes the dimensions of the image in a prolog for the print file being constructed:

```
22: get_image( GxDrawArea, &xi, &width, &height );
23: write_ps_prolog( fp, width, height );
```

These functions are defined in listings discussed later in this chapter.

If an XImage was successfully created, the image data is parsed to determine the pixel values of the cells forming the image:

```
26: data =
27: set_color_data(GxDrawArea, xi, width, height,
28: cm, Red, Green, Blue, &numcols);
```

Finally, the raster image is written to the file

29: write_ps(fp, xi, cm, width, height);

and the destination specified by the user is processed with a call to doPrintTo:

```
36: doPrintTo( _filename, filename );
```

It is now necessary to look at the function in more detail, starting with the creation of an XImage from the drawing area.

Creating an XImage

Listing 27.2 defines the get_image function for creating an XImage from the canvas window.

Listing 27.2 Creating an Image from an X Window

```
static void get_image( Widget w, XImage **xi,
1:
2:
                     Dimension *width, Dimension *height )
3: {
4:
      Window
                 window;
5:
                 max_w, max_h, x = 0, y = 0;
     int
6:
7:
     XtVaGetValues( w,
8:
                     XtNwidth.
                                     width.
9:
                     XtNheight,
                                     height,
10:
                     NULL );
```

```
Listing 27.2 Continued
```

```
11:
12:
      *width -= 2; *height -= 2;
13:
14:
15:
      window = XtWindow( w );
16:
      /* ensure even num of rows, postscript demands it */
17:
18:
      *height -= ((*height)%2);
19:
      if( *width > 1 && *height > 1 ) {
20:
21:
22:
          *xi = XGetImage( XtDisplay(w), window,
23:
                            x, y, *width, *height,
24:
                            AllPlanes, ZPixmap );
25:
      }
26: }
```

The width and height of the image extracted from the GxDrawArea are important to the generation of the PostScript description for it. Therefore, the width and height values are returned separately to the calling function for use elsewhere:

```
7: XtVaGetValues( w,
8: XtNwidth, width,
9: XtNheight, height,
10: NULL );
```

The width and height of the widget are reduced slightly

12: *width -= 2; *height -= 2;

to ensure that the requested image is completely contained in the window. An even number of columns must be specified in the raster image passed to the PostScript interpreter as the language requires it:

18: *height -= ((*height)%2);

Finally, the function is able to request:

```
22: *xi = XGetImage( XtDisplay(w), window,
23: x, y, *width, *height,
24: AllPlanes, ZPixmap );
```

The XGetImage function creates an XImage structure definition from the window specified as the second parameter.

The image created does not have to encompass the entire window. Through use of the x, y and width, height parameters, an image smaller than the entire window can be created.

The constants AllPlanes and ZPixmap are used to specify which planes should be employed when creating the image and what the format of the image data should be.

If the format argument is XYPixmap, the image contains only the bit planes you passed to the plane mask if something other than AllPlanes is used as the argument. If the plane mask argument only requests a subset of the planes of the display, the depth of the returned image will be the number of planes requested. If the format argument is ZPixmap, XGetImage returns as 0 the bits in all planes not specified in the plane mask argument.

Having determined the width and height of the image, you can write the prolog for the page description file.

Creating a PostScript Prolog

The creation of the PostScript Prolog is the responsibility of the write_ps_prolog function shown in Listing 27.3.

Listing 27.3 Defining the PostScript Prolog

```
1: static void write_ps_prolog( FILE *fp, int w, int h )
2: {
3:
     time_t btime;
      time( &btime );
4:
5:
6:
      fprintf(fp, "%%!PS-Adobe-2.0 EPSF-2.0\n"
                                                       );
      fprintf(fp, "%%%Creator: J. R. Brown\n"
7:
                                                       );
      fprintf(fp, "%%%Title: 2D Graphic Editor\n"
8:
                                                       );
9:
      fprintf(fp, "%%%CreationDate: %s", ctime(&btime));
      fprintf(fp, "%%%Pages: 1\n");
10:
     fprintf(fp, "%%%BoundingBox: 0 0 %d %d\n", w, h );
11:
12:
     fprintf(fp, "%%%EndComments\n");
      fprintf(fp, "%%%EndProlog\n\n");
13:
      fprintf(fp, "%%%Page: 1 1\n\n");
14:
15: }
```

Although some of the data written in the prolog is for informational purposes only, as discovered in Chapter 25, in the section "Comments Understood by Ghostscript," page 489, the prolog is an important element of the output file.

Following the creation of the PostScript prolog, you are ready to parse the image created by XGetImage to determine the colors used in the many cells that compose the image, as seen in Listing 27.4.

Parsing the X Colormap

An X Colormap is an array of pixel values managing all colors currently displayed on the screen.

The set_color_data function found in Listing 27.4 parses the default Colormap to determine the colors used by the XImage and create the raster image definition that will be passed to the PostScript page description file.

```
Listing 27.4 Determining Image Color Use
```

```
1:
    #define MAXCOLORMAPSIZE 65536
2: typedef unsigned char byte;
3:
   static byte *set_color_data( Widget w, XImage *image,
4:
                                  int width, int height,
5:
                                 XColor *colors, int *n )
8: {
9:
      Boolean mapcols[MAXCOLORMAPSIZE],
10:
              colused[MAXCOLORMAPSIZE];
11:
12:
      Display *dsp = XtDisplay(w);
      int x = 0, i;
13:
14:
15:
      byte *dptr, *data, *iptr = (byte *) image->data;
16:
17:
      dptr =data=(unsigned char *)XtMalloc(height * width);
18:
      memset( colused, False,
19:
              sizeof( Boolean ) * MAXCOLORMAPSIZE );
20:
21:
      setStatus( "Gathering color data... please wait" );
22:
      for ( i = 0; i < MAXCOLORMAPSIZE; i++) {</pre>
23:
        colors[i].pixel = i;
24:
        colors[i].flags = DoRed | DoGreen | DoBlue;
25:
26:
        XQueryColor(dsp,
27:
                    DefaultColormap(dsp, 0), &colors[i]);
28:
      }
29:
      setStatus( "Determing color usage ..." );
30:
31:
      for (i = 0;
           i < image->bytes per line*height; i++, iptr++) {
32:
33:
34:
        if (x >= image->bytes per line)
35:
          x = 0;
36:
37:
        if (x < width) {
          colused[*iptr] = True; /* mark this color as used */
38:
39:
          *dptr++ = *iptr;
40:
        }
41:
        x++;
```

```
42:
      }
43:
44:
      setStatus( "Processing color data ..." );
45:
      for (i = 0; i < MAXCOLORMAPSIZE; i++) {</pre>
        if( colused[i] ) {
46:
47:
48:
          mapcols[i] = *n;
49:
54:
          (*n)++;
55:
        }
56:
      }
57:
58:
      setStatus( "Transferring image colors..." );
      dptr = data;
59:
60:
      for (i = 0; i < width * height; i++) {
        *dptr = mapcols[*dptr];
61:
62:
        dptr++;
63:
      }
64:
65:
      return data;
66: }
```

The actual image data comprising the XImage data field is an array of pixel values where each pixel corresponds to an entry in the Colormap used to draw the image to the screen.

The set_color_data function in lines 22–28 creates an array of all color data for the pixels contained in the Colormap.

It then traverses the image data to determine which color is used:

```
38: colused[*iptr] = True; /* mark this color as used */
```

The color data extracted from the Colormap is then traversed to see whether the color was used:

```
46: if( colused[i] ) {
```

If the color was used, the number of the color (assigned according to order of occurrence) is used as an index into an array, tracking the colors used from the Colormap:

48: mapcols[i] = *n;

The pixel values stored to reflect the colors used from the Colormap are transferred to the raster data array that is used as the image definition passed to the PostScript interpreter:

```
60: for (i = 0; i < width*height; i++) {
61:     *dptr = mapcols[*dptr];
62:     dptr++;
63: }</pre>
```

Finally, the data is returned to the calling function by passing back the pointer of the beginning of the data block:

65: return data;

With the raster image data formed for the page description file, the data can now be written to the temporary file using the write_ps function defined in Listing 27.5.

Writing the PostScript Page Definition File

The function write_ps provided in Listing 27.5 is responsible for forming the contents of the page description file that is eventually passed to the PostScript interpreter.

Listing 27.5 The write_ps Function

```
1:
    static void write_ps(FILE *fp, XImage *xi,
2:
                         XColor *cm, Dimension h, Dimension w )
3: {
4:
      write_ps_prolog( fp, w, h );
5:
6:
      fprintf(fp, "%% remember original state\n/saveorig save def\n\n");
      fprintf(fp, "%% define string to hold scanline's worth of data\n");
7:
      fprintf(fp, "/imgstr %d string def\n\n", w );
8:
9:
      fprintf(fp, "%% build a temporary dictionary\n20 dict begin\n\n" );
      fprintf(fp, "%% define space for color conversion\n");
10:
      fprintf(fp, "/grays %d string def %% space for gray scale line\n", w*3);
11:
12:
      fprintf(fp, "/npixls 0 def\n/rgbindx 0 def\n\n" );
13:
14:
      fprintf(fp, "%% corner of image\n%d %d translate\n\n", 25, 25);
15:
     fprintf(fp, "%% size of image on paper\n%d %d scale\n\n",
16:
17:
      w > 550 ? 550 : w,
18:
      h > 740 ? 740 : h ); /* could be changed for scaling */
19:
20:
      write ps color conv( fp );
21:
22:
      fprintf( fp, "%d %d 8\n", w, h );
23:
      fprintf( fp, "[%d 0 0 %d 0 0]\n", w, h );
24:
25:
      fprintf( fp, "{ currentfile\n\timgstr readhexstring pop }\n");
26:
      fprintf( fp, "false 3 colorimage\n\n");
27:
28:
      write_ps_data( fp, xi, cm, w, h );
29:
30:
      fprintf( fp, "showpage\n\nend\n\nsaveorig restore\n\n");
31:
      fprintf( fp, "%%%Trailer\n\n%% End-of-file\n");
32: }
```

Following the writing of the PostScript prolog, the current state of the interpreter is stored

6: fprintf(fp, "%% remember original state\n/saveorig save def\n\n");

a variable is defined to store a single line of data from the raster image

```
8: fprintf(fp, "/imgstr %d string def\n\n", w );
```

and a temporary dictionary is defined:

9: fprintf(fp, "% build a temporary dictionary\n20 dict begin\n\n");

Then, in preparation for color translation, space is reserved for performing gray scaling of the color data

```
11: fprintf(fp, "/grays %d string def %% space for gray scale line\n", w*3);
additional variables are defined and initialized
```

12: fprintf(fp, "/npixls 0 def\n/rgbindx 0 def\n\n");

and it's time to start working by assigning the placement of the image corner:

14: fprintf(fp, "%% corner of image\n%d %d translate\n\n", 25, 25);

The image being described in the file is then scaled to fit the paper

```
16: fprintf(fp, "% size of image on paper\n%d %d scale\n\n",
17: w > 550 ? 550 : w,
18: h > 740 ? 740 : h ); /* could be changed for scaling */
```

and a call to the write_ps_color_conv function, introduced in Chapter 26, "Color Versus Black and White," is called to support color printers.

More housekeeping chores are addressed with the definition of the image data size and number of data bits:

```
22: fprintf( fp, "%d %d 8\n", w, h );
23: fprintf( fp, "[%d 0 0 %d 0 0]\n", w, h );
25: fprintf( fp, "{ currentfile\n\timgstr readhexstring pop }\n");
26: fprintf( fp, "false 3 colorimage\n\n");
```

Finally, the image data is written to the file using the write_ps_data function found in Listing 27.6.

The command to show the page is ordered

```
30: fprintf( fp, "showpage\n\nend\n\nsaveorig restore\n\n");
31: fprintf( fp, "%%%Trailer\n\n%% End-of-file\n");
```

and the original state is restored before ending the page.

```
Listing 27.6 The write_ps_data Function
```

```
static void write ps data( FILE *fp, XImage *xi,
1:
2:
                     XColor *cm, Dimension w, Dimension h )
3:
   {
4:
      int
             rows, cols, cell;
5:
6:
      for ( rows = 0; rows < h; rows++ ) {
7:
        for( cols = 0; cols < w; cols++ ) {
8:
9:
          cell = XGetPixel( xi, rows, cols );
10:
          fprintf( fp, "%2.2x", cm[cell].red
11:
                                                / 256 );
          fprintf( fp, "%2.2x", cm[cell].green / 256 );
12:
          fprintf( fp, "%2.2x", cm[cell].blue / 256 );
13:
14:
15:
        fprintf( fp, "\n" );
16:
17:
      }
18: }
```

Using the XImage captured from the GXDrawArea window, the write_ps_data loops over the width and height of the image to generate a row and column coordinate. This row and column is then passed to the XGetPixel to get the pixel value defining the position:

```
9: cell = XGetPixel( xi, rows, cols );
```

The pixel value returned by XGetPixel is used as an index into the cm array generated to reflect the colors in use by the image. Each color component is stored independently to the PostScript file in the range of 0–256:

```
11: fprintf( fp, "%2.2x", cm[cell].red / 256 );
12: fprintf( fp, "%2.2x", cm[cell].green / 256 );
13: fprintf( fp, "%2.2x", cm[cell].blue / 256 );
```

As the entirety of the image cells are considered, the data gets written to the page description file.

With the contents of the PostScript description file complete, it is possible to parse the destination entered by the user at the beginning of the gx_print function to determine where the PostScript file should placed.

Directing the Output to a Printer or File

The value returned by the gxGetFileName function in Listing 27.1

```
20: char *printTo = gxGetFileName();
```

could be the name of a printer configured for the system or, optionally, the name of a file.

The doPrintTo function in Listing 27.7 attempts to determine whether the destination is a printer by looking for a corresponding device in the /dev/ directory; otherwise, it treats the user input as a file.

Listing 27.7 The doprint o Function for Determining the Print Data Destination

```
1:
    static void doPrintTo( char *datafile, char *printTo )
2: {
3:
      struct stat sbuf;
4:
      char cmd[128], lpDev[128];
5:
6:
      FILE *fp = NULL;
7:
      if( printTo ) {
8:
        sprintf( lpDev, "/dev/%s", printTo );
9:
10:
11:
        if( stat( lpDev, &sbuf ) < 0 ) {
          sprintf( cmd, "mv %s %s", datafile, printTo );
12:
        } else {
13:
14:
           sprintf( cmd,
15:
                    "unalias lp; lp -c -d %s %s;rm %s",
16:
                    printTo, datafile, datafile );
17:
        }
18:
      } else {
          sprintf( cmd,
19:
20:
                   "unalias lp;lp -c %s;rm %s",
21:
                   datafile, datafile );
22:
      }
23:
     if((fp = popen(cmd, "r")) != NULL) {
24:
25:
        char buf[128];
26:
        fgets( buf, sizeof( buf ), fp );
27:
        pclose( fp );
28:
      }
29:
30:
      setStatus( "Printing complete!" );
31: }
```

If a printer of the name entered by the user does not match the name of a device in the /dev directory, a command is assembled to move the temporary PostScript page definition file to a file of the name entered:

12: sprintf(cmd, "mv %s %s", datafile, printTo);

Otherwise, if the stat command reflects that a printer of the same name as entered by the user exists, the contents of the temporary PostScript file are directed to the 1p command, specifying the printer name entered as the destination device:

14: sprintf(cmd, 15: "unalias lp; lp -c -d %s %s;rm %s", 16: printTo, datafile, datafile);

If nothing was entered by the user, it is assumed that the system default printer is the desired destination

```
19: sprintf( cmd,
20: "unalias lp;lp -c %s;rm %s",
21: datafile, datafile );
```

and a command is formed omitting the destination (-d) parameter.

With a command properly assembled, it is possible to open a stream with the operating system specifying the command to execute:

24: if((fp = popen(cmd, "r")) != NULL) {

As the stream is opened for reading, it is possible to get the status of the command returned to the application:

```
26: fgets( buf, sizeof( buf ), fp );
```

The stream must be closed when you are finished with it because a finite number of streams can be opened concurrently by an application:

```
27: pclose( fp );
```

This completes the addition of print capability to the Graphics Editor.

Clearly, this approach to printing is direct and easy to implement. However, it could be improved upon; namely, the use of the raster image in the PostScript file does not ensure that the page described by the PostScript file looks exactly like the image on the canvas.

Because PostScript provides graphic primitives similar to the X Window System, a superior approach for printing the canvas would be to enable each of the objects in the editor to have a second draw method defined to use the appropriate PostScript function to draw the graphic objects. Although a more complex solution, it is one that would offer a higher quality print product and *WYSIWYG* (*what you see is what you get*) output.

Next Steps

The final chapters of this text propose ideas for furthering the Graphics Editor project and integrating it into other applications that could benefit from an annotation subsystem.



What's Next?

Chapter 28

In this chapter

- Attributes
- Rotating Objects
- Next Steps

Extending the Graphics Editor

The Graphics Editor project, although able to draw, move, scale, and delete objects, requires many more capabilities before it is considered complete.

This chapter introduces features that the project structure lends itself to but that are left for you to implement.

Attributes

The definition of the common object data structure in Chapter 15, "Common Object Definition," page 305, provided fields within the structure to store the foreground and background colors as well as the line width and line style attributes of the editor objects.

These fields were included in the Save and Restore function of the editor application introduced in Chapter 19, in the section "Common Object Save and Restore," page 367.

However, the construction of the application did not provide for the selection of colors or line attributes by the user.

The following sections pose ideas to lead you in the extension of the Graphics Editor application to support the assignment of object attributes.

Color

The first consideration when extending the Graphics Editor project is how to provide a method for the user to assign, alter, and manipulate colors. Specific to this is the interface mechanism for making colors available to the user for selection and assignment. Either by adding an attributes button panel adjacent to the control panel or a dropdown menu accessed from the canvas window, a palette should be defined for presenting allowable colors.



The phrase allowable colors is important.

One shortcoming of the Graphics Editor application in its current state is the storage of Pixel values in the file created for the Save and Restore feature.

This is counted as a shortcoming because the order of the colors installed in the Colormap, reflected by the Pixel value index into the map, is arbitrary and affected entirely by the order in which the colors are requested.

As applications are started on the desktop, the colors they require are installed into the default Colormap. Changing the order in which the applications are started changes the order in which the colors are loaded.

For this reason, storing the Pixel value does not ensure that, in consecutive sessions of the application, objects restored from the data file will be drawn in the correct or original colors.

Defining *allowable colors* enables the Save and Restore feature to use an index that is meaningful to represent the color values.

After the interface mechanism is decided, the active objects selected by the user could have their color values altered.

In addition to affecting color settings, the Graphics Editor should account for the changing of line attributes, as described in the next section.

Line Attributes

Similar to assigning color values to the editor objects, the extension of the Graphics Editor project requires providing the capability to alter the line width and line style of the point-array-based editor objects.

Line Width

Valid values assigned as the line width attribute of point-array objects are any integer describing the thickness of the line.



A Graphics Context created with the line_width field of the XGCValues structure set to 0 instructs the X Server to draw the line using the fastest algorithm possible.

The interface mechanism to support this could simply be an entry field where the user could provide a number. This clearly would require validation to ensure that characters are not erroneously entered. Optionally, an up and down arrow could be created to increment or decrement the object's line width respectively.

Line Style

Valid values for the line style attribute assignment of editor objects include LineSolid, LineDashed, and LineDoubleDashed.

These draw either a solid, dashed, or double-dashed line, as indicated by their name.



A double-dashed line is drawn with one segment of the dash drawn in the foreground color assigned to the Graphics Context created for the invocation of the X Window Graphic Primitive and the next segment drawn in the background color of the GC.

A normal dashed line uses only the foreground color and skips alternate segments by drawing nothing.

Arc Angles

Complementing the line attributes of the previous section, attributes of the Arc object are provided by the XArc data structure but not addressed by the interface of the Graphics Editor application.

These are, specifically, the angle1 and angle2 fields of the structure. Review Chapter 8, "Vector Versus Raster Graphics," page 197, for a discussion of these fields and valid assignments.

Rotating Objects

In Chapter 11, "Graphic Transformations," in the section "Rotating," page 247, the mathematics required to rotate the point-array editor objects was introduced.

Considered an advanced feature of the editor, structuring the capability of assigning and altering object attributes should include a degree factor of rotation as an editable field.

The interface mechanism most suited for altering the degree field you'll add to the common object definition is the scale bar.

The scale should be assigned a minimum scale factor of 0 and a maximum of 360, enabling the user to apply a meaningful value for degrees of rotation.

Next Steps

After the management of attributes outlined in this chapter is implemented, the Graphics Editor application will be complete. The steps remaining are limited only by your imagination.

The following chapter structures the addition of a context-sensitive help system to the editor application.

Chapter 29

In this chapter

- Processing Help-Related Events
- Widget Paths
- Relating Widgets to Text
- Next Steps

Adding Context-Sensitive Help

I offer this chapter to the reader desiring to continue the development of the Graphics Editor Project beyond the scope of the text.

A suggested modification to the editor application's GUI is the support of a contextsensitive help facility.

As implied by the name, *context-sensitive help* enables a user to query the application's interface effectively for assistance in determining how to use the application. This capability would expand the hints currently provided in the status window at the bottom center of the interface.

Clearly, the graphical user interface for the Graphics Editor is not complex. However, the user has no intuitive way of knowing the specific mouse actions expected for creating each of the objects supported by the editor.

A context-sensitive help system would enable the user to press the F1 key to enter the help mode. The cursor would change to resemble a question mark, and the user would select an item from the application's interface for which help is desired.



The purpose of this chapter is to pose an idea for your continued learning using the Graphics Editor project.

The following sections of this chapter do not fully implement a context-sensitive help system. Instead, a structure for the system is discussed, and key areas are expanded and explained.

Processing Help-Related Events

The key to a context-sensitive help system is trapping the user's selection of the F1 key to enter the help mode. Communication of this event to the Graphics Editor application can be gained in several ways.

For instance, an event handler, as introduced in Chapter 14, "Program Flow," can be assigned the various widgets of the graphical user interface. However, a more grace-ful approach is to define your own event loop to replace the XtAppMainLoop call used by the application.

The event loop is easy to construct and normally would only require an endless loop to extract events from the application's event queue, using XtAppNextEvent and subsequently sending the event to the application's widgets for processing using XtDispatchEvent.

The requirement of the context-sensitive help, however, is to determine when the user presses the F1 key. Therefore, it is necessary to sample the events removed from the queue.

Listing 29.1 demonstrates how a unique event loop can be defined to support the proposed help system.

Listing 29.1 The HelpAppMainLoop Function

```
1:
    void HelpAppMainLoop( XtAppContext app )
2:
   {
3:
      XEvent event;
4:
5:
      for( ;; ) {
        XtAppNextEvent( app, &event );
6:
7:
        XtDispatchEvent( &event );
8:
        if( helpEnabled == G FALSE ) continue;
9:
10:
        switch( event.type ) {
11:
        case KeyPress:
12:
13:
          processKeyPress( &event );
14:
          break;
15:
        default:
16:
          break;
17:
        }
18:
      }
19: }
```

As with the XtAppMainLoop, our event loop will need access to the XtAppContext structure created from the call to XtVaAppInitialize.

The only requirement of the event loop is that the events be removed and dispatched continuously:

6: XtAppNextEvent(app, &event); 7: XtDispatchEvent(&event);

The sample loop from Listing 29.1 enables the help system to be optional by testing an implied global variable helpEnabled:

9: if(helpEnabled == G_FALSE) continue;

In the case of the help system being enabled, the HelpAppMainLoop event type is tested in search of the KeyPress event

12: case KeyPress:

and when it is found, the value of the key pressed by the user is determined in a call to the processKeyEvent function found in Listing 29.2.

```
Listing 29.2 The processKeyEvent Function
```

```
1: static void processKeyPress( XEvent *xe )
2: {
3:
     XEvent e;
4:
      KeySym keysym;
5:
     Widget help_widget;
6:
7:
     Cursor cursor =
8:
        XCreateFontCursor(XtDisplay(GxDrawArea), XC_question_arrow);
9:
10: if( xe->type == KeyPress ) {
11:
        keysym = XKeycodeToKeysym(XtDisplay(GxDrawArea),
12:
                                   xe->xkey.keycode, 0);
13:
14:
        if(keysym == XK F1 || keysym == XK KP F1) {
15:
          /* we got a 'help' KeyPress, activate the selection */
16:
          help_widget = trackingEvent( toplevel, cursor, True, &e );
17:
18:
19:
          XUndefineCursor( ui_display, XtWindow(toplevel));
20:
21:
          /* a specific widget was selected, offer help */
22:
          activate_help( help_widget );
23:
        }
24:
      }
25: }
```

Two critical areas are addressed by the processKeyEvent function. The first entails deciphering the key pressed

11: keysym = XKeycodeToKeysym(XtDisplay(GxDrawArea), 12: xe->xkey.keycode, 0);

and the second is waiting for the next ButtonPress event to determine the widget for which context-sensitive help is being requested:

```
17: help_widget = trackingEvent( toplevel, cursor, True, &e );
```

The KeyPress event structure contains a field xe->xkey.keycode, which holds, for the purpose of supporting multiple language definitions, the encoded key value for the button pressed. The value of the encoded key is converted to a KeySym for comparison by the XKeycodeToKeysum Xlib function.

The trackingEvent function left for your definition must return the widget receiving the very next button press event that signals the user's desire for help on the entity.

Using the widget's explicit path within the application's instance hierarchy, as discussed in the next section, a correlation is made to the help text specific for this widget.

Widget Paths

Every widget has an explicit place in the hierarchy of all widgets contained in the application. This placement in the hierarchy is known as the widget's *path*. If the programmer chooses unique widget names (as assigned by the first parameter of the XtVaCreateManagedWidget function), the path will uniquely identify the widget.

The widget's instance name preceded by the instance name of its parent forms a portion of the widget's path. If you continually prepend the name of the parent widget's parent (and so forth), an explicit widget path for the initial widget is determined.

The getWidgetPath and getWidgetPathComponents functions defined in Listing 29.3 demonstrate how to find a widget path for the widget returned by trackingEvent seen earlier.

```
Listing 29.3 The getWidgetPathComponents Function
```

```
1:
    static void getWidgetPathComponents( Widget parent,
2:
                         Widget w, char **path, int *len )
3: {
4:
      char *wname;
5:
6:
      if( parent ) {
7:
        /* throw away the toplevel */
8:
        if( !XtParent(parent) ) return;
9:
10:
        getWidgetPathComponents( XtParent(parent), w, path, len );
11:
12:
        /* extract the name of the widget */
13:
        wname = XtName( parent );
14:
```

```
/*
15:
16:
         * grow if necessary
         */
17:
18:
        if( strlen(wname) + 1 >= *len ) {
19:
          /*
          * find the current end position
20:
           */
21:
22:
          int endPos = strlen( *path ) - 1;
23:
24:
          (*len) += 512;
25:
          *path = (char *)realloc( *path, *len);
26:
          /*
27:
28:
          * zero out what has just been alloc'd
          */
29:
30:
          memset( *path + endPos, 0, (*len) - endPos );
31:
        }
32:
33:
        /*
         * store the name of this widget and a '.'
34:
35:
         * if there are more components to follow
36:
         */
37:
        strcat( *path, wname );
38:
        if( parent != w )
39:
          strcat( *path, ".\0" );
40:
      }
41: }
42:
43: /*
44: ** getWidgetPath
45: **
46: ** follow the widget up its tree (hierarchy) assembling names
47: */
48: static char *getWidgetPath( Widget w )
49: {
50:
      static char *wpath = NULL;
51:
      static int wpathLen = 0;
52:
53:
      /* get a starting point for our buffer */
54:
      if( wpath == NULL || wpathLen == 0 ) {
55:
       wpathLen = 2048; /* hopefully it won't grow */
56:
        wpath = (char *)malloc( wpathLen );
57:
      }
58:
59:
      /* clear it out */
60:
      memset( wpath, 0, wpathLen );
      getWidgetPathComponents( w, w, &wpath, &wpathLen );
61:
62:
63:
      return wpath;
64: }
```
Invoking getWidgetPath

48: static char *getWidgetPath(Widget w)

for the widget returned from trackingEvent constructs a widget path to uniquely identify it within the application.

The getWidgetPath begins by creating sufficient memory space to store the path

56: wpath = (char *)malloc(wpathLen);

which it initializes

```
60: memset( wpath, 0, wpathLen );
```

and then begins the recursive process of extracting instance names for the widget's ancestors:

```
61: getWidgetPathComponents( w, w, &wpath, &wpathLen );
```

The getWidgetPathComponents function continually invokes itself, passing the parent of the current parent:

```
10: getWidgetPathComponents( XtParent(parent), w, path, len );
```

When the top of the hierarchy is found the recursion ceases

8: if(!XtParent(parent)) return;

and the names are assembled in the memory set aside for its use:

37: strcat(*path, wname);

This widget path is related to either a help text string or the name of an HTML file that is displayed to satisfy the user's query for help, as described in the next section.

Relating Widgets to Text

With a complete path formed for the selected widget, it is possible to search an external configuration file that you will define to relate the widgets of your application to the help file describing each entity of the application.

A sample structure for the entries in the configuration file is provided in Listing 29.4.

```
Listing 29.4 Sample Help Configuration File Entry
```

```
1: static char *help_entry = "<\n\
2: <"filename.html#OPT_TAG\">\n\
3: <widgetpath>\n\
4: <Title>\n\
5: >\n";
```

The purpose of the configuration file entries is to associate each widget path of the application with help information.

This information can either be a string displayed in a help dialog created by the application, or as illustrated in Listing 29.4, an HTML file that the application uses to invoke an external browser to display help for the item selected by the user.

Next Steps

This chapter did not fully implement the context-sensitive help system but formed a to-do list outlining the key elements required to accomplish the task.

Sufficient information and structure is provided for the reader to work independently solving the proposed context-sensitive help system..



Appendixes

Appendix A

Command Shells and Scripting

The UNIX command shell is adequately named because what is visible is only the outside. Internally, there is much more to a command shell than is immediately understood.

UNIX is unique in letting users choose their command shell. Most operating systems (DOS, VMS, and so forth) have the command interpreter built in, giving users no alternative.

Under UNIX, the command interpreter is not a part of the operating system but rather a wrapper around it.

This appendix provides an overview of the command shells commonly available under many UNIX systems. Following an overview of shells, the focus is on writing shell scripts using the Bourne shell.

UNIX Command Shells

The Bourne shell (sh) is the standard among the UNIX shells because it was the original command shell provided by the Computer Research Group of Bell Labs.

The Bourne shell, however, lacks a significant feature known as job control, making it less than ideal for interactive use. Typically, the Bourne shell is used for scripting, independent of the shell assigned for interactive use.



Every user is assigned a default shell in the last field of the /etc/passwd file for the entry defining their account. This is the shell that executes automatically when the user's login process completes.

Knowing the different command shells enables users to execute and employ other shells that are not their default.

Optionally, some versions of UNIX enable users to change their default shell through use of the passwd command by specifying the -s flag. Read the man page for the passwd command available under the version of UNIX you are running to determine whether this option is available; otherwise, a request to your System Administrator might be necessary to change your default shell.

EXCURSION

Controlling Jobs from Within a Command Shell

The job control feature enables processes to be moved from foreground to background and vice versa without closing and restarting the program.

A foreground process running in your command shell does not enable you to continue to issue commands until the process has finished. A background process (executed with a trailing ampersand) enables use of the command shell by the user for issuing commands and navigating the system to continue.

A shell that supports job control will enable a foreground process, meaning a process executed without a trailing ampersand (&), to be suspended by holding down the Control key and pressing the letter Z in the window where the program is running. Further, the process can be continued by use of the fg command to return it to the foreground or the bg command to return it to the background.

The command jobs is used to list the processes that a shell controls.

Each process displayed in the list is preceded with a number assigned sequentially as consecutive commands are executed. This number is referred to as a *job number*, and when prefaced with a percent sign (%) can be used in place of the process's ID obtained with the **ps** (process status) command.

Command shells commonly assigned as the interactive shell employed by users include the Korn shell (ksh), C shell (csh) and TC shell (tcsh).

how too pro nouns' it

The C Shell is pronounced as if it were written *sea shell* and the TC shell as if it were written *tea sea shell*.



The *C shell* first introduced job control. It is a commonly assigned interactive shell but a poor choice for scripting because the method of implementing functions is non-standard between implementations of the shell and because nested if statements often do not work as expected.

The Korn shell is an extension of the Bourne shell and made significant improvements to the Bourne shell's interactive friendliness. As a derivative of the Bourne shell, the Korn shell is well suited for shell programming.

The Korn shell is proprietary to AT&T but is available as the Bourne Again shell (bash) on non-AT&T versions of UNIX.

Command Shell Environment

The command shell you employ dictates certain differences for configuring and working in the UNIX system. For instance, the syntax for setting variables in the shell environment and making them available between sessions varies depending on the shell used.

Shell Variables

The method of defining variables within a command shell varies based on the syntax understood by the shell. The following sections show the differing ways of defining variables for common command shells.

Bourne Shell Variables

The syntax to set a variable in the environment of the Bourne shell or one of its variants adheres to the following form:

VAR=value

By convention, an environment variable is fully capitalized. This aids in distinguishing it from a shell or system command.

Specifying an environment variable on the shell command line defines the variable only for the current session and ensures visibility only for the current shell.

To make the variable visible to sub-shells, the internal shell command export is used.

export VAR



A *sub-shell* is a shell executed from within another shell.



An *internal command* refers to the fact that only the shell will interpret the command. Other shells might not understand it, and it will not physically reside on the UNIX system.

To read about an internal shell command, you will have to display the man page for the shell to which it is internal. An example of this is the export command, introduced for making variables available to subshells.

The export command can be combined with the variable declaration and value assignment.

export VAR=value

C Shell Variables

Setting variables in the C shell or one of its derivatives requires the internal shell command setenv.

setenv VAR value



Syntactically, it is incorrect to use an equal sign (=) with the C shell variable declaration because the equal sign is unique to the Bourne shell. In other words, attempting the Korn shell syntax

VAR=value

generates a command unknown error in the C Shell.

Unlike the Bourne shell, the C shell enables a variable to unset by use of the unsetenv internal shell command.

unsetenv VAR

To display all the variables set within an environment, use the env command. A specific variable's value can be displayed by either echoing its value using the syntax

echo \$VAR

or using the following printenv internal shell command:

printenv VAR

Aliases

Most shells enable UNIX commands to have an alias defined in the shell environment to simplify the command's use. For instance, using C shell syntax, the following alias can be defined for performing long directory listings:

alias ll ls -l

In this example, the alias internal shell command defines the alias 11 to be the 1s command with the -1 flag. After executing the alias command, the 11 alias is available as if it were a valid UNIX or shell command.

Variables and aliases defined on the command line are destroyed when the shell exits unless they are placed in an *environment file* read by the shell when it executes.

Environment Files

Command shells available under UNIX search for and interpret the contents of an environment file when it executes. This environment file enables a user to make variables and aliases available between user sessions.

Which file to create or modify within your user environment depends upon the shell being used.

Bourne Shell Environment Files

The Bourne shell (and derivatives) expects the environment configuration file to be placed in the user's home directory and given the name

\$HOME/.profile



Notice the dot (.) preceding the filename. A file preceded by a dot is called a hidden file because this makes the file invisible to the directory listing obtained by the 1s command unless the -a (all) flag is used.

The contents of the .profile file can be any valid UNIX or internal shell command, one per line or separated by a semicolon (;), and entered exactly as if typed on the command line.

Listing A.1 shows a sample .profile file for initializing an environment when using the Bourne command shell or derivative.

```
Listing A.1 Sample .profile File
```

```
1: # Sample .profile file
2: PATH="$PATH:/usr/X11R6/bin"
3: PS1="[\u@\h ]\\$ "
4:
5: USER=`id -un`
```

A

Listing A.1 Continued

```
6: LOGNAME=$USER
7: MAIL="/var/spool/mail/$USER"
8: HOSTNAME=`/bin/hostname`
9: HISTSIZE=1000
10: HISTFILESIZE=1000
11:
12: export PATH PS1 USER LOGNAME MAIL \
13: HOSTNAME HISTSIZE HISTFILESIZE INPUTRC
```

Understanding Listing A.1 only requires that you apply many of the things you learned in previous chapters of this book.

Starting after the comment of line 1, consider

2: PATH="\$PATH:/usr/X11R6/bin"

which sets an environment variable called PATH. All command shells employ a PATH variable for identifying where the shell should look for UNIX commands. A UNIX command not contained in one of the elements of the PATH variable will be reported as command not found unless an explicit path is specified.

As this setting is in the .profile file, the value is set every time the Bourne shell executes.

Following the assignment of the PATH variable, the variable called PS1 is set:

```
3: PS1="[\u@\h ]\\$ "
```

The PS1 variable is a secondary or command continuation prompt. When the command shell is told that a command will extend to a second line through use of the line continuation character backslash ($\)$ followed by return, the value of PS1 is used as the secondary prompt.

Every command shell will employ the PS1 variable for this purpose; however, the syntax understood by the shell when interpreting this variable depends upon the shell being used.

The syntax demonstrated in Listing A.1 for the Bourne Again Shell (bash) will yield the prompt

```
[username@hostname] $
```

because the token \u specifies the username, \h the hostname and \\\$ simply '\$'.

The next few lines of Listing A.1 set several useful environment variables using standard UNIX commands and variables after they are defined.

```
5: USER=`id -un`
6: LOGNAME=$USER
7: MAIL="/var/spool/mail/$USER"
8: HOSTNAME=`/bin/hostname`
```

Pay close attention to the syntax for calling a UNIX command from within a shell script and employing the result. The shell must be told to evaluate the command before performing the assignment. This is accomplished by encasing the command between tick marks (`).

The HISTSIZE and HISTFILESIZE variables determine the number of commands remembered by the shell during a session and how many are retained in the .history file.

9: HISTSIZE=1000 10: HISTFILESIZE=1000

Shell command history enables the user to repeat commands by either using the up arrow key to have previously executed commands repeated at the command prompt or by using the history command to see and select commands from a list.

EXCURSION

```
Repeating Commands from the Shell History
```

To repeat a command from the history list, preface the sequential number assigned every command with an exclamation point on the command line.

bash[100]: !23

would repeat the 23rd command as it appears in the history list.

Optionally, you can use the exclamation point followed by the first few letters of the command.

bash[101]: !m

would repeat the last command executed that started with the letter m.

bash[102]: !ps

would repeat the last ps command.

The last two lines of Listing A.1

12: export PATH PS1 USER LOGNAME MAIL \ 13: HOSTNAME HISTSIZE HISTFILESIZE INPUTRC

export the variables set to ensure that they are visible to any sub-shells the current shell might parent.

A

C Shell Environment Files

The file sought and loaded by C shell and its derivatives is the .cshrc file. A sample is shown in Listing A.2.

Listing A.2 Sample .cshrc File

```
1:
       # Sample cshrc
2:
3:
       if ($?PATH) then
4:
           setenv PATH "${PATH}:/usr/X11R6/bin"
5:
       else
6:
           setenv PATH "/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin"
7:
       endif
8:
9:
       if ($?prompt) then
10:
         if ($?tcsh) then
           set prompt='[%n@%m %c]$ '
11:
12:
         else
13:
           set prompt=\[`id -nu`@`hostname -s`\]\$\
14:
         endif
       endif
15:
16.
17:
       setenv HOSTNAME `/bin/hostname`
18:
       set history=1000
```

In the sample .cshrc shown in Listing A.2, a test is conducted following the comment to see whether the PATH variable is already set in the environment.

```
3: if ($?PATH) then
4: setenv PATH "${PATH}:/usr/X11R6/bin"
5: else
6: setenv PATH "/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin"
7: endif
```

By now, the syntax \$VAR should be understood as the method for accessing the value of a variable.

The syntax shown in the sample .cshrc introduces the question mark in conjunction with the variable access as the means for testing whether the variable value is of non-zero length to indicate whether a value is set.

The sample either appends a value to an existing value to ensure the X client commands are seen by the shell

4: setenv PATH "\${PATH}:/usr/X11R6/bin"

or performs an initial assignment

6: setenv PATH "/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin"

The next section of Listing A.2 determines whether the prompt value has been set. If not, based on which shell is being used, this section sets it accordingly:

```
9: if ($?prompt) then
10: if ($?tcsh) then
11: set prompt='[%n@%m %c]$ '
12: else
13: set prompt=\[`id -nu`@`hostname -s`\]\$\
14: endif
15: endif
```

The test for which shell is currently being executed is necessary because the syntax understood by the TC Shell differs from the syntax understood by the C shell for specifying the prompt value.

The final step is the setting of the variable HOSTNAME

```
17: setenv HOSTNAME `/bin/hostname`
```

and the setting of the size of the shell history buffer

18: set history=1000

The following section describes how to write shell scripts using the Bourne shell.

Scripting with the Bourne Shell



Although targeted for the Bourne shell, this section applies to the Bourne shell enhancement *Korn shell* and the Korn shell clone *Borne Again shell*.

Shell scripting gives a UNIX software developer the ability to automate simple tasks in a portable and extensible manner.

Scripts are easy to create and modify and are immediately available.

To begin a shell script, the first line *must* specify the UNIX system that is the intended shell for interpreting the script. This designator looks very much like a comment, but is read and used by the system.

The shell designator uses the tokens #! followed by the command for the shell, including an explicit path.

#!/bin/sh

After specifying the interpreter, there are no other requirements beyond obeying the syntax for the commands and conventions used in the body of the shell.

Shell Variables

Declaring variables in shell script follows syntax that you have already seen in the .profile read by the Bourne shell when it executes.

VAR=value

A syntactical requirement is that no spaces exist between the components of the variable definition VAR, =, and value. If a space is present, the Bourne shell will perceive that an attempt is being made to execute a command and attempt to invoke VAR leading to a statement of the sort

```
VAR: command not found
```

A default value can be specified when using a variable by encasing the variable in braces and using a dash (-) separator between the variable name and the default value

```
${VAR-default}
```

Quoting Variables

When learning shell programming, you often won't know when to quote, what to quote, and which quotes to use.

Single forward quotes protect against all processing when spaces are used during variable assignments, as in the following example:

VAR='value1 value2'

Without the use of single quotes when a variable contains nested spaces, the shell will attempt to invoke VAR as a command.

Double quotes enable variable substitutions, with the result being a single argument, shown in the following example:

```
VAR='value1 value2'
VAR2="$VAR"
```

By convention, and to afford maximum portability, double quotes should be used consistently around variables, especially when the script itself does not control the variable's value.

Performing Tests

Performing tests is a crucial part of creating useful programs, as was discussed when reviewing programming conventions in Chapter 2, "Programming Constructs."

Tests in the Bourne shell are performed with either the if command or by simply nesting the test expression in square braces ([]).

The if statement in Bourne shell implies its own body; therefore, start and end body markers are not required. Instead, the if must be closed with the fi keyword.

Test expressions understood by the Bourne shell depend on the variable types being tested. Unlike C, Bourne test expressions yield zero (0) if the condition is true.

Table A.1 shows the test expressions understood by Bourne shell and examples of their use.

Data Type	Test	Description	Example
String	=	Determines whether two strings are equal	if ["\$var1" = "\$var2"]; then echo "Match found"fi
	! =	Determines whether two strings are not equal	if ["\$var1" != "\$var2"]; then echo "No Match found" fi
Numeric	- eq	Determines whether two numbers are equal	if [\$ <i>var</i> 1 -eq \$ <i>var</i> 2]; then echo "Numbers are equal fi
	- nq	Sees whether two numbers are not equal	if [\$var1 -nq \$var2]; then echo "Nums not equal fi
	-1t	Sees whether the first number is less than the second	[\$var1 -lt \$var2]; then echo "\$var1 less than" fi
	-gt	Determines whether the first number is greater than a second	if [\$var1 -gt \$var2]; then echo "\$var2 greater than" fi
	-ge	Determines whether the first number is greater than or equal to a second	fi [\$var1 -ge \$var2]; then echo "\$var1 greateror equal" fi
Files	- f	Tests for existence of a file	if [-f "\$filename"]; then echo "File exists" fi
	- d	Tests for existence of a directory	if [-d "\$dirName"]; then echo "Dir exists" fi
	- r	Tests whether a file is readable	if [-r "\$filename"]; then echo "Can read" fi
	- W	Tests whether a file is writeable	if [-w "\$filename"]; then echo "Can write" fi
	- X	Tests whether a file is executable	if [-x "\$filename"]; then echo "Can exec" fi
Boolean	!	Negates a test	if [! -f "\$filename"]; ther echo "\$filename does not exis fi

Table A.1 Bourne Shell Tests

A

able A.1 continued						
Data Type Test		Description	Example			
	- a	Combines test with the Boolean AND function	if[-f "\$filename" -a -w "\$filename"]; then # do something else # do something fi			
	- 0	Combines test with the Boolean 0R function	if[! -f "\$filename" -o -w "\$filename"]; then exit 1 else # do something fi			





Notice the syntax for specifying an else condition to a test as demonstrated in the last line of Table A.1.

Similar to the C switch statement, the Bourne shell provides for establishing actions for multiple conditions using the case statement.

Case Tables

The syntax for performing a case is

```
case "$file" in
*.c) echo "$file is a C source file"
*.h) echo "$file is a C header file"
     ;;
.*) echo "$file is a hidden file"
     ;;
     echo "$file is an unknown file type"
*)
     ;;
```

A feature of the Bourne shell case statement is the capability to use wildcards in the conditions being tested. These fields, as shown in the previous example, are terminated with a double semicolon (;;) following the condition body.

The syntax for conditions in a case statement can optionally be ORed as well

```
case "$input" in
Q*|q*) echo "Exiting..."
       exit 1
     111
*) echo "Input not recognized ($input)"
     ;;
```

The capability to perform looping is also necessary, as it is with any programming language.

Looping

Two loop constructs, the while loop and the for loop, exist in the Bourne shell.

The while Loop

The following is an example of the while loop:

```
cnt=0
while [ $cnt -lt 10 ]
do
    echo $cnt
    cnt=`expr "$cnt" + 1`
done
```

The sample while loop simply counts to 10, printing the value of the variable cnt for each iteration. The use of do and done statements for marking the code body of the loop is important to all loops understood by the Bourne shell.

EXCURSION

Use the expr Command to Perform Mathematical Equations in a Shell Script

The command expr is a standard UNIX command used to manipulate variables in a variety of ways. It is a useful command. Review its man page for a full description.

Because the Bourne shell has no mathematical capability, you must use the expr command for mathematical operations. Give each component of the expression to expr as a separate argument.

In the while loop example, expr is used to add 1 to the variable cnt. Notice the tick marks instructing the shell to interpret the UNIX call before making the assignment. In this way, the return value of the evaluated expr call is used as the new value of cnt.

The next loop to consider is the Bourne shell for loop.

The for Loop

Consider the following example of the for loop:

```
count=0
for file in `ls /dev/tty*`
do
    count=`expr "$count" + 1'
done
echo "$count TTYs found"
```

The for loop expands the 1s command and makes the result a token, assigning one token per loop iteration to the variable file.

One shortcoming of the Bourne shell for loop is that it will always perform one iteration of the loop, even when the evaluated statement yields no tokens. Many other internal commands and features exist in the Bourne shell script. The preceding sections provide an excellent starting point for writing shell scripts.

Writing a Script with Function Calls

Using your favorite editor, open a file and add the necessary first line to instruct the shell which shell interpreter to use.

Then begin entering shell or UNIX commands to accomplish the purpose of the script.



Listing A.3 shows a sample shell script for printing a menu and responding to user input.

```
Listing A.3 Sample Script menu.sh
```

```
1:
       #!/bin/sh
2:
3:
       print_menu() {
4:
         echo
5:
         echo "
                   Sample Menu"
6:
         echo "
                   ----"
         echo "1) Print a file"
7:
         echo "2) List a file"
8:
9:
         echo "3) Remove a file"
         echo "4) Exit"
10:
         echo
11:
12:
         echo -n "Selection: "
13:
14:
       };
15:
16:
       #
17:
       # Shell execution starts here
18:
       #
19:
       forever=0
20:
       while [ $forever -lt 1 ]
21:
       do
22:
         print menu
23:
24:
         #
25:
         # get user input
26:
         read ans
27:
28:
         case "$ans" in
```

A

```
29:
         1) echo -n "Enter filename: "
30:
           read filename
31:
           if [ "$?filename" ]; then
32:
             echo "Printing file ($filename) to default printer..."
33:
              lp "$filename"
34:
           else
35:
             echo "No file specified"
36:
           fi
37:
           sleep 2 # pause so message is read
38:
           ;; # end condition
39:
         2) echo -n "Enter file: "
40:
           read filename
41:
           if [ -f "$filename" ]; then
             ls -l "$filename"
42:
43:
           else
             echo "No file to list..."
44:
45:
           fi
46:
           sleep 2
47:
           ;;
48:
         3) echo -n "Enter file to remove"
49:
           read filename
50:
           if [ -w "$filename" ]; then
51:
             echo -n "Are you sure [yes/no]: "
52:
             read ans
53:
             if [ "$ans" = "yes" ]; then
             rm -f "$filename"
54:
55:
             fi
56:
           else
57:
             echo "You do not have permission to remove $filename"
58:
           fi
59:
           sleep 2
60:
           ;;
         *) echo "Bye!"
61:
62:
           exit 0
63:
           ;;
64:
         esac
65:
       done
66:
       #
67:
       # end of menu.sh
68:
```

Two new conventions appear in Listing A.3, specifically, the print_menu function definition and the commands sleep, read, and exit.

A function is defined in a Bourne shell by providing a name followed by open and close parentheses. Open and close curly braces are used to mark the beginning and ending of the function body.

The function is invoked by simply entering its name:

22: print_menu

The commands sleep, read, and exit perform the function implied by their name.

The sleep Command

The sleep command causes the shell script to pause for the number of seconds specified as the parameter when the command is executed.

37: sleep 2 # pause so message is read

The read Command

The read command fills a variable passed as a parameter with user input. Everything typed by the user until the Return key is pressed is assigned as the variable's value.

26: read ans

The exit Command

The exit command ceases the shell script's execution and returns as the exit status the value specified as the command's parameter.

62: exit 0



Unlike other languages, Bourne shell uses the exit value of zero (0) to indicate success.

This is useful because many errors can cause a shell script to exit. By returning a value greater than zero when there is an error, the value can be used to determine the failure's cause.

When an error occurs because of unexpected behavior of the script, it can be debugged.

Debugging Shell Scripts

The -x flag for placing the shell in debug mode is common to all UNIX shells.

Two ways to employ the flag are by either editing the first line of the shell script to include the $\,\text{-}x$

```
1: #!/bin/sh -x
```

or executing the script within another shell as in

sh -x menu.sh



Running a shell script within a shell, as demonstrated in the previous example for enabling debug, works for executing a script for which you don't have execute permission.

When a shell script is in debug mode, it prints every line prior to execution. This enables you to see the value of variables as they are evaluated by the script.

Appendix B

Application Layout Code Listing

This appendix provides the complete source code listing from Chapter 6, "Components of an X Window Application."

The code is sufficient to create the initial layout of the Graphics Editor project. The intent is to

- Illustrate proper coding structure and C syntax
- Introduce the Graphics Editor application layout
- Show concise modular programming techniques
- Provide examples of X Toolkit Intrinsics programming

Figure B.1 shows the code structure as captured from my development environment.



The directory 2d-editor is the root directory for the project; it contains a source directory called src and an object directory called i86-Linux.

These directories can be created using the commands discussed in Chapter 1, "UNIX for Developers."

make.defines File Contents

Listing B.1 shows the contents of the make.defines file, which should be placed in the 2d-editor project root directory.

```
Listing B.1 make.defines File Contents
```

```
###
1:
2:
       # 2d-gx make.defines
3:
       #
4:
       # This file should be included in each Make file used with 2d-gx
5:
       #
6:
       # Dependencies on the following environment variables:
7:
       # TARGET - machine-os
8:
9:
       # The syntax of this file is for use
       ➡ # with 'gmake' (GNU version of make)
10:
       #
11:
       # See the text for a discussion of this file and its syntax.
12:
       ###
13:
       TARGET SPARC SUNOS
                              = sun4u-SunOS
14:
       TARGET 186 SOLARIS
                              = i86pc-SunOS
       TARGET_i86_LINUX
                              = i86-Linux
15:
16:
17:
       ifdef GxHOME
           GxSRCDIR = ${GxHOME}/src
18:
19:
       else
20:
           GxSRCDIR = .../src
21:
       endif
22:
23:
       vpath %.h ${GxSRCDIR}/include
24:
       vpath %.c ${GxSRCDIR}
25:
26:
       #
27:
       # Configure for Linux running on a PC
28:
29:
       ifeq ($(TARGET),$(TARGET i86 LINUX))
30:
         X11INC
                   = -I/usr/include/X11
31:
         X11LIB
                   = -L/usr/X11R6/lib -L/usr/lib
32:
         INCS
                   = -I${GxSRCDIR}/include ${X11INC}
33:
34:
         CC
                   = gcc
35:
         OPTS
                   = -ansi -Wall -g
36:
       endif
37:
```

```
38:
       #
39:
       # Configure for Solaris running on a Sparc
40:
       #
41:
       ifeq ($(TARGET),$(TARGET SPARC SUNOS))
42:
         X11INC
                   = -I/usr/openwin/include
43:
         X11LIB
                   = -L/usr/openwin/lib
44:
         INCS
                   = -I${GxSRCDIR}/include
45:
46:
         CC
                   = gcc
47:
         OPTS
                   = -g -Wall -ansi
48:
       endif
49:
50:
       #
       # Configure for Solaris running on a PC
51:
52:
       #
53:
       ifeq ($(TARGET),$(TARGET i86 SOLARIS))
54:
         X11INC
                   = -I/usr/openwin/include
55:
         X11LIB
                   = -L/usr/openwin/lib
56:
         INCS
                   = -I${GxSRCDIR}/include
57:
58:
         CC
                   = gcc
59:
         OPTS
                    = -ansi -Wall -g
60:
       endif
61:
62:
       #
       # Force all Makefiles using this file to check the configuration
63:
64:
       # of the environment before building the target
65:
       #
66:
       all: make-env-check make-target
67:
68:
       #
       # Check to environment variables need to build are set
69:
70:
       #
71:
       make-env-check:
72:
           ifndef TARGET
73:
             @echo
74:
             @echo "TARGET not defined!"
75:
             @echo "Set environment variable TARGET to:"
             @echo "
                                          or"
76:
                        sun4u-Sun0S
77:
             @echo "
                        i86pc-SunOS
                                          or"
             @echo "
78:
                        i86-Linux"
79:
             @echo
80:
             @exit 1
           endif
81:
82:
83:
       clean:
           @rm -f *~ *.o $(PROGRAM)
84:
85:
       #
86:
       # end of make.defines
87:
       #
```

As discussed in Chapter 1 where this file was introduced, the variable TARGET must be set in your environment for this portion of the GNUmakefile to work.

GNUmakefile File Contents

Listing B.2 shows the contents of the GNUmakefile, which should be placed in the src directory below the project root.

Listing B.2 GNUmakefile File Contents

```
1:
      #
      # Makefile for 2d-gx
2:
3:
      #
4:
      # See the text for a discussion of this file and its syntax
5:
      ###
6:
7:
      # Read in the system specific environment configuration
8:
      include ../make.defines
9:
10:
       PROGRAM = 2d - gx
11:
12:
       LIBS
               = -1Xaw -1Xt -1X11
13:
       CFLAGS = $(OPTS) $(INCS)
14:
15:
16:
       OBJS =
                  gxMain.o \
17:
                  gxGraphics.o \
18:
                  gxLine.o \
19:
                  qxText.o \
20:
                  gxArc.o \
21:
                  gxGx.o
22:
23:
24:
       make-target: $(PROGRAM)
25:
26:
       $(PROGRAM): $(OBJS)
27:
           @echo "Building $(PROGRAM) for $(TARGET)..."
28:
           $(CC) -o $(PROGRAM) $(OBJS) $(X11LIB) $(LIBS)
           @echo "done"
29:
30:
       #
       # end of Makefile for 2d-gx
31:
32:
       #
```

I Note

It is necessary to create a *symbolic link* to this file in the object directory where the project will be built. See Chapter 1 for a discussion of symbolic links and how to create them.

gxMain.c File Contents

Listing B.3 shows the primary source file gxMain.c, which should be placed in the src directory.

```
Listing B.3 gxMain.c File Contents
```

```
/**
1:
2:
        ** 2D Graphical Editor (2d-gx)
        **
3:
        ** gxMain.c
4:
5:
        */
6:
       #include <stdlib.h>
7:
       #include <stdio.h>
8:
9:
       #include <X11/Intrinsic.h> /* for creation routines */
10:
       #include <X11/StringDefs.h> /* resource names
                                                               */
11:
       #include <X11/Xaw/Form.h>
12:
13:
       #include "gxGraphics.h"
14:
       #include "gxProtos.h"
15:
       /*
16:
17:
        * Entry point for the application
18:
        */
19:
       int main( int argc, char **argv )
20:
       {
21:
         XtAppContext appContext;
22:
         Widget
                       toplevel;
23:
         Widget
                      form;
24:
         Widget
                      canvas;
25:
26:
         toplevel = XtVaAppInitialize( &appContext, "2D Graphical Editor",
27:
                                        NULL, 0, &argc, argv, NULL,
28:
                                        NULL );
29:
30:
         form = XtVaCreateWidget( "topForm",
31:
                                    formWidgetClass, toplevel,
32:
                                    NULL );
33:
34:
         canvas = create canvas( form );
35:
         create_status( form, canvas );
36:
37:
         create_buttons( form );
38:
39:
         XtManageChild( canvas );
40:
         XtManageChild( form );
41:
42:
         XtRealizeWidget( toplevel );
43:
         XtAppMainLoop( appContext );
44:
45:
         exit(0); /* never reached as XtAppMainLoop is infinite */
46:
       }
47:
       /**
48:
       ** end of gxMain.c
49:
50:
        */
```

```
B
```

gxGraphics.c File Contents

Listing B.4 shows the contents of the gxGraphics.c file, which should be placed in the src directory.

```
Listing B.4 gxGraphics.c File Contents
```

```
/**
1:
2:
       ** 2D Graphical Editor (2d-gx)
3:
       * *
4:
       ** gxGraphics.c
       */
5:
6:
       #include <stdio.h>
7:
       #include <X11/Intrinsic.h>
8:
9:
       #include <X11/StringDefs.h>
10:
       #include <X11/Xaw/Form.h>
       #include <X11/Xaw/Command.h>
11:
12:
       #include <X11/Xaw/Box.h>
       #include <X11/cursorfont.h>
13:
14:
15:
       #include "gxGraphics.h"
16:
       #include "gxIcons.h"
17:
       #include "gxBitmaps.h"
18:
19:
       static GxIconData gxDrawIcons[] = {
20:
       { &line icon, (void (*)(void))gx line,
-
       "Draw an elastic line..."
                                  },
21:
       { &pen_icon, (void (*)(void))gx_pencil,
       "Draw a freestyle line..."
-
                                     },
22:
       { &arc_icon, (void (*)(void))gx_arc,
       "Draw a circle..."
₩.
                                     },
23:
       { &box_icon, (void (*)(void))gx_box,
       "Draw a square or rectangle..." },
•
24:
       { &arr_icon, (void (*)(void))gx_arrow,
       "Draw an arrow..."
₩.
                                      },
25:
       { &text_icon, (void (*)(void))gx_text,
                               },
₩.
       "Draw dynamic text..."
       /*----*/
26:
27:
       /* this list MUST be NULL terminated */
28:
       /*----*/
29:
       { NULL },
30:
       };
31:
32:
        /*
        * Create the region of the application where we will draw
33:
34:
        */
35:
       Widget create canvas( Widget parent )
36:
        {
37:
         GxDrawArea = XtVaCreateWidget( "drawingArea",
38:
                                        formWidgetClass, parent,
39:
                                        XtNbackground,
```

40:		WhitePixelO [.]	fScreen(XtScreen(parent)).
41:		XtNtop.	XawChainTop.
42:		XtNleft.	XawChainLeft.
43:		XtNbottom.	XawChainBottom.
44:		XtNright.	XawChainRight.
45:		XtNheight.	220.
46.		XtNwidth	250
47.		NIIII) ·	2003
48.		1022),	
40.	XtAddEventHandler(GyDrawA	rea PointerMoti	onMask False
-50. 50.	(XtEvent	Handler)drawArea	EventProc
►	(XtEventi (XtPointer)		Eventi 100,
51.	(Xtroincer) XtAdEventHandler(GyDrawA	noll), rea ButtonPress	Mack False
50.		lea, buttonriessi Landlon)dnawAnaal	EventBree
52.	(XtEVentr	naliutei jui awai eai	Eventrioc,
-	(ALFOINTER)	NULL);	
53:	XtaddeventHandler(GxDrawA	rea, ButtonRelea	Semask, Faise,
54:	(XTEVenti	Handler) drawArea	EventProc,
•	(XtPointer)	NULL);	
55:			
56:	return GxDrawArea;		
57:	}		
58:			
59:	/*		
60:	* create_status		
61:	*/		
62:	void create_status(Widget pa	arent, Widget fv	ert)
63:	{		
64:	GxStatusBar =		
₩	XtVaCreateManagedWidget("s	statusBar",	
65:		labelWidgetClas	s, parent,
66:		XtNtop,	XawChainBottom,
67:		XtNleft,	XawChainLeft,
68:		XtNbottom,	XawChainBottom,
69:		XtNright,	XawChainRight,
70:		XtNfromVert,	fvert,
71:		XtNborderWidth,	0,
72:		NULL);	
73:	setStatus("2D-GX (c)Starry	y Knight Softwar	e - Ready");
74:	}		
75:	-		
76:	/*		
77:	* statusProc		
78:	*/		
79:	void statusProc(Widget w.		
	XtPointer msg	. XEvent *xe. Bo	olean flag)
80:	{	,	
81:	if(msg == NULL)		
82.	setStatus("\0").		
83.			
84.	cotstatus(men);		
85.	l		
86.	L		

Listing B.4 continued

```
87:
        /*
         *
88:
            create_icons
         */
89:
90:
        void create icons( Widget parent, GxIconData *iconData,
91:
                            void (*callback)( Widget,
91a:
                                               XtPointer, XtPointer ))
92:
        {
93:
          Widget btn;
          Pixmap pix;
94:
95:
96:
          while( iconData->info != NULL ) {
97:
            if( iconData->info->bits != NULL ) {
98:
              pix = create pixmap( parent, iconData->info );
99:
100:
              btn =
             XtVaCreateManagedWidget( "",
-
101:
                                           commandWidgetClass, parent,
102:
                                           XtNwidth, iconData->info->w + 1,
103:
                                           XtNheight, iconData->info->h + 1,
104:
                                           XtNbackgroundPixmap,
                                                                     pix,
105:
                                           XtNhighlightThickness,
                                                                     1,
106:
                                           NULL );
107:
108:
              XtAddEventHandler( btn, EnterWindowMask, False,
109:
                                  (XtEventHandler)statusProc,
-
                              (XtPointer)iconData->mesg);
              XtAddEventHandler( btn, LeaveWindowMask, False,
110:
111:
                                  (XtEventHandler)statusProc,
-
                              (XtPointer)NULL );
112:
113:
              XtAddCallback( btn, XtNcallback, callback,
-
                          (XtPointer)iconData->func );
114:
            }
            /*
115:
116:
              *
               go to the next element
             */
117:
118:
            iconData++;
119:
          }
120:
        }
121:
122:
        /*
123:
         * Create a panel of buttons that will
      * allow control of the application
•
124:
         */
125:
        void create_buttons( Widget parent )
126:
        {
127:
          Widget butnPanel, exitB;
128:
129:
          /*
130:
           * create a panel for the drawing icons
131:
           */
```

```
132:
          butnPanel = XtVaCreateWidget( "drawButnPanel",
133:
                                           boxWidgetClass, parent,
134:
                                                              XawChainTop,
                                           XtNtop,
135:
                                           XtNright,
                                                              XawChainRight,
136:
                                           XtNbottom,
                                                              XawChainTop,
137:
                                           XtNleft,
                                                              XawChainRight,
138:
                                           XtNhorizDistance, 10,
139:
                                           XtNfromHoriz.
                                                              GxDrawArea,
140:
                                           XtNhSpace,
                                                              1,
                                                              1,
141:
                                           XtNvSpace,
142:
                                           NULL );
143:
144:
          create icons( butnPanel, gxDrawIcons, draw manager );
          XtManageChild( butnPanel );
145:
146:
          exitB = XtVaCreateManagedWidget( " Exit ",
147:
                                             commandWidgetClass, parent,
148:
149:
                                             XtNtop,
                                                            XawChainBottom,
150:
                                             XtNbottom,
                                                            XawChainBottom,
151:
                                             XtNleft,
                                                            XawChainRight,
152:
                                                            XawChainRight,
                                             XtNright,
153:
                                             XtNfromVert,
                                                            butnPanel,
                                             XtNfromHoriz, GxStatusBar,
154:
155:
                                             NULL );
156:
157:
          XtAddCallback( exitB, XtNcallback, gx exit, NULL );
158:
        }
159:
160:
        /*
170:
         * create_pixmap
         */
171:
        Pixmap create_pixmap( Widget w, XbmDataPtr data )
172:
173:
        {
174:
          return( XCreatePixmapFromBitmapData( XtDisplay(w),
175:
                                            DefaultRootWindow(XtDisplay(w)),
176:
                                            (char *)data->bits,
177:
                                            data->w, data->h,
178:
                                            BlackPixelOfScreen(XtScreen(w)),
179:
                                            WhitePixelOfScreen(XtScreen(w)),
180:
                                            DefaultDepthOfScreen(XtScreen(w)));
181:
        }
182:
        /**
183:
184:
         ** end of gxGraphics.c
         */
185:
```

gxGx.c File Contents

Listing B.5 shows the contents of the file gxGx.c, which should be placed in the src directory.

```
Listing B.5 gxGx.c File Contents
```

```
/**
1:
2:
        ** 2D Graphical Editor (2d-gx)
        **
3:
        ** gxGx.c
4:
5:
        */
6:
       #include <stdio.h>
7:
8:
       #include "gxGraphics.h"
9:
       #include "gxProtos.h"
10:
11:
       #include <X11/Xaw/Label.h>
12:
13:
       static void (*draw mgr func)( XEvent * ) = NULL;
14:
15:
       /*
        * gx_exit
16:
17:
        */
18:
       void gx exit( Widget w, XtPointer cd, XtPointer cbs )
19:
       {
20:
         exit(0);
21:
       }
22:
23:
       /*
        * setStatus
24:
        */
25:
26:
       void setStatus( char *message )
27:
       {
28:
         XtVaSetValues( GxStatusBar, XtNlabel, message, NULL );
29:
       }
30:
31:
       /*
32:
       * draw_manager
33:
        */
34:
       void draw_manager( Widget w, XtPointer cd, XtPointer cbs )
35:
       {
36:
         void (*draw_func)( XEvent * ) = (void (*)(XEvent *))cd;
37:
38:
         if( draw_func != NULL ) (*draw_func)( NULL );
39:
         draw mgr func = draw func;
40:
       }
41:
42:
       /*
43:
        * drawAreaEventProc
44:
        */
45:
       void drawAreaEventProc( Widget w, XtPointer cd,
•
                              XEvent *event, Boolean flag )
46:
       {
47:
         if( draw mgr func != NULL ) (*draw mgr func)( event );
48:
       }
49:
```

50: 51: /** 52: ** end of gxGx.c 53: */

gxArc.c File Contents

Listing B.6 shows the contents of the file gxArc.c, which should be placed in the src directory.

Listing B.6 gxArc.c File Contents

```
/**
1:
        ** 2D Graphical Editor (2d-gx)
2:
3:
        * *
        ** gxArc.c
4:
        */
5:
6:
       #include <stdio.h>
7:
8:
       void gx_arc( void )
9:
       {
         printf( "draw an arc...\n" );
10:
11:
       }
12:
       /**
13:
14:
        ** end of gxArc.c
15:
        */
```

Clearly, the files representing the various objects in the Graphics Editor project are simply stubs. Beginning with Chapter 13, "Application Structure," the text will advance these files.

gxLine.c File Contents

Listing B.7 shows the contents of the gxLine.c file, which should be placed in the src directory.

Listing B.7 gxLine.c File Contents

```
/**
1:
        ** 2D Graphical Editor (2d-gx)
2:
        * *
3:
4:
        ** gxLine.c
        */
5:
6:
       #include <stdio.h>
7:
       #include <X11/Intrinsic.h>
8:
9:
       void gx line( XEvent *event )
10:
       {
         printf( "draw a line...\n" );
11:
12:
       }
```

B

Listing B.7 continued

```
13:
14:
       void gx_pencil( XEvent *event )
15:
       {
16:
         printf( "draw freestlye\n" );
17:
       }
18:
19:
       void gx_arrow( XEvent *event )
20:
       {
21:
         printf( "draw an arrow\n" );
22:
       }
23:
24:
       void gx_box( XEvent *event )
25:
       {
26:
         printf( "draw a box\n" );
27:
       }
28:
29:
       /**
        ** end of gxLine.c
30:
        */
31:
```

gxText.c File Contents

Listing B.8 shows the file contents of gxText.c. It, too, should be placed in the src directory.

Listing B.8 gxText.c File Contents

```
1:
       /**
        ** 2D Graphical Editor (2d-gx)
2:
        **
3:
4:
        ** gxText.c
        */
5:
6:
       #include <stdio.h>
7:
8:
       void gx text( void )
9:
       {
10:
         printf( "draw text...\n" );
11:
       }
12:
13:
       /**
        ** end of gxText.c
14:
15:
        */
```

gxGraphics.h File Contents

The gxGraphics.h file is the first header file introduced in the application layout. Its placement will require creating a new directory under src called include.

The contents of the file gxGraphics.h are shown in Listing B.9.

Listing B.9 gxGraphics.h File Contents

```
/**
1:
2:
        ** 2D Graphical Editor (2d-gx)
3:
        **
        ** gxGraphics.h
4:
5:
        */
       #include <X11/Intrinsic.h>
6:
7:
8:
       #ifndef _GX_GRAPHICS_H_INC_
9:
       #define _GX_GRAPHICS_H_INC_
10:
       #endif /* _GX_GRAPHICS_H INC */
11:
12:
13:
       #ifndef GLOBAL
14:
       #define GLOBAL
15:
       #else
       #undef GLOBAL
16:
17:
       #define GLOBAL extern
18:
       #endif
19:
20:
       GLOBAL Widget GxStatusBar;
21:
       GLOBAL Widget GxDrawArea;
22:
23:
       /**
        ** end of gxGraphics.h
24:
        */
25:
```

gxIcons.h File Contents

Listing B.10 shows the contents of the header file gxIcons.h. The placement of this file is in the include directory in src.

```
Listing B.10 gxIcons.h File Contents
```

```
1:
       /**
2:
        ** 2D Graphical Editor (2d-gx)
        **
3:
4:
        ** gxIcons.h
5:
        */
       #ifndef GX ICONS H INC
6:
7:
       q#define _GX_ICONS_H_INC_
8:
9:
       #include "gxProtos.h"
10:
11:
       /*
        * Storage for pertinent XBM data
12:
13:
        */
14:
       typedef struct _xbm_data {
15:
         unsigned char *bits;
16:
         int
                       w, h;
17:
       } XbmData, *XbmDataPtr;
```

B

Listing B.10 continued

```
18:
19:
       /*
20:
        * IconData necessary to create icon
        */
21:
22:
       typedef struct _gx_icon_data {
23:
         XbmDataPtr info;
24:
25:
         void (*func)(void);
26:
         char *mesg;
27:
       } GxIconData, *GxIconDataPtr;
28:
29:
       #define icon static( name, bits, width, height ) \
30:
        static XbmData name = { bits, width, height }
31:
32:
33:
34:
       /* prototypes */
                     create_icons ( Widget, GxIconDataPtr,
35:
       extern void
36:
                                      void (*)(Widget,
                                             XtPointer, XtPointer ));
₩.
37:
       extern Pixmap create_pixmap ( Widget, XbmDataPtr
                                                             );
38:
39:
       #endif /* _GX_ICONS_H_INC_ */
40:
41:
       /**
        ** end of gxIcons.h
42:
        */
43:
```

gxBitmaps.h File Contents

Listing B.11 shows the file contents of the header file gxBitmaps.h, which should be placed in the include directory under src.

```
Listing B.11 gxBitmaps.h File Contents
```

```
1:
     /**
      ** 2D Graphical Editor (2d-gx)
2:
      **
3:
4:
      ** gxBitmaps.h
5:
      */
6:
     #ifndef GX BITMAPS H INC
7:
     #define _GX_BITMAPS_H_INC_
8:
9:
     #include "gxIcons.h"
10:
11:
      /*
       * drawing icons
12:
13:
       */
14:
      static unsigned char line bits[] = {
15:
```

16: 17: 18: 19: 0x00,0x00,0x88,0x00,0x14,0x00,0x00,0x84,0x00,0x14,0x00, 20: 21: 0x02,0x02,0x22,0x00,0x00,0x01,0x02,0x42,0x00,0x00,0x01, 22: 0x04,0x42,0x00,0x80,0x00,0x04,0x42,0x00,0x80,0x00,0x02, 23: 0x01,0x00,0x40,0x00,0x01,0x01,0x00,0x40,0x80,0x00,0x01, 24: 0x00,0x20,0x40,0x00,0x01,0x00,0x20,0x20,0x00,0x01,0x00, 25: 26: 27: 28: 29: $0 \times 00, 0 \times 00 \};$ 30: icon static(line icon, line bits, 36, 32); 31: 32: static unsigned char pencil_bits[] = { 33: $0 \times 00, 0 \times$ 34: $0 \times 00, 0 \times c0, 0 \times 00, 0 \times 00, 0 \times 00, 0 \times 00, 0 \times e0, 0 \times 01, 0 \times 00, 0 \times 00, 0 \times 00, 0 \times d0, 0 \times d0, 0 \times 00, 0 \times 00, 0 \times d0, 0 \times 00, 0 \times$ 35: 36: 37: 38: 39: 40: 41: 42: 0x00,0x00,0x00,0x30,0x01,0x00,0x00,0x28,0xf0,0x00,0x00,0x00, 43: 0x44,0x10,0x00,0x00,0x00,0x84,0x20,0x00,0x00,0x00,0x04,0x41, 0x00,0x00,0x00,0x08,0x42,0x00,0x00,0x00,0x10,0x44,0x00,0x00, 44: 45: 46: $0 \times 00, 0 \times 00, 0 \times 00, 0 \times 00$; 47: icon_static(pen_icon, pencil_bits, 36, 32); 48: 49: static unsigned char arc bits[] = { 50: 51: $0 \times 00, 0 \times$ 52: 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x3f,0x00,0x00,0x00, 53: 0x70,0xc0,0x01,0x00,0x00,0x0c,0x00,0x06,0x00,0x00,0x02,0x00, 54: 55: 0x40,0x00,0x00,0x40,0x00,0x40,0x00,0x04,0x40,0x00,0x20,0x00, 56: 0x04,0x80,0x00,0x20,0x00,0x1f,0x80,0x00,0x20,0x00,0x04,0x80, 57: 0x00,0x40,0x00,0x04,0x40,0x00,0x40,0x00,0x00,0x40,0x00,0x80, 58: 0x08,0x00,0x00,0x0c,0x00,0x86,0x00,0x00,0x70,0xc0,0x81,0x00, 59: 60: 61: $0 \times 00, 0 \times 80, 0 \times 00, 0 \times$ 62: 63: $0 \times 00, 0 \times 00, 0 \times 00, 0 \times 00$; 64: icon_static(arc_icon, arc_bits, 36, 32); 65: 66: static unsigned char box bits[] = { 67: $0 \times 00, 0 \times$
Listing B.11 continued

68:	0×00,0×00,0×00,0×00,0×00,0×00,0×00,0×0
69:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
70:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0xff,0xff
71:	0x00,0x80,0x00,0x00,0x10,0x00,0x80,0x00,0x0
72:	0x80,0x00,0x00,0x10,0x00,0x80,0x00,0x00,
73:	0x00,0x00,0x10,0x00,0x80,0x00,0x00,0x10,0x00,0x80,0x00,
74:	0x00,0x10,0x00,0x80,0x00,0x00,0x10,0x00,0x80,0x00,0x0
75:	0x10,0x00,0x80,0x00,0x00,0x10,0x00,0x80,0x00,0x0
76:	0x00,0x80,0x00,0x00,0x10,0x00,0x80,0x00,0x0
77:	0x80,0x00,0x00,0x10,0x00,0x80,0xff,0xff,0x1f,0x00,0x00,
78:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
79:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
80:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
81:	0×00,0×00,0×00,0×00,0×00,0×00};
82:	<pre>icon_static(box_icon, box_bits, 36, 32);</pre>
83:	
84:	<pre>static unsigned char arrow_bits[] = {</pre>
85:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
86:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
87:	0x00,0x00,0x80,0x02,0x00,0x00,0x00,0x40,0x04,0x00,0x00
88:	0x20,0x08,0x00,0x00,0x00,0x10,0x10,0x00,0x0
89:	0x00,0x00,0x00,0x04,0x40,0x00,0x00,0x00
90:	0x00,0x01,0x00,0x01,0x00,0x80,0x00,0x00,
91:	0x00,0x04,0x00,0x20,0x00,0x00,0x08,0x00,0xe0,0x07,0xc0,0x0f,
92:	0x00,0x00,0x04,0x40,0x00,0x00,0x00,0x04,0x40,0x00,0x00,0x00,
93:	0x04,0x40,0x00,0x00,0x00,0x04,0x40,0x00,0x00,0x00,0x00,0x04,0x40,
94:	0x00,0x00,0x00,0x04,0x40,0x00,0x00,0x00
95:	0x00,0x02,0x80,0x00,0x00,0xc0,0x01,0x00,0x07,0x00,0x00,0x00,
96:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
97:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
98:	0×00,0×00,0×00,0×00};
99:	icon_static(arr_icon, arrow_bits, 36, 32);
100:	
101:	<pre>static unsigned char text_bits[] = {</pre>
102:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
103:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
104:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
105:	0x1f,0x00,0x80,0x83,0x1f,0x1c,0x00,0x80,0x01,0x0f,0x18,
106:	0x00,0x80,0x00,0x0f,0x10,0x00,0x00,0x00,0x0f,0x00,0x00
107:	0x00,0x00,0x0f,0x00,0x00,0x00,0x00,0x0f,0x00,0x00,0x00,
108:	0x00,0x0f,0x00,0x00,0x00,0x00,0x0f,0x00,0x00,0x00,0x00,0x00,
109:	0x0f,0x00,0x00,0x00,0x00,0x0f,0x00,0x00
110:	0x00,0x00,0x00,0x00,0x0f,0x00,0x00,0x00
111:	0x00,0x00,0x00,0x0f,0x00,0x00,0x00,0x00
112:	0x00,0x00,0x0f,0x00,0x00,0x00,0x00,0x0f,0x00,0x00,0x00,
113:	0x80,0x1f,0x00,0x00,0x00,0xe0,0x7f,0x00,0x00,0x00,0x00,
114:	0×00,0×00,0×00,0×00,0×00,0×00,0×00,0×00,0×00,0×00,0×00,0×00,
115:	0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
116:	0×00,0×00,0×00,0×00,0×00,0×00};
117:	<pre>icon_static(text_icon, text_bits, 36, 32);</pre>

```
118:
119: #endif /* _GX_BITMAPS_H_INC_ */
120:
121: /**
122: ** end of gxIcons.h
123: */
```

gxProtos.h File Contents

Listing B.12 shows the file contents of gxProtos.h, which should be placed with the other header files in the include directory under src.

```
Listing B.12 gxProtos.h File Contents
```

```
1:
       /**
2:
        ** 2D Graphical Editor (2d-gx)
        **
3:
4:
        ** gxProtos.h
        */
5:
6:
       #include <X11/Intrinsic.h>
7:
       #include "gxIcons.h"
8:
       #ifndef EXTERN
9:
10:
       #define EXTERN
11:
       #else
12:
       #undef EXTERN
13:
       #define EXTERN extern
14:
       #endif
15:
16:
       /*
        * Creations routines in gxGraphics.c
17:
18:
       */
       EXTERN Widget create canvas ( Widget );
19:
20:
                    create status ( Widget, Widget );
       EXTERN void
       EXTERN void create_buttons( Widget );
21:
22:
       EXTERN void drawAreaEventProc(Widget, XtPointer,
                                      XEvent *, Boolean );
•
23:
24:
       /*
25:
       * Drawing functions used in GxIcons.h
26:
        */
27:
       EXTERN void gx line( void );
28:
       EXTERN void gx_pencil( void );
29:
       EXTERN void gx arc( void );
30:
       EXTERN void gx box( void );
31:
       EXTERN void gx arrow( void );
32:
       EXTERN void gx_text( void );
33:
       /*
34:
35:
       * Control functions used in GxIcons.h
36:
        */
```

Listing B.12 continued

```
37:
       EXTERN void gx_exit( Widget, XtPointer, XtPointer );
38:
39:
       /*
       * Utilities in gxGx.c
40:
41:
       */
42:
      EXTERN void setStatus( char * );
43:
       EXTERN void draw_manager( Widget, XtPointer, XtPointer );
44:
       EXTERN void ctrl_manager( Widget, XtPointer, XtPointer );
45:
46:
       /**
      ** end of gxProtos.h
47:
48:
       */
```

Appendix C

In this Appendix

- Script Simple Vector Font Set
- Triplex Bold Vector Font Set
- Triplex Bold Italic Vector Font Set
- The vector_chars.h Header File

Additional Vector Font Sets and vector_chars.h

The following sections introduce several additional groupings of vector font sets that can be used in the Graphics Editor. This appendix also provides the definition of the contents of the vector_chars.h header file needed by the Text object introduced in Chapter 24, "Vector Text Object."

Script Simple Vector Font Set

Listing C.1 defines a simple italic-style vector font set. It can be used in lieu of the simplex font introduced in Chapter 24.

```
Listing C.1 The script_simplex.h Font Definition
```

```
1:
    #ifndef S SIM INC H
2:
   #define S_SIM_INC_H
3:
4:
   #include "vfonts/vector chars.h"
5:
6: XPoint **script simplex[] = {
7:
   char699, char2764, char2778, char733, char2769, char2271,
8: char2768, char2767, char2771, char2772, char2773, char725,
9: char2761, char724, char710, char2770, char2750, char2751,
10: char2752, char2753, char2754, char2755, char2756, char2757,
11: char2758, char2759, char2762, char2763, char2241, char726,
12: char2242, char2765, char2273, char551, char552, char553,
13: char554, char555, char556, char557, char558, char559,
14: char560, char561, char562, char563, char564, char565,
15: char566, char567, char568, char569, char570, char571,
```

continues

```
16:
    char572, char573, char574, char575, char576, char2223,
17: char804, char2224, char2262, char999, char2766, char651,
18:
    char652, char653, char654, char655, char656, char657,
19: char658, char659, char660, char661, char662, char663,
20: char664, char665, char666, char667, char668, char669,
21: char670, char671, char672, char673, char674, char675,
22: char676, char2225, char723, char2226, char2246, char718,
23: };
24:
25: int *script simplex p[] = {
26: char_p699, char_p2764, char_p2778, char_p733, char_p2769,
27: char p2271, char p2768, char p2767, char p2771, char p2772,
28: char p2773, char p725, char p2761, char p724, char p710,
    char_p2770, char_p2750, char_p2751, char_p2752, char_p2753,
29:
30:
    char_p2754, char_p2755, char_p2756, char_p2757, char_p2758,
31: char p2759, char p2762, char p2763, char p2241, char p726,
    char p2242, char p2765, char p2273, char p551, char p552,
32:
33: char_p553, char_p554, char_p555, char_p556, char_p557,
34: char p558, char p559, char p560, char p561, char p562,
35: char p563, char p564, char p565, char p566, char p567,
36:
    char_p568, char_p569, char_p570, char_p571, char_p572,
37: char p573, char p574, char p575, char p576, char p2223,
38: char_p804, char_p2224, char_p2262, char_p999, char_p2766,
39:
    char_p651, char_p652, char_p653, char_p654, char_p655,
40: char_p656, char_p657, char_p658, char_p659, char_p660,
    char p661, char p662, char p663, char p664, char p665,
41:
42: char_p666, char_p667, char_p668, char_p669, char_p670,
43:
    char p671, char p672, char p673, char p674, char p675,
44: char p676, char p2225, char p723, char p2226, char p2246,
45: char p718,
46: };
47:
48: #endif /* S SIM INC H */
```

Triplex Bold Vector Font Set

Listing C.2 introduces a complex vector font that appears as elaborate boldface font.

Listing C.2 The triplex_bold.h Font Definition

```
#ifndef TRI_BLD_INC_H
1:
2:
    #define TRI BLD INC H
3:
4:
    #include "vfonts/vector chars.h"
5:
6:
   XPoint **plain triplex bold[] = {
7:
     char3199, char3214, char3228, char2275, char3219, char2271,
     char3218, char3217, char3221, char3222, char3223, char3225,
8:
9:
     char3211, char3224, char3210, char3220, char3200, char3201,
```

```
10: char3202, char3203, char3204, char3205, char3206, char3207,
     char3208, char3209, char3212, char3213, char2241, char3226,
11:
     char2242, char3215, char2273, char3001, char3002, char3003,
12:
13:
    char3004, char3005, char3006, char3007, char3008, char3009,
14: char3010, char3011, char3012, char3013, char3014, char3015,
    char3016, char3017, char3018, char3019, char3020, char3021,
15:
16: char3022, char3023, char3024, char3025, char3026, char2223,
17:
    char804, char2224, char2262, char999, char3216, char3101,
     char3102, char3103, char3104, char3105, char3106, char3107,
18:
19:
     char3108, char3109, char3110, char3111, char3112, char3113,
20:
     char3114, char3115, char3116, char3117, char3118, char3119,
     char3120, char3121, char3122, char3123, char3124, char3125,
21:
22:
     char3126, char2225, char2229, char2226, char2246, char3229,
23: \};
24:
25: int *plain triplex bold p[] = {
26:
     char_p3199, char_p3214, char_p3228, char_p2275, char_p3219,
27:
    char p2271, char p3218, char p3217, char p3221, char p3222,
28:
    char_p3223, char_p3225, char_p3211, char_p3224, char_p3210,
29:
    char_p3220, char_p3200, char_p3201, char_p3202, char_p3203,
    char p3204, char p3205, char p3206, char p3207, char p3208,
30:
31:
    char p3209, char p3212, char p3213, char p2241, char p3226,
32:
    char_p2242, char_p3215, char_p2273, char_p3001, char_p3002,
33:
     char_p3003, char_p3004, char_p3005, char_p3006, char_p3007,
34:
    char p3008, char p3009, char p3010, char p3011, char p3012,
35: char_p3013, char_p3014, char_p3015, char_p3016, char_p3017,
36:
    char p3018, char p3019, char p3020, char p3021, char p3022,
37:
    char p3023, char p3024, char p3025, char p3026, char p2223,
    char p804, char p2224, char p2262, char p999, char p3216,
38:
     char p3101, char p3102, char p3103, char p3104, char p3105,
39:
40:
     char_p3106, char_p3107, char_p3108, char_p3109, char_p3110,
41:
    char_p3111, char_p3112, char_p3113, char_p3114, char_p3115,
    char p3116, char p3117, char p3118, char p3119, char p3120,
42:
43:
     char p3121, char p3122, char p3123, char p3124, char p3125,
     char_p3126, char_p2225, char_p2229, char_p2226, char_p2246,
44:
45:
    char p3229,
46: };
47:
48: #endif /* TRI BLD INC H */
```

Triplex Bold Italic Vector Font Set

Similar in appearance to the vector font introduced in Listing C.2, the triplex bold italic font found in Listing C.3 can be used for the Text object introduced in Chapter 24.

```
Listing C.3 The triplex_bold_italic.h Font Definition
```

```
1: #ifndef TRI_BLD_ITL_INC_H
2: #define TRI_BLD_ITL_INC_H
3:
4: #include "vfonts/vector_chars.h"
```

```
5:
    XPoint **triplex bold italic[] = {
6:
7:
     char3249, char3264, char3278, char2275, char3269, char2271,
8:
     char3268, char3267, char3271, char3272, char3273, char3275,
9:
     char3261, char3274, char3260, char3270, char3250, char3251,
     char3252, char3253, char3254, char3255, char3256, char3257,
10:
     char3258, char3259, char3262, char3263, char2241, char3276,
11:
     char2242, char3265, char2273, char3051, char3052, char3053,
12:
     char3054, char3055, char3056, char3057, char3058, char3059,
13:
14:
     char3060, char3061, char3062, char3063, char3064, char3065,
15:
     char3066, char3067, char3068, char3069, char3070, char3071,
16:
     char3072, char3073, char3074, char3075, char3076, char2223,
     char804, char2224, char2262, char999, char3266, char3151,
17:
18:
     char3152, char3153, char3154, char3155, char3156, char3157,
19:
     char3158, char3159, char3160, char3161, char3162, char3163,
     char3164, char3165, char3166, char3167, char3168, char3169,
20:
     char3170, char3171, char3172, char3173, char3174, char3175,
21:
     char3176, char2225, char2229, char2226, char2246, char3279,
22:
23: };
24:
25: int *triplex bold italic p[] = {
26:
     char_p3249, char_p3264, char_p3278, char_p2275, char_p3269,
     char p2271, char p3268, char p3267, char p3271, char p3272,
27:
    char p3273, char p3275, char p3261, char p3274, char p3260,
28:
29:
     char_p3270, char_p3250, char_p3251, char_p3252, char_p3253,
30:
     char_p3254, char_p3255, char_p3256, char_p3257, char_p3258,
31:
     char p3259, char p3262, char p3263, char p2241, char p3276,
     char p2242, char p3265, char p2273, char p3051, char p3052,
32:
     char_p3053, char_p3054, char_p3055, char_p3056, char_p3057,
33:
34:
     char p3058, char p3059, char p3060, char p3061, char p3062,
     char p3063, char p3064, char p3065, char p3066, char p3067,
35:
     char_p3068, char_p3069, char_p3070, char_p3071, char_p3072,
36:
     char p3073, char p3074, char p3075, char p3076, char p2223,
37:
     char_p804, char_p2224, char_p2262, char_p999, char_p3266,
38:
     char_p3151, char_p3152, char_p3153, char_p3154, char_p3155,
39:
40:
     char_p3156, char_p3157, char_p3158, char_p3159, char_p3160,
     char p3161, char p3162, char p3163, char p3164, char p3165,
41:
42:
     char_p3166, char_p3167, char_p3168, char_p3169, char_p3170,
43:
     char p3171, char p3172, char p3173, char p3174, char p3175,
     char p3176, char p2225, char p2229, char p2226, char p2246,
44:
45:
     char p3279,
46: };
47:
49: #endif /* TRI BLD ITL INC H */
```

The vector_chars.h Header File

The content of the vector_chars.h header file is found in Listing C.4. It provides the character, segment, and point definitions for all the vector font sets introduced in this appendix as well as in Chapter 24.

```
Listing C.4 The vector_chars.h File
```

```
1:
                                #ifndef VEC FONTS INC H
2:
                               #define VEC_FONTS_INC_H
3:
4:
                               #include <X11/Xlib.h>
5:
6:
                               static XPoint seg0_501[] = {
7:
                                                     \{0, -12\}, \{-8, 9\},\
8:
                               };
9:
                               static XPoint seg1 501[] = {
10:
                                                     \{0, -12\}, \{8, 9\},\
11:
                               };
12:
                               static XPoint seg2 501[] = {
13:
                                                     \{-5,2\},\{5,2\},
14:
                               };
15:
                               static XPoint *char501[] = {
16:
                                                     seg0 501, seg1 501, seg2 501,
17:
                                                    NULL,
18:
                               };
19:
                               static int char_p501[] = {
20:
                                                    XtNumber(seg0_501),XtNumber(seg1_501),XtNumber(seg2_501),
21:
                                };
22:
                               static XPoint seg0 502[] = {
23:
                                                     \{-7, -12\}, \{-7, 9\},\
24:
                               };
25:
                               static XPoint seg1 502[] = {
26:
                                                     \{-7, -12\}, \{2, -12\}, \{5, -11\}, \{6, -10\}, \{7, -8\}, \{7, -6\}, \{6, -4\}, \{5, -3\}, \{6, -10\}, \{6, -10\}, \{7, -6\}, \{6, -4\}, \{5, -3\}, \{6, -10\}, \{6, -10\}, \{7, -10\}, \{6, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, 
27:
                                                     \{2, -2\},\
28:
                                };
29:
                                static XPoint seg2 502[] = {
30:
                                                     \{-7, -2\}, \{2, -2\}, \{5, -1\}, \{6, 0\}, \{7, 2\}, \{7, 5\}, \{6, 7\}, \{5, 8\},
31:
                                                     \{2,9\},\{-7,9\},\
32:
                               };
33:
                               static XPoint *char502[] = {
34:
                                                     seg0_502, seg1_502, seg2_502,
35:
                                                    NULL,
36:
                                };
37:
                                static int char p502[] = {
38:
                                                    XtNumber(seg0_502),XtNumber(seg1_502),XtNumber(seg2_502),
39:
                                };
40:
                               static XPoint seg0 503[] = {
41:
                                                     \{8, -7\}, \{7, -9\}, \{5, -11\}, \{3, -12\}, \{-1, -12\}, \{-3, -11\}, \{-5, -9\},
42:
                                                     \{-6, -7\}, \{-7, -4\}, \{-7, 1\}, \{-6, 4\}, \{-5, 6\}, \{-3, 8\}, \{-1, 9\}, \{3, 9\}, \{-6, -7\}, \{-7, -4\}, \{-7, -1\}, \{-6, -7\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{-7, -4\}, \{
43:
                                                     \{5,8\},\{7,6\},\{8,4\},
44:
                               };
45:
                               static XPoint *char503[] = {
46:
                                                     seg0 503,
47:
                                                    NULL,
48:
                                };
49:
                               static int char_p503[] = {
50:
                                                    XtNumber(seg0_503),
51:
                               };
```

```
Listing C.4 Continued
```

```
52:
                    static XPoint seg0 504[] = {
53:
                                 \{-7, -12\}, \{-7, 9\},\
54:
                   };
55:
                   static XPoint seg1_504[] = {
56:
                                 \{-7, -12\}, \{0, -12\}, \{3, -11\}, \{5, -9\}, \{6, -7\}, \{7, -4\}, \{7, 1\}, \{6, 4\},
57:
                                 \{5,6\},\{3,8\},\{0,9\},\{-7,9\},
58:
                   };
59:
                   static XPoint *char504[] = {
60:
                                 seg0_504, seg1_504,
61:
                                 NULL,
62:
                   };
63:
                   static int char p504[] = {
64:
                                 XtNumber(seg0_504),XtNumber(seg1_504),
65:
                    };
66:
                   static XPoint seg0_505[] = {
67:
                                 \{-6, -12\}, \{-6, 9\},\
68:
                    };
69:
                   static XPoint seg1_505[] = {
70:
                                 \{-6, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7, -12\}, \{7,
71:
                   };
72:
                   static XPoint seg2 505[] = {
73:
                                 \{-6, -2\}, \{2, -2\}, 
74:
                   };
                   static XPoint seg3_505[] = {
75:
76:
                                 \{-6,9\},\{7,9\},\
77:
                   };
78:
                   static XPoint *char505[] = {
79:
                                 seg0 505, seg1 505, seg2 505, seg3 505,
80:
                                 NULL,
81:
                   };
82:
                   static int char p505[] = {
83:
                                 XtNumber(seg0 505),XtNumber(seg1 505),XtNumber(seg2 505),
84:
                                 XtNumber(seg3 505),
85:
                    };
86:
                   static XPoint seg0 506[] = {
87:
                                 \{-6, -12\}, \{-6, 9\},\
88:
                   };
89:
                   static XPoint seg1_506[] = {
90:
                                 \{-6, -12\}, \{7, -12\}, 
91:
                   };
                   static XPoint seg2 506[] = {
92:
93:
                                 \{-6, -2\}, \{2, -2\},\
94:
                   };
95:
                   static XPoint *char506[] = {
96:
                                 seg0_506, seg1_506, seg2_506,
97:
                                 NULL,
98:
                   };
99:
                    static int char p506[] = {
100:
                             XtNumber(seg0_506),XtNumber(seg1_506),XtNumber(seg2_506),
101: };
```

```
102:
                  static XPoint seg0_507[] = {
103:
                                \{8, -7\}, \{7, -9\}, \{5, -11\}, \{3, -12\}, \{-1, -12\}, \{-3, -11\},
104:
                                \{-5, -9\}, \{-6, -7\}, \{-7, -4\}, \{-7, 1\}, \{-6, 4\}, \{-5, 6\}, \{-3, 8\},
105:
                                \{-1,9\},\{3,9\},\{5,8\},\{7,6\},\{8,4\},\{8,1\},
106:
                  };
107:
                   static XPoint seg1 507[] = {
108:
                                \{3,1\},\{8,1\},\
109: };
                   static XPoint *char507[] = {
110:
111:
                                seg0_507, seg1_507,
112:
                                NULL,
113:
                  };
114:
                  static int char p507[] = {
115:
                               XtNumber(seg0_507),XtNumber(seg1_507),
116: };
117:
                  static XPoint seg0 508[] = {
118:
                                \{-7, -12\}, \{-7, 9\},\
119: };
120:
                  static XPoint seg1_508[] = {
121:
                               \{7, -12\}, \{7, 9\},\
122:
                };
123:
                  static XPoint seg2 508[] = {
124:
                                \{-7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, 
125: };
126:
                   static XPoint *char508[] = {
127:
                                seg0_508, seg1_508, seg2_508,
128:
                                NULL,
129:
                  };
                  static int char_p508[] = {
130:
131:
                                XtNumber(seg0 508), XtNumber(seg1 508), XtNumber(seg2 508),
132:
                  };
133:
                  static XPoint seg0_509[] = {
134:
                                \{0, -12\}, \{0, 9\},\
135:
                  };
136:
                  static XPoint *char509[] = {
137:
                                seg0 509,
138:
                                NULL,
139:
                   };
140:
                   static int char p509[] = {
141:
                                XtNumber(seg0_509),
142:
                   };
                  static XPoint seg0 510[] = {
143:
144:
                                \{4, -12\}, \{4, 4\}, \{3, 7\}, \{2, 8\},\
145:
                                \{0,9\},\{-2,9\},\{-4,8\},\{-5,7\},\{-6,4\},\{-6,2\},
146:
                  };
147:
                   static XPoint *char510[] = {
148:
                                seg0_510,
149:
                                NULL,
150:
                  };
151:
                  static int char p510[] = {
152:
                                XtNumber(seg0_510),
153: };
```

```
Listing C.4 Continued
```

```
154:
      static XPoint seg0 511[] = {
155:
           \{-7, -12\}, \{-7, 9\},\
156:
      };
157:
      static XPoint seg1 511[] = {
158:
           \{7, -12\}, \{-7, 2\},\
159:
      };
      static XPoint seg2 511[] = {
160:
161:
           \{-2, -3\}, \{7, 9\},\
162:
      };
      static XPoint *char511[] = {
163:
164:
           seg0_511, seg1_511, seg2_511,
165:
           NULL,
166:
      };
167:
      static int char_p511[] = {
168:
           XtNumber(seg0_511),XtNumber(seg1_511),XtNumber(seg2_511),
169:
      };
170:
      static XPoint seg0 512[] = {
171:
           \{-6, -12\}, \{-6, 9\},\
172:
      };
173:
      static XPoint seg1 512[] = {
174:
           \{-6,9\},\{6,9\},
175:
      };
176:
      static XPoint *char512[] = {
177:
           seg0_512, seg1_512,
178:
           NULL,
179:
      };
180:
      static int char p512[] = {
181:
           XtNumber(seg0_512),XtNumber(seg1_512),
182:
      };
183:
      static XPoint seg0_513[] = {
184:
           {-8,-12},{-8,9},
185:
      };
186:
      static XPoint seg1_513[] = {
187:
           \{-8, -12\}, \{0, 9\}, 
188:
      };
189:
      static XPoint seg2 513[] = {
190:
           \{8, -12\}, \{0, 9\},\
191:
     };
192:
      static XPoint seg3 513[] = {
193:
           \{8, -12\}, \{8, 9\},\
194:
      };
      static XPoint *char513[] = {
195:
196:
           seg0_513, seg1_513, seg2_513, seg3_513,
197:
           NULL,
198:
      };
199:
      static int char_p513[] = {
200:
           XtNumber(seg0_513),XtNumber(seg1_513),XtNumber(seg2_513),
201:
           XtNumber(seg3 513),
202:
      };
203:
      static XPoint seg0 514[] = {
204:
           \{-7, -12\}, \{-7, 9\},\
205:
     };
```

```
206:
                            static XPoint seg1 514[] = {
207:
                                                 \{-7, -12\}, \{7, 9\},\
208:
                            };
209:
                            static XPoint seg2 514[] = {
210:
                                                 \{7, -12\}, \{7, 9\},\
211:
                            };
212:
                            static XPoint *char514[] = {
213:
                                                 seg0_514, seg1_514, seg2_514,
214:
                                                NULL,
215:
                            };
216:
                            static int char p514[] = {
217:
                                                XtNumber(seg0 514), XtNumber(seg1 514), XtNumber(seg2 514),
218:
                            };
219:
                            static XPoint seg0 515[] = {
220:
                                                 \{-2, -12\}, \{-4, -11\}, \{-6, -9\}, \{-7, -7\}, \{-8, -4\}, \{-8, 1\},
221:
                                                 \{-7,4\},\{-6,6\},\{-4,8\},\{-2,9\},\{2,9\},\{4,8\},\{6,6\},\{7,4\},\{8,1\},
222:
                                                 \{8, -4\}, \{7, -7\}, \{6, -9\}, \{4, -11\}, \{2, -12\}, \{-2, -12\}, 
223:
                           };
224:
                            static XPoint *char515[] = {
225:
                                                 seg0_515,
226:
                                                NULL,
227:
                            };
228:
                            static int char_p515[] = {
229:
                                                XtNumber(seg0_515),
230:
                          };
231:
                            static XPoint seg0 516[] = {
232:
                                                \{-7, -12\}, \{-7, 9\},\
233:
                           };
234:
                            static XPoint seg1 516[] = {
235:
                                                 \{-7, -12\}, \{2, -12\}, \{5, -11\}, \{6, -10\}, \{7, -8\}, \{7, -5\}, \{6, -3\}, \{5, -2\}, \{6, -3\}, \{5, -2\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -2\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -2\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -
236:
                                                 \{2, -1\}, \{-7, -1\},\
237:
                           };
238:
                            static XPoint *char516[] = {
239:
                                                 seg0 516, seg1 516,
240:
                                                NULL,
241:
                            };
242:
                            static int char p516[] = {
243:
                                                XtNumber(seg0_516),XtNumber(seg1_516),
244:
                           };
245:
                            static XPoint seg0 517[] = {
                                                 \{-2, -12\}, \{-4, -11\}, \{-6, -9\}, \{-7, -7\}, \{-8, -4\}, \{-8, 1\}, \{-7, 4\},
246:
247:
                                                 \{-6,6\},\{-4,8\},\{-2,9\},\{2,9\},\{4,8\},\{6,6\},\{7,4\},\{8,1\},\{8,-4\},
248:
                                                 \{7, -7\}, \{6, -9\}, \{4, -11\}, \{2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2,
249:
                            };
                            static XPoint seg1_517[] = {
250:
251:
                                                 \{1,5\},\{7,11\},\
252:
                         };
253:
                            static XPoint *char517[] = {
254:
                                                 seg0_517, seg1_517,
255:
                                                 NULL,
256: };
```

```
static int char p517[] = {
257:
258:
                                                 XtNumber(seg0_517),XtNumber(seg1_517),
259:
                             };
260:
                             static XPoint seg0 518[] = {
261:
                                                 \{-7, -12\}, \{-7, 9\},
262:
                            };
                             static XPoint seg1 518[] = {
263:
264:
                                                 \{-7, -12\}, \{2, -12\}, \{5, -11\}, \{6, -10\}, \{7, -8\}, \{7, -6\}, \{6, -4\}, \{5, -3\}, \{6, -4\}, \{5, -3\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -
265:
                                                 \{2, -2\}, \{-7, -2\},\
266:
                             };
267:
                             static XPoint seg2 518[] = {
268:
                                                 \{0, -2\}, \{7, 9\},\
269:
                           };
270:
                             static XPoint *char518[] = {
271:
                                                 seg0_518, seg1_518, seg2_518,
272:
                                                 NULL,
273:
                             };
274:
                             static int char p518[] = {
275:
                                                 XtNumber(seg0_518),XtNumber(seg1_518),XtNumber(seg2_518),
276:
                            };
277:
                             static XPoint seg0_519[] = {
278:
                                                 \{7, -9\}, \{5, -11\}, \{2, -12\}, \{-2, -12\}, \{-5, -11\}, \{-7, -9\},
279:
                                                 \{-7, -7\}, \{-6, -5\}, \{-5, -4\}, \{-3, -3\}, \{3, -1\}, \{5, 0\}, \{6, 1\},
280:
                                                 \{7,3\},\{7,6\},\{5,8\},\{2,9\},\{-2,9\},\{-5,8\},\{-7,6\},
281:
                             };
282:
                             static XPoint *char519[] = {
283:
                                                 seg0 519,
284:
                                                 NULL,
285:
                             };
286:
                             static int char_p519[] = {
287:
                                                 XtNumber(seg0 519),
288:
                             };
289:
                            static XPoint seg0_520[] = {
290:
                                                 \{0, -12\}, \{0, 9\},\
291:
                             };
292:
                             static XPoint seg1 520[] = {
293:
                                                 \{-7, -12\}, \{7, -12\}, 
294:
                             };
295:
                             static XPoint *char520[] = {
296:
                                                 seg0 520, seg1 520,
297:
                                                 NULL,
298:
                             };
299:
                             static int char p520[] = {
300:
                                                 XtNumber(seg0 520), XtNumber(seg1 520),
301:
                            };
302:
                             static XPoint seg0 521[] = {
303:
                                                 \{-7, -12\}, \{-7, 3\}, \{-6, 6\}, \{-4, 8\}, \{-1, 9\}, \{1, 9\}, \{1, 9\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{
304:
                                                 \{4,8\},\{6,6\},\{7,3\},\{7,-12\},
305:
                             };
306:
                             static XPoint *char521[] = {
307:
                                                 seg0_521,
308:
                                                 NULL,
309:
                            };
```

```
310:
      static int char_p521[] = {
311:
           XtNumber(seg0_521),
312:
      };
313:
     static XPoint seg0 522[] = {
314:
           \{-8, -12\}, \{0, 9\},\
315:
      };
316:
      static XPoint seg1_522[] = {
317:
           \{8, -12\}, \{0, 9\},\
318:
      };
319:
      static XPoint *char522[] = {
320:
           seg0_522, seg1_522,
321:
           NULL,
322:
      };
323: static int char_p522[] = {
324:
           XtNumber(seg0 522),XtNumber(seg1 522),
325:
      };
326:
      static XPoint seg0_523[] = {
327:
           \{-10, -12\}, \{-5, 9\},\
328: };
329:
      static XPoint seg1_523[] = {
330:
           \{0, -12\}, \{-5, 9\},\
331: };
332:
      static XPoint seg2_523[] = {
333:
           \{0, -12\}, \{5, 9\},\
334:
      };
335:
      static XPoint seg3_523[] = {
336:
           \{10, -12\}, \{5, 9\},\
337:
      };
338:
      static XPoint *char523[] = {
339:
           seg0 523, seg1 523, seg2 523, seg3 523,
340:
           NULL,
341:
     };
342:
      static int char p523[] = {
343:
           XtNumber(seg0 523),XtNumber(seg1 523),XtNumber(seg2 523),
344:
           XtNumber(seg3_523),
345:
      };
346:
      static XPoint seg0_524[] = {
347:
           \{-7, -12\}, \{7, 9\},\
348:
      };
349:
      static XPoint seg1_524[] = {
350:
           \{7, -12\}, \{-7, 9\},\
351:
      };
352:
      static XPoint *char524[] = {
353:
           seg0_524, seg1_524,
354:
           NULL,
355:
      };
      static int char_p524[] = {
356:
357:
           XtNumber(seg0 524),XtNumber(seg1 524),
358:
      };
359:
      static XPoint seg0_525[] = {
360:
           \{-8, -12\}, \{0, -2\}, \{0, 9\}, 
361: };
```

```
Listing C.4 Continued
```

```
static XPoint seg1 525[] = {
362:
363:
                              \{8, -12\}, \{0, -2\},\
364:
                  };
365:
                 static XPoint *char525[] = {
366:
                              seg0 525, seg1 525,
367:
                              NULL,
368:
                  };
369:
                  static int char p525[] = {
370:
                              XtNumber(seg0_525),XtNumber(seg1_525),
371:
                  };
372:
                  static XPoint seg0_526[] = {
373:
                              \{7, -12\}, \{-7, 9\},\
374:
                 };
375:
                  static XPoint seg1 526[] = {
376:
                              \{-7, -12\}, \{7, -12\}, 
377:
                 };
378:
                 static XPoint seg2 526[] = {
379:
                              \{-7,9\},\{7,9\},
380:
                 };
381:
                  static XPoint *char526[] = {
382:
                              seg0_526, seg1_526, seg2_526,
383:
                              NULL,
384:
                 };
385:
                  static int char p526[] = {
386:
                              XtNumber(seg0_526),XtNumber(seg1_526),XtNumber(seg2_526),
387:
                  };
388:
                  static XPoint seg0 551[] = {
389:
                              \{-11,9\}, \{-9,8\}, \{-6,5\}, \{-3,1\}, \{1,-6\},
390:
                              \{4, -12\}, \{4, 9\}, \{3, 6\}, \{1, 3\}, \{-1, 1\}, \{-4, -1\},
391:
                              \{-6, -1\}, \{-7, 0\}, \{-7, 2\}, \{-6, 4\}, \{-4, 6\}, \{-1, 8\}, \{2, 9\}, \{7, 9\},
392:
                  };
393:
                  static XPoint *char551[] = {
394:
                              seg0 551,
395:
                              NULL,
396:
                  };
397:
                  static int char_p551[] = {
398:
                              XtNumber(seg0 551),
399:
                  };
400:
                  static XPoint seg0 552[] = {
401:
                              \{1, -10\}, \{2, -9\}, \{2, -6\}, \{1, -2\},\
402:
                              \{0,1\},\{-1,3\},\{-3,6\},\{-5,8\},\{-7,9\},\{-8,9\},\{-9,8\},\{-9,5\},\{-8,0\},
403:
                              \{-7, -3\}, \{-6, -5\}, \{-4, -8\}, \{-2, -10\}, \{0, -11\}, \{3, -12\}, \{6, -12\}, \{-7, -3\}, \{-6, -5\}, \{-4, -8\}, \{-2, -10\}, \{-1, -1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}, \{-1, -1\}
404:
                              \{8, -11\}, \{9, -9\}, \{9, -7\}, \{8, -5\}, \{7, -4\}, \{5, -3\}, \{2, -2\},
405:
                  };
406:
                  static XPoint seg1 552[] = {
407:
                              \{1, -2\}, \{2, -2\}, \{5, -1\}, \{6, 0\}, \{7, 2\}, \{7, 5\}, \{6, 7\}, \{5, 8\},
408:
                              \{3,9\},\{0,9\},\{-2,8\},\{-3,6\},
409:
                  };
                  static XPoint *char552[] = {
410:
411:
                              seg0 552, seg1 552,
```

```
412:
                                                                                                                          NULL,
413:
                                                                         };
414:
                                                                         static int char p552[] = {
415:
                                                                                                                            XtNumber(seg0 552),XtNumber(seg1 552),
416:
                                                                         };
417:
                                                                         static XPoint seg0 553[] = {
418:
                                                                                                                            \{2, -6\}, \{2, -5\}, \{3, -4\}, \{5, -4\}, \{7, -5\},\
419:
                                                                                                                            \{8, -7\}, \{8, -9\}, \{7, -11\}, \{5, -12\}, \{2, -12\}, \{-1, -11\}, \{-3, -9\},
420:
                                                                                                                            \{-5, -6\}, \{-6, -4\}, \{-7, 0\}, \{-7, 4\}, \{-6, 7\}, \{-5, 8\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\},
421:
                                                                                                                            \{-1,9\},\{2,8\},\{4,6\},\{5,4\},
422:
                                                                         };
423:
                                                                         static XPoint *char553[] = {
424:
                                                                                                                          seg0 553,
425:
                                                                                                                          NULL,
426:
                                                                         };
427:
                                                                         static int char p553[] = {
428:
                                                                                                                          XtNumber(seg0_553),
429:
                                                                       };
430:
                                                                       static XPoint seg0_554[] = {
431:
                                                                                                                            \{2, -12\}, \{0, -11\}, \{-1, -9\}, \{-2, -5\}, \{-3, 1\}, \{-4, 4\}, \{-5, 6\}, \{-7, 8\}, \{-7, 8\}, \{-1, -2\}, \{-1, -2\}, \{-2, -2\}, \{-3, -2\}, \{-4, -2\}, \{-2, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\},
432:
                                                                                                                            \{-9,9\},\{-11,9\},\{-12,8\},\{-12,6\},\{-11,5\},\{-9,5\},\{-7,6\},\{-5,8\},
433:
                                                                                                                            \{-2,9\},\{1,9\},\{4,8\},\{6,6\},\{8,2\},\{9,-3\},\{9,-7\},\{8,-10\},\{7,-11\},
434:
                                                                                                                            \{5, -12\}, \{2, -12\}, \{0, -10\}, \{0, -8\}, \{1, -5\}, \{3, -2\}, \{5, 0\}, \{8, 2\}, \{5, 0\}, \{8, 2\}, \{1, -5\}, \{2, -12\}, \{2, -12\}, \{2, -10\}, \{2, -10\}, \{2, -10\}, \{2, -10\}, \{2, -10\}, \{3, -2\}, \{3, -2\}, \{5, 0\}, \{3, -2\}, \{5, 0\}, \{3, -2\}, \{5, 0\}, \{3, -2\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{5, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\}, \{6, 0\},
435:
                                                                                                                            \{10,3\},\
436:
                                                                       };
437:
                                                                         static XPoint *char554[] = {
438:
                                                                                                                            seg0 554,
439:
                                                                                                                          NULL,
440:
                                                                         };
441:
                                                                         static int char p554[] = {
442:
                                                                                                                          XtNumber(seg0 554),
443:
                                                                       };
                                                                       static XPoint seg0_555[] = {
444:
445:
                                                                                                                            \{4, -8\}, \{4, -7\}, \{5, -6\}, \{7, -6\}, \{8, -7\}, \{8, -9\}, \{7, -11\}, \{4, -12\},
446:
                                                                                                                            \{0, -12\}, \{-3, -11\}, \{-4, -9\}, \{-4, -6\}, \{-3, -4\}, \{-2, -3\}, \{1, -2\}, \{-2, -3\}, \{-2, -3\}, \{-2, -3\}, \{-2, -3\}, \{-2, -3\}, \{-2, -3\}, \{-2, -3\}, \{-2, -3\}, \{-2, -3\}, \{-2, -3\}, \{-2, -3\}, \{-3, -4\}, \{-2, -3\}, \{-3, -4\}, \{-2, -3\}, \{-3, -4\}, \{-2, -3\}, \{-3, -4\}, \{-2, -3\}, \{-3, -4\}, \{-2, -3\}, \{-3, -4\}, \{-2, -3\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3,
447:
                                                                                                                            \{-2, -2\}, \{-5, -1\}, \{-6, 0\}, \{-7, 2\}, \{-7, 5\}, \{-6, 7\}, \{-5, 8\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\},
448:
                                                                                                                            \{1,9\},\{4,8\},\{6,6\},\{7,4\},
449:
                                                                         };
450:
                                                                           static XPoint *char555[] = {
451:
                                                                                                                            seg0 555,
452:
                                                                                                                          NULL,
453:
                                                                         };
454:
                                                                         static int char p555[] = {
455:
                                                                                                                          XtNumber(seg0 555),
456:
                                                                         };
457:
                                                                         static XPoint seg0 556[] = {
458:
                                                                                                                            \{0, -6\}, \{-2, -6\}, \{-4, -7\}, \{-5, -9\}, \{-4, -11\}, \{-1, -12\}, \{2, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1,
459:
                                                                                                                            \{6, -11\}, \{9, -11\}, \{11, -12\}, 
460:
                                                                       };
461:
                                                                           static XPoint seg1 556[] = {
462:
                                                                                                                            \{6, -11\}, \{4, -4\}, \{2, 2\}, \{0, 6\}, \{-2, 8\}, \{-4, 9\}, \{-6, 9\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-
463:
                                                                                                                            \{-9,6\},\{-9,4\},\{-8,3\},\{-6,3\},\{-4,4\},
464:
                                                                     };
```

```
Listing C.4 Continued
```

```
static XPoint seg2 556[] = {
465:
466:
                                             \{-1, -2\}, \{8, -2\},\
467:
                           };
468:
                           static XPoint *char556[] = {
469:
                                             seg0 556, seg1 556, seg2 556,
470:
                                             NULL,
471:
                           };
472:
                           static int char p556[] = {
473:
                                             XtNumber(seg0 556), XtNumber(seg1 556), XtNumber(seg2 556),
474:
                           };
475:
                           static XPoint seg0_557[] = {
476:
                                             \{-11,9\},\{-9,8\},\{-5,4\},\{-2,-1\},\{-1,-4\},\{0,-8\},
477:
                                             \{0, -11\}, \{-1, -12\}, \{-2, -12\}, \{-3, -11\}, \{-4, -9\}, \{-4, -6\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{
478:
                                             \{-1, -3\}, \{3, -3\}, \{6, -4\}, \{7, -5\}, \{8, -7\}, \{8, -1\}, \{7, 4\}, \{6, 6\},
479:
                                             \{4,8\},\{1,9\},\{-3,9\},\{-6,8\},\{-8,6\},\{-9,4\},\{-9,2\},
480:
                          };
481:
                           static XPoint *char557[] = {
482:
                                             seg0_557,
483:
                                             NULL,
484:
                           };
485:
                           static int char p557[] = {
486:
                                             XtNumber(seg0_557),
487:
                           };
488:
                           static XPoint seg0 558[] = {
489:
                                             \{-5, -5\}, \{-7, -6\}, \{-8, -8\}, \{-8, -9\}, \{-7, -11\},
490:
                                             \{-5, -12\}, \{-4, -12\}, \{-2, -11\}, \{-1, -9\}, \{-1, -7\}, \{-2, -3\}, \{-2, -3\}, \{-2, -3\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{
491:
                                             \{-4,3\},\{-6,7\},\{-8,9\},\{-10,9\},\{-11,8\},\{-11,6\},
492:
                           };
493:
                           static XPoint seg1 558[] = {
494:
                                             \{-5,0\},\{4,-3\},\{6,-4\},\{9,-6\},\{11,-8\},\{12,-10\},\{12,-11\},\{11,-12\},
495:
                                             \{10, -12\}, \{8, -10\}, \{6, -6\}, \{4, 0\}, \{3, 5\}, \{3, 8\}, \{4, 9\}, \{5, 9\}, \{7, 8\},
496:
                                             \{8,7\},\{10,4\},\
497:
                           };
                           static XPoint *char558[] = {
498:
499:
                                             seg0 558, seg1 558,
500:
                                             NULL,
501:
                           };
502:
                           static int char p558[] = {
503:
                                             XtNumber(seg0 558),XtNumber(seg1 558),
504:
                           };
505:
                           static XPoint seg0 559[] = {
506:
                                             \{5,4\},\{3,2\},\{1,-1\},\{0,-3\},\{-1,-6\},\{-1,-9\},\{0,-11\},
507:
                                             \{1, -12\}, \{3, -12\}, \{4, -11\}, \{5, -9\}, \{5, -6\}, \{4, -1\}, \{2, 4\}, \{1, 6\},
508:
                                             \{-1,8\},\{-3,9\},\{-5,9\},\{-7,8\},\{-8,6\},\{-8,4\},\{-7,3\},\{-5,3\},
509:
                                             \{-3,4\},\
510:
                           };
                           static XPoint *char559[] = {
511:
512:
                                             seg0 559,
513:
                                             NULL,
514: };
```

```
515:
                static int char p559[] = {
516:
                           XtNumber(seg0_559),
517:
               };
518:
                static XPoint seg0 560[] = {
519:
                           \{2,12\},\{0,9\},\{-2,4\},\{-3,-2\},\{-3,-8\},\{-2,-11\},\{0,-12\},\{2,-12\},
520:
                           \{3, -11\}, \{4, -8\}, \{4, -5\}, \{3, 0\}, \{0, 9\}, \{-2, 15\}, \{-3, 18\}, \{-4, 20\},
521:
                           \{-6,21\},\{-7,20\},\{-7,18\},\{-6,15\},\{-4,12\},\{-2,10\},\{1,8\},\{5,6\},
522:
               };
523:
                static XPoint *char560[] = {
524:
                           seg0_560,
525:
                           NULL,
526:
                };
527:
                static int char p560[] = {
528:
                           XtNumber(seg0 560),
529:
               };
530:
                static XPoint seg0 561[] = {
531:
                           \{-5, -5\}, \{-7, -6\}, \{-8, -8\}, \{-8, -9\}, \{-7, -11\}, \{-5, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-
532:
                           \{-2, -11\}, \{-1, -9\}, \{-1, -7\}, \{-2, -3\}, \{-4, 3\}, \{-6, 7\}, 
533:
                           \{-8,9\},\{-10,9\},\{-11,8\},\{-11,6\},
534:
                };
535:
                static XPoint seg1 561[] = {
536:
                           \{12, -9\}, \{12, -11\}, \{11, -12\}, \{10, -12\}, \{8, -11\}, \{6, -9\}, \{4, -6\}, 
537:
                           \{2, -4\}, \{0, -3\}, \{-2, -3\}, 
538:
                };
539:
                static XPoint seg2 561[] = {
540:
                           \{0, -3\}, \{1, -1\}, \{1, 6\}, \{2, 8\}, \{3, 9\}, \{4, 9\}, \{6, 8\}, \{7, 7\},
541:
                           {9,4},
542:
                };
543:
                static XPoint *char561[] = {
544:
                           seg0 561, seg1 561, seg2 561,
545:
                           NULL,
546:
               };
547:
                static int char p561[] = {
548:
                           XtNumber(seg0 561), XtNumber(seg1 561), XtNumber(seg2 561),
549:
                };
550:
                static XPoint seg0 562[] = {
551:
                           \{-5,0\},\{-3,0\},\{1,-1\},\{4,-3\},\{6,-5\},\{7,-7\},\{7,-10\},\{6,-12\},
552:
                           \{4, -12\}, \{3, -11\}, \{2, -9\}, \{1, -4\}, \{0, 1\}, \{-1, 4\}, \{-2, 6\}, \{-4, 8\},
553:
                           \{-6,9\},\{-8,9\},\{-9,8\},\{-9,6\},\{-8,5\},\{-6,5\},\{-4,6\},\{-1,8\},
554:
                           \{2,9\},\{4,9\},\{7,8\},\{9,6\},
555:
                };
                static XPoint *char562[] = {
556:
557:
                           seg0_562,
558:
                           NULL,
559:
                };
560:
                static int char p562[] = {
561:
                           XtNumber(seg0_562),
562:
                };
563:
               static XPoint seg0_563[] = {
564:
                           \{-13, -5\}, \{-15, -6\}, \{-16, -8\}, \{-16, -9\}, \{-15, -11\}, \{-13, -12\}, 
565:
                           \{-12, -12\}, \{-10, -11\}, \{-9, -9\}, \{-9, -7\}, \{-10, -2\}, \{-11, 2\}, \{-13, 9\},
566:
               };
```

```
567:
                                                           static XPoint seg1 563[] = {
568:
                                                                                                  \{-11,2\},\{-8,-6\},\{-6,-10\},\{-5,-11\},\{-3,-12\},\{-2,-12\},\{0,-11\},
569:
                                                                                                  \{1, -9\}, \{1, -7\}, \{0, -2\}, \{-1, 2\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{
570:
                                                          };
571:
                                                          static XPoint seg2 563[] = {
572:
                                                                                                  \{-1,2\},\{2,-6\},\{4,-10\},\{5,-11\},\{7,-12\},\{8,-12\},\{10,-11\},\{11,-9\},
573:
                                                                                                  \{11, -7\}, \{10, -2\}, \{8, 5\}, \{8, 8\}, \{9, 9\}, \{10, 9\}, \{12, 8\}, \{13, 7\}, \{15, 4\},
574:
                                                          };
575:
                                                          static XPoint *char563[] = {
576:
                                                                                                  seg0 563, seg1 563, seg2 563,
577:
                                                                                                 NULL,
578:
                                                          };
                                                          static int char p563[] = {
579:
580:
                                                                                                 XtNumber(seg0_563),XtNumber(seg1_563),XtNumber(seg2_563),
581:
                                                          };
                                                        static XPoint seg0_564[] = {
582:
583:
                                                                                                  \{-8, -5\}, \{-10, -6\}, \{-11, -8\}, \{-11, -9\}, \{-10, -11\}, \{-8, -12\}, \{-7, -12\}, \{-7, -12\}, \{-10, -11\}, \{-8, -12\}, \{-7, -12\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-
                                                                                                  \{-5, -11\}, \{-4, -9\}, \{-4, -7\}, \{-5, -2\}, \{-6, 2\}, \{-8, 9\},
584:
585:
                                                          };
586:
                                                          static XPoint seg1 564[] = {
587:
                                                                                                  \{-6,2\},\{-3,-6\},\{-1,-10\},\{0,-11\},\{2,-12\},\{4,-12\},\{6,-11\},\{7,-9\},
588:
                                                                                                  \{7, -7\}, \{6, -2\}, \{4, 5\}, \{4, 8\}, \{5, 9\}, \{6, 9\}, \{8, 8\}, \{9, 7\}, \{11, 4\},
589:
                                                          };
590:
                                                           static XPoint *char564[] = {
591:
                                                                                                  seg0_564, seg1_564,
592:
                                                                                                  NULL,
593:
                                                          };
594:
                                                          static int char p564[] = {
595:
                                                                                                 XtNumber(seg0_564),XtNumber(seg1_564),
596:
                                                          };
597:
                                                           static XPoint seg0 565[] = {
598:
                                                                                                  \{2, -12\}, \{-1, -11\}, \{-3, -9\}, \{-5, -6\}, \{-6, -4\}, \{-7, 0\}, \{-7, 4\}, \{-6, 7\},
599:
                                                                                                  \{-5,8\},\{-3,9\},\{-1,9\},\{2,8\},\{4,6\},\{6,3\},\{7,1\},\{8,-3\},\{8,-7\},
600:
                                                                                                  \{7, -10\}, \{6, -11\}, \{4, -12\}, \{2, -12\}, \{0, -10\}, \{0, -7\}, \{1, -4\}, \{3, -1\}, \{1, -4\}, \{3, -1\}, \{1, -4\}, \{3, -1\}, \{2, -10\}, \{1, -4\}, \{3, -1\}, \{2, -10\}, \{1, -4\}, \{3, -1\}, \{3, -1\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -1
601:
                                                                                                  \{5,1\},\{8,3\},\{10,4\},\
602:
                                                          };
603:
                                                          static XPoint *char565[] = {
604:
                                                                                                  seg0_565,
605:
                                                                                                 NULL,
606:
                                                          };
607:
                                                          static int char_p565[] = {
608:
                                                                                                 XtNumber(seg0 565),
609:
                                                          };
610:
                                                          static XPoint seg0 566[] = {
611:
                                                                                                  \{1, -10\}, \{2, -9\}, \{2, -6\}, \{1, -2\}, \{0, 1\}, \{-1, 3\}, \{-3, 6\}, \{-5, 8\}, \{-5, 8\}, \{-1, 3\}, \{-3, 6\}, \{-3, 6\}, \{-5, 8\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{
612:
                                                                                                  \{-7,9\},\{-8,9\},\{-9,8\},\{-9,5\},\{-8,0\},\{-7,-3\},\{-6,-5\},\{-4,-8\},
613:
                                                                                                  \{-2, -10\}, \{0, -11\}, \{3, -12\}, \{8, -12\}, \{10, -11\}, \{11, -10\}, \{12, -8\}, \{10, -11\}, \{11, -10\}, \{12, -8\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12, -10\}, \{12
614:
                                                                                                  \{12, -5\}, \{11, -3\}, \{10, -2\}, \{8, -1\}, \{5, -1\}, \{3, -2\}, \{2, -3\}, \{2, -3\}, \{3, -2\}, \{2, -3\}, \{3, -2\}, \{2, -3\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}
615:
                                                        };
616:
                                                          static XPoint *char566[] = {
617:
                                                                                                  seg0 566,
```

```
618:
                                                                                                                      NULL,
619:
                                                                       };
620:
                                                                       static int char p566[] = {
621:
                                                                                                                        XtNumber(seg0 566),
622:
                                                                       };
623:
                                                                       static XPoint seg0 567[] = {
624:
                                                                                                                        \{3, -6\}, \{2, -4\}, \{1, -3\}, \{-1, -2\}, \{-3, -2\}, \{-4, -4\}, \{-4, -6\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9
625:
                                                                                                                        \{-1, -11\}, \{2, -12\}, \{5, -12\}, \{7, -11\}, \{8, -9\}, \{8, -5\},
626:
                                                                                                                        \{7, -2\}, \{5, 1\}, \{1, 5\}, \{-2, 7\}, \{-4, 8\}, \{-7, 9\}, \{-9, 9\}, \{-10, 8\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-10, 6\}, \{-1
627:
                                                                                                                        \{-9,5\},\{-7,5\},\{-5,6\},\{-2,8\},\{1,9\},\{4,9\},\{7,8\},\{9,6\},
628:
                                                                       };
629:
                                                                       static XPoint *char567[] = {
630:
                                                                                                                        seg0 567,
631:
                                                                                                                      NULL,
632:
                                                                       };
633:
                                                                       static int char p567[] = {
634:
                                                                                                                      XtNumber(seg0_567),
635:
                                                                     };
636:
                                                                     static XPoint seg0_568[] = {
637:
                                                                                                                        \{1, -10\}, \{2, -9\}, \{2, -6\}, \{1, -2\}, \{0, 1\}, \{-1, 3\}, \{-3, 6\}, \{-5, 8\},
638:
                                                                                                                        \{-7,9\},\{-8,9\},\{-9,8\},\{-9,5\},\{-8,0\},\{-7,-3\},\{-6,-5\},\{-4,-8\},
639:
                                                                                                                        \{-2, -10\}, \{0, -11\}, \{3, -12\}, \{7, -12\}, \{9, -11\}, \{10, -10\}, \{11, -8\},
640:
                                                                                                                        \{11, -5\}, \{10, -3\}, \{9, -2\}, \{7, -1\}, \{4, -1\}, \{1, -2\}, \{2, -1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}, \{3,
641:
                                                                                                                        \{3,6\},\{4,8\},\{6,9\},\{8,8\},\{9,7\},\{11,4\},
642:
                                                                       };
643:
                                                                       static XPoint *char568[] = {
644:
                                                                                                                        seg0 568,
645:
                                                                                                                      NULL,
646:
                                                                       };
647:
                                                                       static int char p568[] = {
648:
                                                                                                                      XtNumber(seg0 568),
649:
                                                                       };
                                                                     static XPoint seg0_569[] = {
650:
651:
                                                                                                                        \{-10,9\}, \{-8,8\}, \{-6,6\}, \{-3,2\}, \{-1,-1\}, \{1,-5\}, \{2,-8\}, \{2,-11\}, \{1,-5\}, \{2,-8\}, \{2,-11\}, \{2,-11\}, \{2,-11\}, \{2,-11\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\}, \{3,-12\},
652:
                                                                                                                        \{1, -12\}, \{0, -12\}, \{-1, -11\}, \{-2, -9\}, \{-2, -7\}, \{-1, -5\}, \{1, -3\}, \{4, -1\}, \{-1, -5\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, -3\}, \{-1, 
653:
                                                                                                                        \{6,1\},\{7,3\},\{7,5\},\{6,7\},\{5,8\},\{2,9\},\{-2,9\},\{-5,8\},\{-7,6\},
654:
                                                                                                                        \{-8,4\},\{-8,2\},
655:
                                                                       };
656:
                                                                         static XPoint *char569[] = {
657:
                                                                                                                        seg0 569,
658:
                                                                                                                        NULL,
659:
                                                                       };
660:
                                                                       static int char p569[] = {
661:
                                                                                                                      XtNumber(seg0 569),
662:
                                                                       };
663:
                                                                       static XPoint seg0 570[] = {
664:
                                                                                                                        \{0, -6\}, \{-2, -6\}, \{-4, -7\}, \{-5, -9\}, \{-4, -11\}, \{-1, -12\}, \{2, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1,
665:
                                                                                                                        \{6, -11\}, \{9, -11\}, \{11, -12\}, 
666:
                                                                       };
667:
                                                                         static XPoint seg1 570[] = {
668:
                                                                                                                        \{6, -11\}, \{4, -4\}, \{2, 2\}, \{0, 6\}, \{-2, 8\}, \{-4, 9\}, \{-6, 9\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-8, 8\}, \{-
669:
                                                                                                                        \{-9,6\},\{-9,4\},\{-8,3\},\{-6,3\},\{-4,4\},
670:
                                                                     };
```

```
Listing C.4 Continued
```

```
671:
                                                         static XPoint *char570[] = {
672:
                                                                                                seg0_570, seg1_570,
673:
                                                                                               NULL,
674:
                                                         };
675:
                                                         static int char_p570[] = {
676:
                                                                                               XtNumber(seg0 570),XtNumber(seg1 570),
677:
                                                         };
678:
                                                         static XPoint seg0 571[] = {
679:
                                                                                                \{-8, -5\}, \{-10, -6\}, \{-11, -8\}, \{-11, -9\}, \{-10, -11\}, \{-8, -12\}, \{-7, -12\}, \{-7, -12\}, \{-10, -11\}, \{-8, -12\}, \{-7, -12\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-10, -11\}, \{-
680:
                                                                                                \{-5, -11\}, \{-4, -9\}, \{-4, -7\}, \{-5, -3\}, \{-6, 0\}, \{-7, 4\},
681:
                                                                                                \{-7,6\}, \{-6,8\}, \{-4,9\}, \{-2,9\}, \{0,8\}, \{1,7\}, \{3,3\}, \{6,-5\}, \{8,-12\}, \{0,8\}, \{1,7\}, \{3,3\}, \{1,7\}, \{2,1,2\}, \{1,7\}, \{2,1,2\}, \{2,1,2\}, \{2,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1,2\}, \{3,1
682:
                                                       };
683:
                                                         static XPoint seg1 571[] = {
684:
                                                                                                \{6, -5\}, \{5, -1\}, \{4, 5\}, \{4, 8\}, \{5, 9\}, \{6, 9\}, \{8, 8\}, \{9, 7\},
685:
                                                                                                {11,4},
686:
                                                       };
687:
                                                         static XPoint *char571[] = {
688:
                                                                                                seg0 571, seg1 571,
689:
                                                                                               NULL,
690:
                                                         };
691:
                                                         static int char p571[] = {
692:
                                                                                               XtNumber(seg0 571),XtNumber(seg1 571),
693:
                                                       };
694:
                                                         static XPoint seg0 572[] = {
695:
                                                                                                \{-7, -5\}, \{-9, -6\}, \{-10, -8\}, \{-10, -9\}, \{-9, -11\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, 
696:
                                                                                                \{-6, -12\}, \{-4, -11\}, \{-3, -9\}, \{-3, -7\}, \{-4, -3\}, \{-5, 0\}, \{-6, 4\},
697:
                                                                                                \{-6,7\},\{-5,9\},\{-3,9\},\{-1,8\},\{2,5\},\{4,2\},\{6,-2\},\{7,-5\},
698:
                                                                                                \{8, -9\}, \{8, -11\}, \{7, -12\}, \{6, -12\}, \{5, -11\}, \{4, -9\}, \{4, -7\}, \{5, -4\},
699:
                                                                                                \{7, -2\}, \{9, -1\},\
700:
                                                       };
701:
                                                         static XPoint *char572[] = {
702:
                                                                                                seg0_572,
703:
                                                                                               NULL,
704:
                                                         };
705:
                                                         static int char_p572[] = {
706:
                                                                                               XtNumber(seg0 572),
707:
                                                     };
708:
                                                         static XPoint seg0 573[] = {
709:
                                                                                                \{-10, -5\}, \{-12, -6\}, \{-13, -8\}, \{-13, -9\}, \{-12, -11\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -1
710:
                                                                                                \{-9, -12\}, \{-7, -11\}, \{-6, -9\}, \{-6, -6\}, \{-7, 9\}, \{-9, -12\}, \{-7, -11\}, \{-6, -9\}, \{-7, -11\}, \{-9, -12\}, \{-7, -11\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1
711:
                                                         };
712:
                                                         static XPoint seg1 573[] = {
713:
                                                                                                \{3, -12\}, \{-7, 9\},\
714: };
715:
                                                         static XPoint seg2 573[] = {
716:
                                                                                                \{3, -12\}, \{1, 9\},\
717:
                                                         };
718:
                                                         static XPoint seg3 573[] = {
719:
                                                                                                \{15, -12\}, \{13, -11\}, \{10, -8\}, \{7, -4\}, \{4, 2\}, \{1, 9\},\
720: };
```

```
721:
                                                                 static XPoint *char573[] = {
722:
                                                                                                               seg0_573, seg1_573, seg2_573, seg3_573,
723:
                                                                                                             NULL,
724:
                                                                };
725:
                                                                 static int char_p573[] = {
726:
                                                                                                               XtNumber(seg0 573), XtNumber(seg1 573), XtNumber(seg2 573),
727:
                                                                                                             XtNumber(seg3 573),
728:
                                                                 };
729:
                                                                 static XPoint seg0 574[] = {
730:
                                                                                                               \{-4, -6\}, \{-6, -6\}, \{-7, -7\}, \{-7, -9\}, \{-6, -11\}, \{-4, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-
731:
                                                                                                               \{0, -11\}, \{1, -9\}, \{1, -6\}, \{-1, 3\}, \{-1, 6\}, \{0, 8\}, \{2, 9\}, \{4, 9\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}
732:
                                                                                                               \{7,6\},\{7,4\},\{6,3\},\{4,3\},
733:
                                                                 };
734:
                                                                 static XPoint seg1 574[] = {
735:
                                                                                                               \{11, -9\}, \{11, -11\}, \{10, -12\}, \{8, -12\}, \{6, -11\}, \{4, -9\}, \{2, -6\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2, 3\}, \{-2
736:
                                                                                                               \{-4,6\}, \{-6,8\}, \{-8,9\}, \{-10,9\}, \{-11,8\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11,6\}, \{-11
737:
                                                                 };
738:
                                                                 static XPoint *char574[] = {
739:
                                                                                                               seg0_574, seg1_574,
740:
                                                                                                             NULL,
741:
                                                                 };
742:
                                                                 static int char p574[] = {
743:
                                                                                                             XtNumber(seg0_574),XtNumber(seg1_574),
744:
                                                                };
745:
                                                                 static XPoint seg0 575[] = {
746:
                                                                                                               \{-7, -5\}, \{-9, -6\}, \{-10, -8\}, \{-10, -9\}, \{-9, -11\}, \{-7, -12\}, \{-6, -12\}, \{-6, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-10, -12\}, \{-
747:
                                                                                                               \{-4, -11\}, \{-3, -9\}, \{-3, -7\}, \{-4, -3\}, \{-5, 0\}, \{-6, 4\}, \{-6, 6\}, \{-5, 8\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 
748:
                                                                                                               \{-4,9\},\{-2,9\},\{0,8\},\{2,6\},\{4,3\},\{5,1\},\{7,-5\},
749:
                                                                 };
750:
                                                                 static XPoint seg1 575[] = {
751:
                                                                                                               \{9, -12\}, \{7, -5\}, \{4, 5\}, \{2, 11\}, \{0, 16\}, \{-2, 20\}, \{-4, 21\}, \{-5, 20\}, 
752:
                                                                                                               \{-5,18\},\{-4,15\},\{-2,12\},\{1,9\},\{4,7\},\{9,4\},
753:
                                                                 };
754:
                                                                 static XPoint *char575[] = {
755:
                                                                                                               seg0_575, seg1_575,
756:
                                                                                                             NULL,
757:
                                                                 };
758:
                                                                 static int char p575[] = {
759:
                                                                                                             XtNumber(seg0 575),XtNumber(seg1 575),
760:
                                                                };
761:
                                                                 static XPoint seg0 576[] = {
762:
                                                                                                               \{3, -6\}, \{2, -4\}, \{1, -3\}, \{-1, -2\}, \{-3, -2\}, \{-4, -4\}, \{-4, -6\},
763:
                                                                                                               \{-3, -9\}, \{-1, -11\}, \{2, -12\}, \{5, -12\}, \{7, -11\}, \{8, -9\}, \{8, -5\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, -2\}, \{7, 
764:
                                                                                                               \{5,2\},\{2,5\},\{-2,8\},\{-4,9\},\{-7,9\},\{-8,8\},\{-8,6\},\{-7,5\},\{-4,5\},
765:
                                                                                                               \{-2,6\},\{-1,7\},\{0,9\},\{0,12\},\{-1,15\},\{-2,17\},\{-4,20\},\{-6,21\},
766:
                                                                                                               \{-7, 20\}, \{-7, 18\}, \{-6, 15\}, \{-4, 12\}, \{-1, 9\}, \{2, 7\}, \{8, 4\},
767:
                                                                 };
768:
                                                                 static XPoint *char576[] = {
769:
                                                                                                               seg0_576,
770:
                                                                                                             NULL,
771:
                                                                 };
772:
                                                                 static int char p576[] = {
773:
                                                                                                             XtNumber(seg0_576),
774: };
```

```
775:
                  static XPoint seg0 601[] = {
776:
                               \{6, -5\}, \{6, 9\},\
777:
                  };
778:
               static XPoint seg1_601[] = {
                               \{6, -2\}, \{4, -4\}, \{2, -5\}, \{-1, -5\}, \{-3, -4\}, \{-5, -2\}, \{-6, 1\}, \{-6, 3\},
779:
780:
                               \{-5,6\},\{-3,8\},\{-1,9\},\{2,9\},\{4,8\},\{6,6\},
781:
                 };
782:
                  static XPoint *char601[] = {
783:
                               seg0_601, seg1_601,
784:
                               NULL,
785:
                 };
786:
                  static int char p601[] = {
                               XtNumber(seg0_601),XtNumber(seg1_601),
787:
788:
                  };
789:
                 static XPoint seg0 602[] = {
790:
                               \{-6, -12\}, \{-6, 9\},\
791:
                  };
792:
                 static XPoint seg1 602[] = {
793:
                               \{-6, -2\}, \{-4, -4\}, \{-2, -5\}, \{1, -5\}, \{3, -4\}, \{5, -2\}, \{6, 1\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6,
794:
                               \{5,6\},\{3,8\},\{1,9\},\{-2,9\},\{-4,8\},\{-6,6\},
795:
                  };
                 static XPoint *char602[] = {
796:
797:
                               seg0_602, seg1_602,
798:
                               NULL,
799:
                 };
800:
                  static int char p602[] = {
801:
                               XtNumber(seg0_602),XtNumber(seg1_602),
802:
                  };
803:
                  static XPoint seg0_603[] = {
                               \{6, -2\}, \{4, -4\}, \{2, -5\}, \{-1, -5\}, \{-3, -4\}, \{-5, -2\}, \{-6, 1\},\
804:
805:
                               \{-6,3\},\{-5,6\},\{-3,8\},\{-1,9\},\{2,9\},\{4,8\},\{6,6\},
806:
                 };
807:
                  static XPoint *char603[] = {
808:
                               seg0 603,
809:
                               NULL,
810:
                  };
                  static int char p603[] = {
811:
812:
                               XtNumber(seg0_603),
813:
                 };
814:
                  static XPoint seg0 604[] = {
815:
                               \{6, -12\}, \{6, 9\},\
816:
                  };
817:
                 static XPoint seg1 604[] = {
                               \{6, -2\}, \{4, -4\}, \{2, -5\}, \{-1, -5\}, \{-3, -4\}, \{-5, -2\}, \{-6, 1\}, \{-6, 3\},
818:
819:
                               \{-5,6\},\{-3,8\},\{-1,9\},\{2,9\},\{4,8\},\{6,6\},
820:
                  };
                  static XPoint *char604[] = {
821:
822:
                               seg0_604, seg1_604,
823:
                               NULL,
824: };
```

```
825:
                                        static int char p604[] = {
826:
                                                                   XtNumber(seg0_604),XtNumber(seg1_604),
827:
                                       };
828:
                                        static XPoint seg0 605[] = {
829:
                                                                    \{-6,1\},\{6,1\},\{6,-1\},\{5,-3\},\{4,-4\},\{2,-5\},\{-1,-5\},\{-3,-4\},
830:
                                                                    \{-5, -2\}, \{-6, 1\}, \{-6, 3\}, \{-5, 6\}, \{-3, 8\}, \{-1, 9\}, \{2, 9\}, \{4, 8\}, \{6, 6\},
831:
                                        };
832:
                                        static XPoint *char605[] = {
833:
                                                                    seg0 605,
834:
                                                                   NULL,
835:
                                        };
836:
                                        static int char p605[] = {
837:
                                                                   XtNumber(seg0 605),
838:
                                       };
839:
                                        static XPoint seg0 606[] = {
840:
                                                                    \{5, -12\}, \{3, -12\}, \{1, -11\}, \{0, -8\},\
841:
                                                                    {0,9},
842:
                                      };
843:
                                        static XPoint seg1_606[] = {
844:
                                                                    \{-3, -5\}, \{4, -5\}, 
845:
                                       };
846:
                                        static XPoint *char606[] = {
847:
                                                                    seg0_606, seg1_606,
848:
                                                                   NULL,
849:
                                       };
850:
                                        static int char_p606[] = {
851:
                                                                   XtNumber(seg0 606),XtNumber(seg1 606),
852:
                                        };
853:
                                        static XPoint seg0 607[] = {
854:
                                                                    \{6, -5\}, \{6, 11\}, \{5, 14\}, \{4, 15\}, \{2, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\}, \{-1, 16\},
855:
                                                                    \{-3, 15\},\
856:
                                        };
857:
                                        static XPoint seg1 607[] = {
858:
                                                                    \{6, -2\}, \{4, -4\}, \{2, -5\}, \{-1, -5\}, \{-3, -4\}, \{-5, -2\}, \{-6, 1\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}, \{-6, 3\}
859:
                                                                    \{-5,6\},\{-3,8\},\{-1,9\},\{2,9\},\{4,8\},\{6,6\},
860:
                                        };
861:
                                        static XPoint *char607[] = {
862:
                                                                    seg0_607, seg1_607,
863:
                                                                   NULL,
864:
                                        };
865:
                                        static int char p607[] = {
866:
                                                                   XtNumber(seg0 607),XtNumber(seg1 607),
867:
                                        };
868:
                                        static XPoint seg0 608[] = {
869:
                                                                    \{-5, -12\}, \{-5, 9\},\
870:
                                       };
871:
                                        static XPoint seg1_608[] = {
872:
                                                                    \{-5, -1\}, \{-2, -4\}, \{0, -5\}, \{3, -5\}, \{5, -4\}, \{6, -1\}, \{6, 9\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, \{6, -1\}, 
873:
                                       };
874:
                                        static XPoint *char608[] = {
875:
                                                                    seg0_608, seg1_608,
876:
                                                                    NULL,
877: };
```

```
Listing C.4 Continued
```

```
878:
                   static int char p608[] = {
879:
                                 XtNumber(seg0_608),XtNumber(seg1_608),
880:
                   };
881:
                   static XPoint seg0 609[] = {
882:
                                 {-1,-12},
883:
                                 \{0, -11\}, \{1, -12\}, \{0, -13\}, \{-1, -12\},\
884:
                   };
885:
                   static XPoint seg1_609[] = {
886:
                                 \{0, -5\}, \{0, 9\},\
887:
                   };
888:
                   static XPoint *char609[] = {
889:
                                 seg0 609, seg1 609,
890:
                                 NULL,
891:
                   };
892:
                   static int char_p609[] = {
893:
                                 XtNumber(seg0 609),XtNumber(seg1 609),
894:
                   };
895:
                   static XPoint seg0 610[] = {
896:
                                 \{0, -12\}, \{1, -11\}, \{2, -12\}, \{1, -13\}, \{0, -12\}, 
897:
                   };
898:
                   static XPoint seg1_610[] = {
899:
                                 \{1, -5\}, \{1, 12\}, \{0, 15\}, \{-2, 16\}, \{-4, 16\},
900:
                  };
901:
                   static XPoint *char610[] = {
902:
                                 seg0 610, seg1 610,
903:
                                 NULL,
904:
                   };
905:
                   static int char p610[] = {
906:
                                 XtNumber(seg0_610),XtNumber(seg1_610),
907:
                  };
908:
                   static XPoint seg0 611[] = {
909:
                                 \{-5, -12\}, \{-5, 9\},
910:
                   };
911:
                   static XPoint seg1 611[] = {
912:
                                 \{5, -5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{-5, 5\}, \{
913:
                   };
914:
                   static XPoint seg2 611[] = {
915:
                                 \{-1,1\},\{6,9\},
916:
                   };
                   static XPoint *char611[] = {
917:
918:
                                 seg0_611, seg1_611, seg2_611,
919:
                                 NULL,
920:
                   };
921:
                   static int char p611[] = {
922:
                                 XtNumber(seg0_611),XtNumber(seg1_611),XtNumber(seg2_611),
923:
                   };
924:
                   static XPoint seg0_612[] = {
925:
                                 \{0, -12\}, \{0, 9\},\
926: };
```

```
927: static XPoint *char612[] = {
928:
                               seg0_612,
929:
                              NULL,
930:
                 };
931:
                 static int char p612[] = {
932:
                              XtNumber(seg0_612),
933:
                 };
934:
                 static XPoint seg0 613[] = {
935:
                               \{-11, -5\}, \{-11, 9\},\
936:
                  };
937:
                 static XPoint seg1 613[] = {
938:
                               \{-11, -1\}, \{-8, -4\}, \{-6, -5\}, \{-3, -5\}, \{-1, -4\}, \{0, -1\}, \{0, 9\}, \{-1, -4\}, \{0, -1\}, \{0, 9\}, \{-1, -4\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, 
939: };
940:
                 static XPoint seg2 613[] = {
941:
                               \{0, -1\}, \{3, -4\}, \{5, -5\}, \{8, -5\}, \{10, -4\}, \{11, -1\}, \{11, 9\},
942: };
943:
                 static XPoint *char613[] = {
944:
                               seg0_613, seg1_613, seg2_613,
945:
                              NULL,
946:
                 };
947:
                  static int char p613[] = {
948:
                              XtNumber(seg0 613), XtNumber(seg1 613), XtNumber(seg2 613),
949:
               };
950:
                 static XPoint seg0 614[] = {
951:
                              {-5,-5},
952:
                               \{-5,9\},\
953: };
954:
                  static XPoint seg1 614[] = {
955:
                               \{-5, -1\}, \{-2, -4\}, \{0, -5\}, \{3, -5\}, \{5, -4\}, \{6, -1\}, \{6, 9\},
956:
               };
957:
                 static XPoint *char614[] = {
958:
                              seg0_614, seg1_614,
959:
                              NULL,
960:
                 };
961:
                  static int char p614[] = {
962:
                              XtNumber(seg0_614),XtNumber(seg1_614),
963:
               };
964:
                 static XPoint seg0 615[] = {
965:
                               \{-1, -5\}, \{-3, -4\}, \{-5, -2\}, \{-6, 1\}, \{-6, 3\}, \{-5, 6\}, \{-3, 8\},
966:
                               \{-1,9\},\{2,9\},\{4,8\},\{6,6\},\{7,3\},\{7,1\},\{6,-2\},\{4,-4\},
967:
                               \{2, -5\}, \{-1, -5\},\
968:
                 };
969:
                 static XPoint *char615[] = {
970:
                               seg0_615,
971:
                              NULL,
972:
                 };
973:
                 static int char p615[] = {
974:
                              XtNumber(seg0_615),
975:
                 };
976: static XPoint seg0_616[] = {
977:
                               \{-6, -5\}, \{-6, 16\},
978: };
```

```
979:
                 static XPoint seg1_616[] = {
980:
                              \{-6, -2\}, \{-4, -4\}, \{-2, -5\}, \{1, -5\}, \{3, -4\}, \{5, -2\}, \{6, 1\}, \{6, 3\},
981:
                              \{5,6\},\{3,8\},\{1,9\},\{-2,9\},\{-4,8\},\{-6,6\},
982:
                 };
983:
                  static XPoint *char616[] = {
984:
                              seg0_616, seg1_616,
985:
                              NULL,
986:
                 };
987:
                  static int char p616[] = {
988:
                              XtNumber(seg0 616),XtNumber(seg1 616),
989:
                  };
990:
                 static XPoint seg0 617[] = {
991:
                              \{6, -5\}, \{6, 16\},\
992:
                  };
993:
                 static XPoint seg1_617[] = {
994:
                              \{6, -2\}, \{4, -4\}, \{2, -5\}, \{-1, -5\}, \{-3, -4\}, \{-5, -2\}, \{-6, 1\}, \{-6, 3\},
995:
                              \{-5,6\},\{-3,8\},\{-1,9\},\{2,9\},\{4,8\},\{6,6\},
996:
                 };
997:
                 static XPoint *char617[] = {
998:
                              seg0_617, seg1_617,
999:
                              NULL,
1000: };
1001: static int char p617[] = {
                             XtNumber(seg0_617),XtNumber(seg1_617),
1002:
1003: };
1004: static XPoint seg0 618[] = {
1005:
                              \{-3, -5\}, \{-3, 9\},
1006: };
1007: static XPoint seg1 618[] = {
1008:
                              \{-3,1\},\{-2,-2\},\{0,-4\},\{2,-5\},\{5,-5\},
1009: };
1010: static XPoint *char618[] = {
1011:
                              seg0_618, seg1_618,
1012:
                              NULL,
1013: \};
1014: static int char p618[] = {
1015:
                              XtNumber(seg0 618),XtNumber(seg1 618),
1016: };
1017: static XPoint seg0 619[] = {
1018:
                              \{6, -2\}, \{5, -4\}, \{2, -5\}, \{-1, -5\}, \{-4, -4\}, \{-5, -2\}, \{-4, 0\}, \{-2, 1\},
1019:
                              \{3,2\},\{5,3\},\{6,5\},\{6,6\},\{5,8\},\{2,9\},\{-1,9\},\{-4,8\},\{-5,6\},
1020: \};
1021: static XPoint *char619[] = {
1022:
                              seg0 619,
1023:
                              NULL,
1024: };
1025: static int char_p619[] = {
1026:
                             XtNumber(seg0 619),
1027: };
1028: static XPoint seg0 620[] = {
1029:
                              \{0, -12\}, \{0, 5\}, \{1, 8\}, \{3, 9\}, \{0, 5\}, \{1, 8\}, \{3, 9\}, \{1, 8\}, \{3, 9\}, \{1, 8\}, \{3, 9\}, \{1, 8\}, \{2, 9\}, \{1, 8\}, \{3, 9\}, \{1, 8\}, \{2, 9\}, \{1, 8\}, \{2, 9\}, \{2, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3
```

```
1030:
                                {5,9},
1031: \};
1032: static XPoint seg1 620[] = {
1033:
                                \{-3, -5\}, \{4, -5\}, 
1034: };
1035: static XPoint *char620[] = {
1036:
                                seg0 620, seg1 620,
1037:
                                NULL,
1038: };
1039: static int char p620[] = {
1040:
                                XtNumber(seg0_620),XtNumber(seg1_620),
1041: };
1042: static XPoint seg0 621[] = {
                                \{-5, -5\}, \{-5, 5\}, \{-4, 8\}, \{-2, 9\}, \{1, 9\}, \{3, 8\},
1043:
1044:
                                {6,5},
1045: };
1046: static XPoint seg1_621[] = {
1047:
                               \{6, -5\}, \{6, 9\},\
1048: };
1049: static XPoint *char621[] = {
1050:
                                seg0 621, seg1 621,
1051:
                                NULL,
1052: \};
1053: static int char p621[] = {
1054:
                               XtNumber(seg0 621),XtNumber(seg1 621),
1055: };
1056: static XPoint seg0 622[] = {
1057:
                               \{-6, -5\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0
1058: };
1059: static XPoint seg1 622[] = {
1060:
                               \{6, -5\}, \{0, 9\},\
1061: };
1062: static XPoint *char622[] = {
1063:
                                seg0 622, seg1 622,
1064:
                               NULL,
1065: };
1066: static int char p622[] = {
                               XtNumber(seg0_622),XtNumber(seg1_622),
1067:
1068: };
1069: static XPoint seg0 623[] = {
1070:
                               \{-8, -5\}, \{-4, 9\},
1071: };
1072: static XPoint seg1 623[] = {
1073:
                                \{0, -5\}, \{-4, 9\},\
1074: };
1075: static XPoint seg2_623[] = {
1076:
                                \{0, -5\}, \{4, 9\},\
1077: };
1078: static XPoint seg3_623[] = {
1079:
                                \{8, -5\}, \{4, 9\},\
1080: };
1081: static XPoint *char623[] = {
1082:
                                seg0_623, seg1_623, seg2_623, seg3_623,
```

```
1083:
                                                   NULL,
1084: };
1085: static int char p623[] = {
1086:
                                                  XtNumber(seg0 623),XtNumber(seg1 623),XtNumber(seg2 623),
1087:
                                                   XtNumber(seg3 623),
1088: };
1089: static XPoint seg0_624[] = {
1090:
                                                   \{-5, -5\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6
1091: };
1092: static XPoint seg1 624[] = {
1093:
                                                 {6,-5},{-5,9},
1094: };
1095: static XPoint *char624[] = {
1096:
                                                   seg0_624, seg1_624,
1097:
                                                  NULL,
1098: };
1099: static int char p624[] = {
1100:
                                                 XtNumber(seg0 624), XtNumber(seg1 624),
1101: };
1102: static XPoint seg0 625[] = {
1103:
                                                   \{-6, -5\}, \{0, 9\},\
1104: };
1105: static XPoint seg1_625[] = {
1106:
                                                   \{6, -5\}, \{0, 9\}, \{-2, 13\}, \{-4, 15\}, \{-6, 16\}, \{-7, 16\},
1107: };
1108: static XPoint *char625[] = {
1109:
                                                   seg0_625, seg1_625,
1110:
                                                  NULL,
1111: };
1112: static int char p625[] = {
1113:
                                                  XtNumber(seg0 625),XtNumber(seg1 625),
1114: \};
1115: static XPoint seg0 626[] = {
1116:
                                                   \{6, -5\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{
1117: \};
1118: static XPoint seg1_626[] = {
1119:
                                                 \{-5, -5\}, \{6, -5\},\
1120: };
1121: static XPoint seg2 626[] = {
1122:
                                                   \{-5,9\},\{6,9\},
1123: };
1124: static XPoint *char626[] = {
1125:
                                                   seg0 626, seg1 626, seg2 626,
1126:
                                                   NULL,
1127: \};
1128: static int char p626[] = {
1129:
                                                  XtNumber(seg0 626), XtNumber(seg1 626), XtNumber(seg2 626),
1130: };
1131: static XPoint seg0 651[] = {
1132:
                                                   \{3,3\},\{2,1\},\{0,0\},\{-2,0\},\{-4,1\},\{-5,2\},\{-6,4\},\{-6,6\},\{-5,8\},
```

```
1133:
                            \{-3,9\},\{-1,9\},\{1,8\},\{2,6\},\{4,0\},\{3,5\},
1134:
                            \{3,8\},\{4,9\},\{5,9\},\{7,8\},\{8,7\},\{10,4\},
1135: \};
1136: static XPoint *char651[] = {
1137:
                            seg0 651,
1138:
                            NULL,
1139: };
1140: static int char p651[] = \{
1141:
                            XtNumber(seg0 651),
1142: \};
1143: static XPoint seg0_652[] = {
1144:
                            \{-5,4\}, \{-3,1\}, \{0,-4\}, \{1,-6\}, \{2,-9\}, \{2,-11\}, \{1,-12\}, \{-1,-11\}, \{1,-12\}, \{-1,-11\}, \{-1,-12\}, \{-1,-11\}, \{-1,-12\}, \{-1,-11\}, \{-1,-12\}, \{-1,-11\}, \{-1,-12\}, \{-1,-11\}, \{-1,-12\}, \{-1,-11\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12\}, \{-1,-12
1145:
                            \{-2, -9\}, \{-3, -5\}, \{-4, 2\}, \{-4, 8\}, \{-3, 9\}, \{-2, 9\}, \{0, 8\}, \{2, 6\},
1146:
                            \{3,3\},\{3,0\},\{4,4\},\{5,5\},\{7,5\},\{9,4\},
1147: \};
1148: static XPoint *char652[] = {
1149:
                            seg0 652,
1150:
                            NULL,
1151: };
1152: static int char p652[] = {
1153:
                            XtNumber(seg0 652),
1154: };
1155: static XPoint seg0 653[] = {
                            \{2,2\},\{2,1\},\{1,0\},\{-1,0\},\{-3,1\},\{-4,2\},\{-5,4\},\{-5,6\},
1156:
1157:
                            \{-4,8\},\{-2,9\},\{1,9\},\{4,7\},\{6,4\},
1158: };
1159: static XPoint *char653[] = {
1160:
                            seg0 653,
1161:
                            NULL,
1162: };
1163: static int char p653[] = {
1164:
                           XtNumber(seg0 653),
1165: };
1166: static XPoint seg0_654[] = {
1167:
                            \{3,3\},\{2,1\},\{0,0\},\{-2,0\},
                            \{-4,1\},\{-5,2\},\{-6,4\},\{-6,6\},\{-5,8\},\{-3,9\},\{-1,9\},\{1,8\},\{2,6\},
1168:
1169:
                            {8, -12},
1170: };
1171: static XPoint seg1 654[] = {
1172:
                            \{4,0\},\{3,5\},\{3,8\},\{4,9\},\{5,9\},\{7,8\},\{8,7\},\{10,4\},
1173: };
1174: static XPoint *char654[] = {
1175:
                            seg0_654, seg1_654,
1176:
                            NULL,
1177: };
1178: static int char p654[] = {
1179:
                            XtNumber(seg0 654),XtNumber(seg1 654),
1180: };
1181: static XPoint seg0 655[] = {
                            \{-3,7\},\{-1,6\},\{0,5\},\{1,3\},\{1,1\},\{0,0\},\{-1,0\},\{-3,1\},\{-4,3\},
1182:
1183:
                            \{-4,6\},\{-3,8\},\{-1,9\},\{1,9\},\{3,8\},\{4,7\},\{6,4\},
1184: };
```

```
Listing C.4 Continued
```

```
1185: static XPoint *char655[] = {
1186:
                                            seg0 655,
1187:
                                            NULL,
1188: };
1189: static int char_p655[] = {
1190:
                                           XtNumber(seg0_655),
1191: };
1192: static XPoint seg0_656[] = {
1193:
                                            \{-3,4\},\{1,-1\},\{3,-4\},\{4,-6\},\{5,-9\},\{5,-11\},\{4,-12\},\{2,-11\},
1194:
                                             \{1, -9\}, \{-1, -1\}, \{-4, 8\}, \{-7, 15\}, \{-8, 18\}, \{-8, 20\}, \{-7, 21\}, \{-5, 20\}, \{-5, 20\}, \{-1, -1\}, \{-1, -1\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 20\}, \{-2, 2
1195:
                                             \{-4, 17\}, \{-3, 8\}, \{-2, 9\}, \{0, 9\}, \{2, 8\}, \{3, 7\}, \{5, 4\},
1196: \};
1197: static XPoint *char656[] = {
1198:
                                            seg0 656,
1199:
                                           NULL,
1200: };
1201: static int char_p656[] = {
1202:
                                           XtNumber(seg0 656),
1203: };
1204: static XPoint seg0_657[] = {
1205:
                                             \{3,3\},\{2,1\},\{0,0\},\{-2,0\},\{-4,1\},\{-5,2\},
1206:
                                             \{-6,4\},\{-6,6\},\{-5,8\},\{-3,9\},\{-1,9\},\{1,8\},\{2,7\},
1207: };
1208: static XPoint seg1_657[] = {
1209:
                                          \{4,0\},\{2,7\},\{-2,18\},\{-3,20\},\{-5,21\},\{-6,20\},\{-6,18\},\{-5,15\},
1210:
                                            \{-2, 12\}, \{1, 10\}, \{3, 9\}, \{6, 7\}, \{9, 4\}, 
1211: \};
1212: static XPoint *char657[] = {
1213:
                                            seg0 657, seg1 657,
                                            NULL,
1214:
1215: };
1216: static int char p657[] = {
1217:
                                           XtNumber(seg0_657),XtNumber(seg1_657),
1218: };
1219: static XPoint seg0 658[] = {
1220:
                                            \{-5,4\},\{-3,1\},\{0,-4\},\{1,-6\},
1221:
                                             \{2, -9\}, \{2, -11\}, \{1, -12\}, \{-1, -11\}, \{-2, -9\}, \{-3, -5\}, \{-4, 1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-5, 9\}, \{-
1222: };
1223: static XPoint seg1 658[] = {
1224:
                                             \{-5,9\},\{-4,6\},\{-3,4\},\{-1,1\},\{1,0\},\{3,0\},\{4,1\},\{4,3\},
1225:
                                             \{3,6\},\{3,8\},\{4,9\},\{5,9\},\{7,8\},\{8,7\},\{10,4\},
1226: \};
1227: static XPoint *char658[] = {
1228:
                                             seg0 658, seg1 658,
1229:
                                            NULL,
1230: };
1231: static int char_p658[] = {
1232:
                                           XtNumber(seg0 658),XtNumber(seg1 658),
1233: };
1234: static XPoint seg0 659[] = {
1235:
                                            \{1, -5\}, \{1, -4\},\
```

```
1236:
                                           \{2, -4\}, \{2, -5\}, \{1, -5\}, 
1237: \};
1238: static XPoint seg1 659[] = {
1239:
                                           \{-2,4\},\{0,0\},\{-2,6\},\{-2,8\},\{-1,9\},\{0,9\},\{2,8\},\{3,7\},
1240:
                                           {5,4},
1241: };
1242: static XPoint *char659[] = {
1243:
                                           seg0_659, seg1_659,
1244:
                                           NULL,
1245: \};
1246: static int char p659[] = {
                                           XtNumber(seg0 659),XtNumber(seg1 659),
1247:
1248: };
1249: static XPoint seg0 660[] = {
1250:
                                           \{1, -5\}, \{1, -4\}, \{2, -4\}, \{2, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{1, -5\}, \{2, -5\}, \{2, -5\}, \{1, -5\}, \{2, -5\}, \{2, -5\}, \{2, -5\}, \{3, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{
1251: };
1252: static XPoint seg1_660[] = {
1253:
                                           \{-2,4\},\{0,0\},\{-6,18\},\{-7,20\},\{-9,21\},\{-10,20\},\{-10,18\},\{-9,15\},
1254:
                                            \{-6, 12\}, \{-3, 10\}, \{-1, 9\}, \{2, 7\}, \{5, 4\},
1255: \};
1256: static XPoint *char660[] = {
1257:
                                           seg0 660, seg1 660,
1258:
                                           NULL,
1259: };
1260: static int char p660[] = {
1261:
                                           XtNumber(seg0_660),XtNumber(seg1_660),
1262: \};
1263: static XPoint seg0 661[] = {
1264:
                                           \{-5,4\},\{-3,1\},\{0,-4\},\{1,-6\},
                                            \{2, -9\}, \{2, -11\}, \{1, -12\}, \{-1, -11\}, \{-2, -9\}, \{-3, -5\}, \{-4, 1\}, \{-5, 9\}, \{-1, -11\}, \{-2, -9\}, \{-3, -5\}, \{-4, -1\}, \{-5, 9\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -
1265:
1266: \};
1267: static XPoint seg1_661[] = {
1268:
                                            \{-5,9\},\{-4,6\},\{-3,4\},\{-1,1\},\{1,0\},\{3,0\},\{4,1\},\{4,3\},
1269:
                                            \{2,4\},\{-1,4\},\
1270: };
1271: static XPoint seg2 661[] = {
1272:
                                            \{-1,4\},\{1,5\},\{2,8\},\{3,9\},\{4,9\},\{6,8\},\{7,7\},\{9,4\},
1273: \};
1274: static XPoint *char661[] = {
1275:
                                            seg0 661, seg1 661, seg2 661,
1276:
                                           NULL,
1277: };
1278: static int char p661[] = {
1279:
                                           XtNumber(seg0 661),XtNumber(seg1 661),XtNumber(seg2 661),
1280: };
1281: static XPoint seg0 662[] = {
1282:
                                            \{-3,4\},\{-1,1\},\{2,-4\},\{3,-6\},\{4,-9\},\{4,-11\},\{3,-12\},\{1,-11\},
1283:
                                            \{0, -9\}, \{-1, -5\}, \{-2, 2\}, \{-2, 8\}, \{-1, 9\}, \{0, 9\}, \{2, 8\}, \{3, 7\}, \{5, 4\},
1284: };
1285: static XPoint *char662[] = {
1286:
                                            seg0 662,
1287:
                                           NULL,
1288: \};
```

```
Listing C.4 Continued
```

```
1289: static int char p662[] = {
1290:
                             XtNumber(seg0 662),
1291: };
1292: static XPoint seg0_663[] = {
1293:
                              \{-13,4\},\
1294:
                              \{-11,1\},\{-9,0\},\{-8,1\},\{-8,2\},\{-9,6\},\{-10,9\},
1295: \};
1296: static XPoint seg1 663[] = {
                              \{-9,6\},\{-8,4\},\{-6,1\},\{-4,0\},\{-2,0\},\{-1,1\},\{-1,2\},\{-2,6\},
1297:
1298:
                              \{-3,9\},\
1299: };
1300: static XPoint seg2 663[] = {
1301:
                              \{-2,6\}, \{-1,4\}, \{1,1\}, \{3,0\}, \{5,0\}, \{6,1\}, \{6,3\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}, \{5,6\}
1302:
                              \{5,8\},\{6,9\},\{7,9\},\{9,8\},\{10,7\},\{12,4\},
1303: };
1304: static XPoint *char663[] = {
1305:
                              seg0 663, seg1 663, seg2 663,
1306:
                              NULL,
1307: };
1308: static int char_p663[] = {
1309:
                              XtNumber(seg0 663),XtNumber(seg1 663),XtNumber(seg2 663),
1310: };
1311: static XPoint seg0 664[] = {
1312:
                              \{-8,4\},\{-6,1\},\{-4,0\},
1313:
                              \{-3,1\},\{-3,2\},\{-4,6\},\{-5,9\},
1314: };
1315: static XPoint seg1 664[] = {
1316:
                              \{-4,6\},\{-3,4\},\{-1,1\},\{1,0\},\{3,0\},\{4,1\},\{4,3\},\{3,6\},
1317:
                              \{3,8\},\{4,9\},\{5,9\},\{7,8\},\{8,7\},\{10,4\},
1318: };
1319: static XPoint *char664[] = {
1320:
                              seg0 664, seg1 664,
1321:
                              NULL,
1322: };
1323: static int char p664[] = {
1324:
                             XtNumber(seg0_664),XtNumber(seg1_664),
1325: \};
1326: static XPoint seg0 665[] = {
1327:
                              \{0,0\},\{-2,0\},\{-4,1\},\
1328:
                              \{-5,2\},\{-6,4\},\{-6,6\},\{-5,8\},\{-3,9\},\{-1,9\},\{1,8\},\{2,7\},\{3,5\},
1329:
                              \{3,3\},\{2,1\},\{0,0\},\{-1,1\},\{-1,3\},\{0,5\},\{2,6\},\{5,6\},\{7,5\},
1330:
                              {8,4},
1331: };
1332: static XPoint *char665[] = {
1333:
                              seg0 665,
1334:
                              NULL,
1335: };
1336: static int char p665[] = {
1337:
                              XtNumber(seg0_665),
1338: };
```

```
1339: static XPoint seg0 666[] = {
           \{-7,4\},\{-5,1\},\{-4,-1\},\{-5,3\},\{-11,21\},
1340:
1341: };
1342: static XPoint seg1 666[] = {
1343:
           \{-5,3\},\{-4,1\},\{-2,0\},\{0,0\},\{2,1\},\{3,3\},\{3,5\},\{2,7\},
1344:
           \{1,8\},\{-1,9\},\
1345: };
1346: static XPoint seg2 666[] = {
1347:
           \{-5,8\},\{-3,9\},\{0,9\},\{3,8\},\{5,7\},\{8,4\},
1348: };
1349: static XPoint *char666[] = {
1350:
           seg0 666, seg1 666, seg2 666,
1351:
           NULL,
1352: \};
1353: static int char p666[] = {
1354:
           XtNumber(seg0 666),XtNumber(seg1 666),XtNumber(seg2 666),
1355: };
1356: static XPoint seg0 667[] = {
1357:
           \{3,3\},\{2,1\},\{0,0\},\{-2,0\},\{-4,1\},\{-5,2\},\{-6,4\},\{-6,6\},
1358:
           \{-5,8\},\{-3,9\},\{-1,9\},\{1,8\},
1359: };
1360: static XPoint seq1 667[] = {
1361:
           \{4,0\},\{3,3\},\{1,8\},\{-2,15\},\{-3,18\},\{-3,20\},\{-2,21\},\{0,20\},
1362:
           \{1,17\},\{1,10\},\{3,9\},\{6,7\},\{9,4\},
1363: };
1364: static XPoint *char667[] = {
1365:
           seg0 667, seg1 667,
1366:
           NULL,
1367: };
1368: static int char p667[] = {
1369:
           XtNumber(seg0 667),XtNumber(seg1 667),
1370: };
1371: static XPoint seg0 668[] = {
1372:
           \{-5,4\},\{-3,1\},\{-2,-1\},\{-2,1\},
1373:
           \{1,1\},\{2,2\},\{2,4\},\{1,7\},\{1,8\},\{2,9\},\{3,9\},\{5,8\},\{6,7\},
1374:
           {8,4},
1375: };
1376: static XPoint *char668[] = {
1377:
           seg0 668,
1378:
           NULL,
1379: };
1380: static int char p668[] = {
1381:
           XtNumber(seg0 668),
1382: \};
1383: static XPoint seg0 669[] = {
1384:
           \{-4,4\},\{-2,1\},\{-1,-1\},\{-1,1\},\{1,4\},\{2,6\},\{2,8\},\{0,9\},
1385: };
1386: static XPoint seg1 669[] = {
1387:
           \{-4,8\},\{-2,9\},\{2,9\},\{4,8\},\{5,7\},\{7,4\},
1388: };
1389: static XPoint *char669[] = {
1390:
           seg0 669, seg1 669,
1391:
           NULL,
1392: };
```

```
1393: static int char p669[] = {
           XtNumber(seg0 669),XtNumber(seg1 669),
1394:
1395: \};
1396: static XPoint seg0 670[] = {
1397:
           \{-3,4\},\{-1,1\},
1398:
           {1,-3},
1399: };
1400: static XPoint seg1 670[] = {
1401:
           \{4, -12\}, \{-2, 6\}, \{-2, 8\}, \{-1, 9\}, \{1, 9\}, \{3, 8\}, \{4, 7\}, \{6, 4\},
1402: };
1403: static XPoint seg2 670[] = {
1404:
           \{-2, -4\}, \{5, -4\}, 
1405: };
1406: static XPoint *char670[] = {
1407:
           seg0 670, seg1 670, seg2 670,
1408:
           NULL,
1409: };
1410: static int char p670[] = {
1411:
           XtNumber(seg0 670), XtNumber(seg1 670), XtNumber(seg2 670),
1412: };
1413: static XPoint seg0_671[] = {
1414:
           \{-6,4\},\{-4,0\},\{-6,6\},\{-6,8\},\{-5,9\},\{-3,9\},
1415:
           \{-1,8\},\{1,6\},\{3,3\},
1416: };
1417: static XPoint seg1_671[] = {
1418:
           \{4,0\},\{2,6\},\{2,8\},\{3,9\},\{4,9\},\{6,8\},\{7,7\},\{9,4\},
1419: };
1420: static XPoint *char671[] = {
1421:
           seg0 671, seg1 671,
           NULL,
1422:
1423: };
1424: static int char p671[] = {
1425:
           XtNumber(seg0 671), XtNumber(seg1 671),
1426: };
1427: static XPoint seg0 672[] = {
           \{-6,4\},\{-4,0\},\{-5,5\},\{-5,8\},\{-4,9\},\{-3,9\},\{0,8\},\{2,6\},\{3,3\},
1428:
1429:
           \{3, 0\},\
1430: };
1431: static XPoint seg1_672[] = {
1432:
           \{3,0\},\{4,4\},\{5,5\},\{7,5\},\{9,4\},
1433: };
1434: static XPoint *char672[] = {
1435:
           seg0 672, seg1 672,
1436:
           NULL,
1437: };
1438: static int char p672[] = {
1439:
           XtNumber(seg0 672),XtNumber(seg1 672),
1440: };
1441: static XPoint seg0 673[] = {
1442:
          \{-6,0\},\{-8,2\},\{-9,5\},
```

```
1443:
                           \{-9,7\},\{-8,9\},\{-6,9\},\{-4,8\},\{-2,6\},
1444: \};
1445: static XPoint seg1 673[] = {
1446:
                           \{0,0\},\{-2,6\},\{-2,8\},\{-1,9\},\{1,9\},\{3,8\},\{5,6\},\{6,3\},
1447:
                           {6,0},
1448: };
1449: static XPoint seg2 673[] = {
1450:
                           \{6,0\},\{7,4\},\{8,5\},\{10,5\},\{12,4\},
1451: };
1452: static XPoint *char673[] = {
1453:
                           seg0 673, seg1 673, seg2 673,
1454:
                           NULL,
1455: };
1456: static int char p673[] = {
1457:
                           XtNumber(seg0 673),XtNumber(seg1 673),XtNumber(seg2 673),
1458: };
1459: static XPoint seg0 674[] = {
1460:
                           \{-8,4\},\{-6,1\},\{-4,0\},
1461:
                           \{-2,0\},\{-1,1\},\{-1,8\},\{0,9\},\{3,9\},\{6,7\},\{8,4\},
1462: };
1463: static XPoint seg1 674[] = {
1464:
                           \{5,1\},\{4,0\},\{2,0\},\{1,1\},\{-3,8\},\{-4,9\},\{-6,9\},\{-7,8\},
1465: };
1466: static XPoint *char674[] = {
1467:
                           seg0 674, seg1 674,
1468:
                           NULL,
1469: };
1470: static int char p674[] = {
                          XtNumber(seg0 674), XtNumber(seg1 674),
1471:
1472: };
1473: static XPoint seg0 675[] = {
1474:
                           \{-6,4\},\{-4,0\},\{-6,6\},\{-6,8\},\{-5,9\},\{-3,9\},\{-1,8\},\{1,6\},\{3,3\},
1475: };
1476: static XPoint seg1 675[] = {
1477:
                           \{4,0\},\{-2,18\},\{-3,20\},\{-5,21\},\{-6,20\},\{-6,18\},\{-5,15\},\{-2,12\},
1478:
                           \{1,10\},\{3,9\},\{6,7\},\{9,4\},
1479: };
1480: static XPoint *char675[] = {
1481:
                           seg0 675, seg1 675,
1482:
                           NULL,
1483: };
1484: static int char p675[] = {
1485:
                          XtNumber(seg0 675),XtNumber(seg1 675),
1486: \};
1487: static XPoint seg0 676[] = {
1488:
                           \{-6,4\},\{-4,1\},\{-2,0\},\{0,0\},\{2,2\},\{2,4\},\{1,6\},\{-1,8\},\{-4,9\},
1489:
                           \{-2,10\},\{-1,12\},\{-1,15\},\{-2,18\},\{-3,20\},\{-5,21\},\{-6,20\},
1490:
                           \{-6, 18\}, \{-5, 15\}, \{-2, 12\}, \{1, 10\}, \{5, 7\}, \{8, 4\}, \{-6, 18\}, \{-5, 15\}, \{-2, 12\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\}, \{-1, 10\},
1491: };
1492: static XPoint *char676[] = {
1493:
                           seg0 676,
1494:
                           NULL,
1495: };
```
```
Listing C.4 Continued
```

```
1496: static int char_p676[] = {
1497:
                                                                    XtNumber(seg0 676),
1498: };
1499: static XPoint seg0 699[] = {
1500:
                                                                      {-8,8},
1501: };
1502: static XPoint *char699[] = {
1503:
                                                                      seg0 699,
1504:
                                                                      NULL,
1505: };
1506: static int char p699[] = {
1507:
                                                                    XtNumber(seg0 699),
1508: };
1509: static XPoint seg0 700[] = {
1510:
                                                                      \{-1, -12\}, \{-4, -11\}, \{-6, -8\}, \{-7, -3\}, \{-7, 0\}, \{-6, 5\}, \{-4, 8\}, \{-1, 9\},
1511:
                                                                       \{1,9\},\{4,8\},\{6,5\},\{7,0\},\{7,-3\},\{6,-8\},\{4,-11\},\{1,-12\},\{-1,-12\},
1512: };
1513: static XPoint *char700[] = {
1514:
                                                                      seg0_700,
1515:
                                                                      NULL,
1516: };
1517: static int char p700[] = {
1518:
                                                                    XtNumber(seg0 700),
1519: };
1520: static XPoint seg0 701[] = {
1521:
                                                                   {-4,-8},
1522:
                                                                      \{-2, -9\}, \{1, -12\}, \{1, 9\}, 
1523: };
1524: static XPoint *char701[] = {
1525:
                                                                      seg0_701,
1526:
                                                                      NULL,
1527: };
1528: static int char_p701[] = {
1529:
                                                                    XtNumber(seg0 701),
1530: };
1531: static XPoint seg0 702[] = {
1532:
                                                                      \{-6, -7\}, \{-6, -8\}, \{-5, -10\}, \{-4, -11\}, \{-2, -12\}, \{2, -12\}, \{2, -12\}, \{-6, -7\}, \{-6, -8\}, \{-2, -10\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2,
1533:
                                                                       \{4, -11\}, \{5, -10\}, \{6, -8\}, \{6, -6\}, \{5, -4\}, \{3, -1\}, \{-7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7,
1534: };
1535: static XPoint *char702[] = {
1536:
                                                                      seg0 702,
1537:
                                                                      NULL,
1538: \};
1539: static int char p702[] = {
1540:
                                                                    XtNumber(seg0_702),
1541: };
1542: static XPoint seg0_703[] = {
1543:
                                                                      \{-5, -12\}, \{6, -12\}, \{0, -4\}, \{3, -4\}, \{5, -3\}, \{6, -2\}, \{7, 1\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7,
1544:
                                                                       \{6,6\},\{4,8\},\{1,9\},\{-2,9\},\{-5,8\},\{-6,7\},\{-7,5\},
1545: };
```

```
1546: static XPoint *char703[] = {
1547:
                                                seg0_703,
1548:
                                                NULL,
1549: };
1550: static int char_p703[] = {
                                               XtNumber(seg0 703),
1551:
1552: };
1553: static XPoint seg0 704[] = {
1554:
                                                \{3, -12\}, \{-7, 2\}, \{8, 2\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12
1555: \};
1556: static XPoint seg1 704[] = {
1557:
                                                \{3, -12\}, \{3, 9\},\
1558: \};
1559: static XPoint *char704[] = {
1560:
                                                seg0 704, seg1 704,
1561:
                                                NULL,
1562: };
1563: static int char p704[] = {
1564:
                                                XtNumber(seg0_704),XtNumber(seg1_704),
1565: \};
1566: static XPoint seg0 705[] = {
1567:
                                                \{5, -12\}, \{-5, -12\}, \{-6, -3\}, \{-5, -4\}, \{-2, -5\}, \{1, -5\},
1568:
                                                \{4, -4\}, \{6, -2\}, \{7, 1\}, \{7, 3\}, \{6, 6\}, \{4, 8\}, \{1, 9\}, \{-2, 9\}, \{-5, 8\},
1569:
                                                \{-6,7\},\{-7,5\},
1570: };
1571: static XPoint *char705[] = {
1572:
                                                seg0 705,
1573:
                                                NULL,
1574: };
1575: static int char p705[] = {
1576:
                                               XtNumber(seg0_705),
1577: };
1578: static XPoint seg0 706[] = {
1579:
                                                \{6, -9\}, \{5, -11\}, \{2, -12\}, \{0, -12\}, \{-3, -11\}, \{-5, -8\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6,
1580:
                                                \{-6,2\},\{-5,6\},\{-3,8\},\{0,9\},\{1,9\},\{4,8\},\{6,6\},\{7,3\},\{7,2\},
1581:
                                                \{6, -1\}, \{4, -3\}, \{1, -4\}, \{0, -4\}, \{-3, -3\}, \{-5, -1\}, \{-6, 2\},\
1582: };
1583: static XPoint *char706[] = {
1584:
                                                seg0 706,
1585:
                                                NULL,
1586: \};
1587: static int char p706[] = {
1588:
                                               XtNumber(seg0 706),
1589: \};
1590: static XPoint seg0 707[] = {
1591:
                                               \{7, -12\}, \{-3, 9\},\
1592: };
1593: static XPoint seg1 707[] = {
1594:
                                               \{-7, -12\}, \{7, -12\}, 
1595: \};
1596: static XPoint *char707[] = {
1597:
                                                seg0 707, seg1 707,
1598:
                                                NULL,
1599: };
```

```
1600: static int char p707[] = {
                                               XtNumber(seg0 707),XtNumber(seg1 707),
1601:
1602: };
1603: static XPoint seg0_708[] = {
1604:
                                                \{-2, -12\}, \{-5, -11\}, \{-6, -9\}, \{-6, -7\}, \{-5, -5\}, \{-3, -4\},
1605:
                                                \{1, -3\}, \{4, -2\}, \{6, 0\}, \{7, 2\}, \{7, 5\}, \{6, 7\}, \{5, 8\}, \{2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}
1606:
                                                \{-5,8\},\{-6,7\},\{-7,5\},\{-7,2\},\{-6,0\},\{-4,-2\},\{-1,-3\},\{3,-4\},
1607:
                                                \{5, -5\}, \{6, -7\}, \{6, -9\}, \{5, -11\}, \{2, -12\}, \{-2, -12\}, 
1608: };
1609: static XPoint *char708[] = {
1610:
                                               seg0_708,
1611:
                                                NULL,
1612: };
1613: static int char p708[] = {
1614:
                                               XtNumber(seg0 708),
1615: };
1616: static XPoint seg0 709[] = {
1617:
                                                \{6, -5\}, \{5, -2\}, \{3, 0\}, \{0, 1\}, \{-1, 1\}, \{-4, 0\}, \{-6, -2\}, \{-7, -5\},
1618:
                                                \{-7, -6\}, \{-6, -9\}, \{-4, -11\}, \{-1, -12\}, \{0, -12\}, \{3, -11\}, \{5, -9\}, \{-7, -6\}, \{-6, -9\}, \{-4, -11\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -
1619:
                                                \{6, -5\}, \{6, 0\}, \{5, 5\}, \{3, 8\}, \{0, 9\}, \{-2, 9\}, \{-5, 8\}, \{-6, 6\},
1620: \};
1621: static XPoint *char709[] = {
1622:
                                                seg0 709,
1623:
                                                NULL,
1624: };
1625: static int char p709[] = {
                                               XtNumber(seg0_709),
1626:
1627: \};
1628: static XPoint seg0 710[] = {
1629:
                                                \{0,7\},\{-1,8\},\{0,9\},\{1,8\},\{0,7\},
1630: };
1631: static XPoint *char710[] = {
1632:
                                                seg0 710,
1633:
                                                NULL,
1634: };
1635: static int char_p710[] = {
1636:
                                                XtNumber(seg0 710),
1637: };
1638: static XPoint seg0_711[] = {
1639:
                                                \{1,8\},\{0,9\},\{-1,8\},\
1640:
                                                \{0,7\},\{1,8\},\{1,10\},\{0,12\},\{-1,13\},
1641: \};
1642: static XPoint *char711[] = {
1643:
                                                seg0 711,
1644:
                                                NULL,
1645: };
1646: static int char p711[] = {
1647:
                                                XtNumber(seg0 711),
1648: };
```

```
1649: static XPoint seg0 712[] = {
1650:
                                                                                                    \{0, -5\}, \{-1, -4\}, \{0, -3\}, \{1, -4\}, \{0, -3\}, \{1, -4\}, \{0, -3\}, \{1, -4\}, \{0, -3\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{2, -4\}, \{2, -4\}, \{2, -4\}, \{2, -4\}, \{2, -4\}, \{3, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, \{4, -4\}, 
1651:
                                                                                                    \{0, -5\},\
1652: };
1653: static XPoint seg1_712[] = {
1654:
                                                                                                 \{0,7\},\{-1,8\},\{0,9\},\{1,8\},\{0,7\},
1655: \};
1656: static XPoint *char712[] = {
1657:
                                                                                                    seg0 712, seg1 712,
1658:
                                                                                                 NULL,
1659: \};
1660: static int char p712[] = {
 1661:
                                                                                                  XtNumber(seg0 712),XtNumber(seg1 712),
1662: \};
1663: static XPoint seg0 713[] = {
1664:
                                                                                                    \{0, -5\}, \{-1, -4\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, \{0, -3\}, 
1665:
                                                                                                  \{1, -4\}, \{0, -5\},\
1666: };
1667: static XPoint seg1 713[] = {
1668:
                                                                                                  \{1,8\},\{0,9\},\{-1,8\},\{0,7\},\{1,8\},\{1,10\},\{0,12\},\{-1,13\},
1669: };
1670: static XPoint *char713[] = {
1671:
                                                                                                  seg0_713, seg1_713,
1672:
                                                                                                 NULL,
1673: };
1674: static int char p713[] = {
 1675:
                                                                                                  XtNumber(seg0 713),XtNumber(seg1 713),
1676: };
1677: static XPoint seg0 714[] = {
1678:
                                                                                                    \{0, -12\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0
1679: };
1680: static XPoint seg1_714[] = {
1681:
                                                                                                  \{0,7\},\{-1,8\},\{0,9\},\{1,8\},\{0,7\},
1682: };
1683: static XPoint *char714[] = {
1684:
                                                                                                  seg0 714, seg1 714,
1685:
                                                                                                  NULL,
1686: };
1687: static int char p714[] = {
1688:
                                                                                                  XtNumber(seg0 714),XtNumber(seg1 714),
1689: };
1690: static XPoint seg0 715[] = {
1691:
                                                                                                  \{-6, -7\}, \{-6, -8\}, \{-5, -10\}, \{-4, -11\}, \{-2, -12\}, \{2, -12\}, \{4, -11\}, \{-6, -7\}, \{-6, -8\}, \{-5, -10\}, \{-6, -11\}, \{-2, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7,
1692:
                                                                                                    \{5, -10\}, \{6, -8\}, \{6, -6\}, \{5, -4\}, \{4, -3\}, \{0, -1\}, \{0, 2\}, \{0, -1\}, \{0, 2\}, \{0, -1\}, \{0, 2\}, \{0, -1\}, \{0, 2\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, 
1693: };
1694: static XPoint seg1 715[] = {
1695:
                                                                                                  \{0,7\},\{-1,8\},\{0,9\},\{1,8\},\{0,7\},
1696: };
1697: static XPoint *char715[] = {
1698:
                                                                                                  seg0 715, seg1 715,
1699:
                                                                                                  NULL,
1700: };
1701: static int char_p715[] = {
1702:
                                                                                                  XtNumber(seg0_715),XtNumber(seg1_715),
1703: };
```

```
Listing C.4 Continued
```

```
1704: static XPoint seg0 717[] = {
1705:
                                               {-4,-12},
1706:
                                                  \{-4, -5\},\
1707: };
1708: static XPoint seg1 717[] = {
1709:
                                                 \{4, -12\}, \{4, -5\},\
1710: \};
1711: static XPoint *char717[] = {
1712:
                                                 seg0 717, seg1 717,
1713:
                                                 NULL,
1714: };
1715: static int char p717[] = {
1716:
                                                 XtNumber(seg0_717),XtNumber(seg1_717),
1717: };
1718: static XPoint seg0 718[] = {
1719:
                                                 \{-1, -12\}, \{-3, -11\}, \{-4, -9\}, \{-4, -7\}, \{-3, -5\}, \{-1, -4\},
1720:
                                                  \{1, -4\}, \{3, -5\}, \{4, -7\}, \{4, -9\}, \{3, -11\}, \{1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12
1721: };
1722: static XPoint *char718[] = {
1723:
                                                 seg0 718,
1724:
                                                 NULL,
1725: };
1726: static int char_p718[] = {
1727:
                                                 XtNumber(seg0 718),
1728: };
1729: static XPoint seg0 719[] = {
1730:
                                                 \{-2, -16\}, \{-2, 13\}, 
1731: };
1732: static XPoint seg1 719[] = {
1733:
                                                \{2, -16\}, \{2, 13\},\
1734: };
1735: static XPoint seg2_719[] = {
1736:
                                                 \{7, -9\}, \{5, -11\}, \{2, -12\}, \{-2, -12\}, \{-5, -11\}, \{-7, -9\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7
1737:
                                                  \{-6, -5\}, \{-5, -4\}, \{-3, -3\}, \{3, -1\}, \{5, 0\}, \{6, 1\}, \{7, 3\}, \{7, 6\},
1738:
                                                 \{5,8\},\{2,9\},\{-2,9\},\{-5,8\},\{-7,6\},
1739: };
1740: static XPoint *char719[] = {
1741:
                                                  seg0_719, seg1_719, seg2_719,
1742:
                                                 NULL,
1743: };
1744: static int char_p719[] = {
1745:
                                                 XtNumber(seg0 719),XtNumber(seg1 719),XtNumber(seg2 719),
1746: };
1747: static XPoint seg0 720[] = {
1748:
                                                  \{9, -16\}, \{-9, 16\}, 
1749: };
1750: static XPoint *char720[] = {
1751:
                                                 seg0 720,
1752:
                                                 NULL,
1753: };
```

```
1754: static int char p720[] = {
1755:
                                XtNumber(seg0_720),
1756: };
1757: static XPoint seg0 721[] = {
1758:
                                \{4, -16\}, \{2, -14\}, \{0, -11\}, \{-2, -7\},\
1759:
                                 \{-3, -2\}, \{-3, 2\}, \{-2, 7\}, \{0, 11\}, \{2, 14\}, \{4, 16\},
1760: };
1761: static XPoint *char721[] = {
1762:
                                 seg0 721,
1763:
                                NULL,
1764: };
1765: static int char p721[] = {
1766:
                                XtNumber(seg0 721),
1767: };
1768: static XPoint seg0 722[] = {
1769:
                                 \{-4, -16\}, \{-2, -14\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0
1770:
                                \{2, -7\}, \{3, -2\}, \{3, 2\}, \{2, 7\}, \{0, 11\}, \{-2, 14\}, \{-4, 16\},
1771: };
1772: static XPoint *char722[] = {
1773:
                                seg0_722,
1774:
                                NULL,
1775: };
1776: static int char_p722[] = {
1777:
                                XtNumber(seg0_722),
1778: };
1779: static XPoint seg0_723[] = {
1780:
                               \{0, -16\}, \{0, 16\}, 
1781: };
1782: static XPoint *char723[] = {
1783:
                                seg0 723,
1784:
                                NULL,
1785: };
1786: static int char p723[] = {
1787:
                                XtNumber(seg0 723),
1788: };
1789: static XPoint seg0 724[] = {
1790:
                                \{-9,0\},\{9,0\},
1791: };
1792: static XPoint *char724[] = {
1793:
                                seg0 724,
1794:
                                NULL,
1795: };
1796: static int char p724[] = {
1797:
                                XtNumber(seg0 724),
1798: };
1799: static XPoint seg0_725[] = {
1800:
                                \{0, -9\}, \{0, 9\}, 
1801: };
1802: static XPoint seg1_725[] = {
1803:
                                \{-9,0\},\{9,0\},
1804: };
```

```
Listing C.4 Continued
```

```
1805: static XPoint *char725[] = {
1806:
                                                      seg0_725, seg1_725,
1807:
                                                      NULL,
1808: };
1809: static int char p725[] = {
1810:
                                                     XtNumber(seg0_725),XtNumber(seg1_725),
1811: };
1812: static XPoint seg0_726[] = {
1813:
                                                    \{-9, -3\}, \{9, -3\}, 
1814: };
1815: static XPoint seg1_726[] = {
1816:
                                                      \{-9,3\},\{9,3\},
1817: };
1818: static XPoint *char726[] = {
1819:
                                                      seg0_726, seg1_726,
1820:
                                                      NULL,
1821: };
1822: static int char_p726[] = {
1823:
                                                     XtNumber(seg0 726),XtNumber(seg1 726),
1824: };
1825: static XPoint seg0_730[] = {
1826:
                                                      \{1, -12\},\
1827:
                                                       \{0, -11\}, \{-1, -9\}, \{-1, -7\}, \{0, -6\}, \{1, -7\}, \{0, -8\},
1828: };
1829: static XPoint *char730[] = {
1830:
                                                      seg0 730,
1831:
                                                      NULL,
1832: };
1833: static int char p730[] = {
                                                     XtNumber(seg0_730),
1834:
1835: };
1836: static XPoint seg0_731[] = {
1837:
                                                      \{0, -10\}, \{-1, -11\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0,
1838:
                                                      \{1, -11\}, \{1, -9\}, \{0, -7\}, \{-1, -6\},\
1839: };
1840: static XPoint *char731[] = {
1841:
                                                      seg0 731,
1842:
                                                      NULL,
1843: };
1844: static int char p731[] = {
1845:
                                                     XtNumber(seg0_731),
1846: \};
1847: static XPoint seg0 733[] = {
1848:
                                                    \{1, -16\}, \{-6, 16\}, 
1849: };
1850: static XPoint seg1_733[] = {
1851:
                                                     \{7, -16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{0, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, \{1, 16\}, 
1852: };
1853: static XPoint seg2 733[] = {
1854:
                                                      \{-6, -3\}, \{8, -3\}, \{
1855: \};
```

```
1856: static XPoint seg3 733[] = {
                                              \{-7,3\},\{7,3\},
1857:
1858: };
1859: static XPoint *char733[] = {
                                              seg0_733, seg1_733, seg2_733, seg3_733,
1860:
1861:
                                             NULL,
1862: };
1863: static int char p733[] = {
                                              XtNumber(seg0 733), XtNumber(seg1 733), XtNumber(seg2 733),
1864:
1865:
                                             XtNumber(seg3 733),
1866: };
1867: static XPoint seg0 734[] = {
1868:
                                             \{10, -3\}, \{10, -4\}, \{9, -5\}, \{8, -5\}, \{7, -4\}, \{6, -2\}, \{4, 3\}, \{2, 6\}, \{0, 8\},
1869:
                                              \{-2,9\},\{-6,9\},\{-8,8\},\{-9,7\},\{-10,5\},\{-10,3\},\{-9,1\},\{-8,0\},
1870:
                                              \{-1, -4\}, \{0, -5\}, \{1, -7\}, \{1, -9\}, \{0, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -11\}, \{-2, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -12\}, \{-4, -
1871:
                                              \{-5, -9\}, \{-5, -7\}, \{-4, -4\}, \{-2, -1\}, \{3, 6\}, \{5, 8\}, \{7, 9\}, \{9, 9\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{10, 8\}, \{
1872:
                                             {10,7},
1873: };
1874: static XPoint *char734[] = {
1875:
                                             seg0_734,
1876:
                                             NULL,
1877: };
1878: static int char_p734[] = {
1879:
                                            XtNumber(seg0 734),
1880: };
1881: static XPoint seg0 804[] = {
1882:
                                            \{-7, -12\}, \{7, 12\}, 
1883: };
1884: static XPoint *char804[] = {
1885:
                                              seg0 804,
1886:
                                             NULL,
1887: };
1888: static int char p804[] = {
1889:
                                             XtNumber(seg0 804),
1890: };
1891: static XPoint seg0 834[] = {
1892:
                                              \{-14,0\},\{14,0\},
1893: };
1894: static XPoint seg1 834[] = {
1895:
                                             \{-14,0\},\{0,16\},\
1896: \};
1897: static XPoint seg2 834[] = {
1898:
                                             \{14,0\},\{0,16\},\
1899: };
1900: static XPoint *char834[] = {
1901:
                                              seg0 834, seg1 834, seg2 834,
1902:
                                             NULL,
1903: };
1904: static int char_p834[] = {
1905:
                                             XtNumber(seg0 834), XtNumber(seg1 834), XtNumber(seg2 834),
1906: };
1907: static XPoint seg0 840[] = {
1908:
                                              \{-1, -7\}, \{-4, -6\}, \{-6, -4\}, \{-7, -1\}, \{-7, 1\}, \{-6, 4\},
```

```
1909:
                                 \{-4,6\},\{-1,7\},\{1,7\},\{4,6\},\{6,4\},\{7,1\},\{7,-1\},\{6,-4\},\{4,-6\},
1910:
                                 \{1, -7\}, \{-1, -7\}, 
1911: };
1912: static XPoint *char840[] = {
1913:
                                 seg0_840,
1914:
                                 NULL,
1915: };
1916: static int char_p840[] = {
1917:
                                XtNumber(seg0 840),
1918: };
1919: static XPoint seg0 844[] = {
1920:
                                \{0, -9\}, \{-2, -3\}, 
1921:
                                 \{-8, -3\}, \{-3, 1\}, \{-5, 7\}, \{0, 3\}, \{5, 7\}, \{3, 1\}, \{8, -3\}, \{2, -3\}, \{0, -9\},
1922: };
1923: static XPoint *char844[] = {
1924:
                                 seg0 844,
1925:
                                 NULL,
1926: };
1927: static int char_p844[] = {
1928:
                                XtNumber(seg0_844),
1929: };
1930: static XPoint seg0 845[] = {
1931:
                                \{0, -7\}, \{0, 7\},\
1932: };
1933: static XPoint seg1 845[] = {
1934:
                                \{-7,0\},\{7,0\},
1935: };
1936: static XPoint *char845[] = {
1937:
                                 seg0 845, seg1 845,
1938:
                                 NULL,
1939: };
1940: static int char p845[] = {
1941:
                                XtNumber(seg0_845),XtNumber(seg1_845),
1942: };
1943: static XPoint seg0_847[] = {
1944:
                                \{0, -6\}, \{0, 6\},\
1945: };
1946: static XPoint seg1_847[] = {
1947:
                                \{-5, -3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5, 3\}, \{5
1948: };
1949: static XPoint seg2 847[] = {
1950:
                                 \{5, -3\}, \{-5, 3\}, 
1951: };
1952: static XPoint *char847[] = {
1953:
                                 seg0_847, seg1_847, seg2_847,
1954:
                                 NULL,
1955: };
1956: static int char p847[] = {
1957:
                                 XtNumber(seg0_847),XtNumber(seg1_847),XtNumber(seg2_847),
1958: };
```

```
1959: static XPoint seg0 850[] = {
1960:
                             \{-1, -4\}, \{-3, -3\}, \{-4, -1\}, \{-4, 1\}, \{-3, 3\}, \{-1, 4\},
1961:
                             \{1,4\},\{3,3\},\{4,1\},\{4,-1\},\{3,-3\},\{1,-4\},\{-1,-4\},
1962: };
1963: static XPoint seg1_850[] = {
1964:
                            \{-3, -1\}, \{-3, 1\},
1965: };
1966: static XPoint seg2 850[] = {
1967:
                             \{-2, -2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, 
1968: };
1969: static XPoint seg3 850[] = {
1970:
                             \{-1, -3\}, \{-1, 3\}, 
1971: };
1972: static XPoint seg4 850[] = {
1973:
                             \{0, -3\}, \{0, 3\},\
1974: };
1975: static XPoint seg5 850[] = {
1976:
                            \{1, -3\}, \{1, 3\}, 
1977: };
1978: static XPoint seg6 850[] = {
1979:
                             \{2, -2\}, \{2, 2\}, 
1980: };
1981: static XPoint seg7_850[] = {
1982:
                             \{3, -1\}, \{3, 1\},\
1983: };
1984: static XPoint *char850[] = {
1985:
                             seg0 850, seg1 850, seg2 850, seg3 850, seg4 850, seg5 850,
                             seg6_850,seg7 850,
1986:
1987:
                             NULL,
1988: };
1989: static int char p850[] = {
                             XtNumber(seg0_850),XtNumber(seg1_850),XtNumber(seg2_850),
1990:
1991:
                             XtNumber(seg3 850),XtNumber(seg4 850),XtNumber(seg5 850),
1992:
                             XtNumber(seg6 850),XtNumber(seg7 850),
1993: };
1994: static XPoint seg0 855[] = {
1995:
                             \{6,0\},\{-3,-5\},\{-3,5\},\{6,0\},
1996: \};
1997: static XPoint seg1 855[] = {
1998:
                             \{3,0\},\{-2,-3\},\
1999: \};
2000: static XPoint seg2 855[] = {
2001:
                             \{3,0\},\{-2,3\},\
2002: };
2003: static XPoint seg3 855[] = {
2004:
                             \{0,0\},\{-2,-1\},\
2005: };
2006: static XPoint seg4 855[] = {
2007:
                             \{0,0\},\{-2,1\},\
2008: };
2009: static XPoint *char855[] = {
2010:
                             seg0 855, seg1 855, seg2 855, seg3 855, seg4 855,
2011:
                             NULL,
2012: };
```

```
2013: static int char p855[] = {
2014:
                                                  XtNumber(seg0 855),XtNumber(seg1 855),XtNumber(seg2 855),
2015:
                                                  XtNumber(seg3 855),XtNumber(seg4 855),
2016: };
2017: static XPoint seg0 866[] = {
2018:
                                                  \{-2, -6\}, \{-2, -2\}, \{-6, -2\}, \{-6, 2\}, 
2019:
                                                   \{-2,2\},\{-2,6\},\{2,6\},\{2,2\},\{6,2\},\{6,-2\},\{2,-2\},\{2,-6\},\{-2,-6\},
2020: };
2021: static XPoint *char866[] = {
2022:
                                                   seg0_866,
2023:
                                                  NULL,
2024: };
2025: static int char_p866[] = {
2026:
                                                  XtNumber(seq0 866),
2027: };
2028: static XPoint seg0 999[] = {
2029:
                                                   \{-8,11\},\{8,11\},
2030: };
2031: static XPoint *char999[] = {
2032:
                                                  seg0 999,
                                                  NULL,
2033:
2034: };
2035: static int char p999[] = {
2036:
                                                  XtNumber(seg0 999),
2037: };
2038: static XPoint seg0 2219[] = {
2039:
                                                  \{0, -12\}, \{0, 0\},\
2040: };
2041: static XPoint seg1 2219[] = {
2042:
                                                  \{-5, -9\}, \{5, -3\},
2043: };
2044: static XPoint seg2 2219[] = {
2045:
                                                  \{5, -9\}, \{-5, -3\},\
2046: };
2047: static XPoint *char2219[] = {
2048:
                                                   seg0_2219, seg1_2219, seg2_2219,
2049:
                                                  NULL,
2050: };
2051: static int char p2219[] = {
                                                  XtNumber(seg0_2219),XtNumber(seg1_2219),XtNumber(seg2_2219),
2052:
2053: };
2054: static XPoint seg0 2223[] = {
2055:
                                                  \{-3, -16\}, \{-3, 16\}, 
2056: };
2057: static XPoint seg1 2223[] = {
2058:
                                                  \{-2, -16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2
2059: };
2060: static XPoint seg2 2223[] = {
2061:
                                                   \{-3, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4, -16\}, \{4,
2062: };
```

```
2063: static XPoint seg3 2223[] = {
2064:
                                                          \{-3, 16\}, \{4, 16\}, 
2065: };
2066: static XPoint *char2223[] = {
2067:
                                                          seg0_2223, seg1_2223, seg2_2223, seg3_2223,
2068:
                                                         NULL,
2069: };
2070: static int char p2223[] = {
2071:
                                                          XtNumber(seg0 2223), XtNumber(seg1 2223), XtNumber(seg2 2223),
2072:
                                                         XtNumber(seg3 2223),
2073: };
2074: static XPoint seg0 2224[] = {
2075:
                                                         \{2, -16\}, \{2, 16\}, 
2076: };
2077: static XPoint seg1 2224[] = {
2078:
                                                          \{3, -16\}, \{3, 16\}, 
2079: };
2080: static XPoint seg2 2224[] = {
2081:
                                                          {-4,-16},{3,-16},
2082: };
2083: static XPoint seg3 2224[] = {
2084:
                                                         \{-4, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, \{3, 16\}, 
2085: };
2086: static XPoint *char2224[] = {
2087:
                                                          seg0 2224, seg1 2224, seg2 2224, seg3 2224,
2088:
                                                         NULL,
2089: };
2090: static int char p2224[] = {
                                                         XtNumber(seg0 2224), XtNumber(seg1 2224), XtNumber(seg2 2224),
2091:
2092:
                                                         XtNumber(seg3 2224),
2093: };
2094: static XPoint seg0_2225[] = {
2095:
                                                          \{2, -16\}, \{0, -15\}, \{-1, -14\}, \{-2, -12\}, \{-2, -10\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-1, -8\}, \{-2, -10\}, \{-2, -10\}, \{-1, -8\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -1
2096:
                                                          \{0, -7\}, \{1, -5\}, \{1, -3\}, \{-1, -1\},\
2097: };
2098: static XPoint seg1 2225[] = {
2099:
                                                          \{0, -15\}, \{-1, -13\}, \{-1, -11\}, \{0, -9\}, \{1, -8\}, \{2, -6\}, \{2, -4\}, \{1, -2\}, \{1, -2\}, \{1, -2\}, \{1, -2\}, \{2, -4\}, \{1, -2\}, \{1, -2\}, \{2, -4\}, \{1, -2\}, \{2, -4\}, \{2, -4\}, \{1, -2\}, \{2, -4\}, \{3, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -
2100:
                                                          \{-3,0\},\{1,2\},\{2,4\},\{2,6\},\{1,8\},\{0,9\},\{-1,11\},\{-1,13\},\{0,15\},
2101: };
2102: static XPoint seg2_2225[] = {
2103:
                                                          \{-1,1\},\{1,3\},\{1,5\},\{0,7\},\{-1,8\},\{-2,10\},\{-2,12\},\{-1,14\},
2104:
                                                          \{0, 15\}, \{2, 16\}, 
2105: };
2106: static XPoint *char2225[] = {
2107:
                                                          seg0 2225, seg1 2225, seg2 2225,
2108:
                                                         NULL,
2109: };
2110: static int char p2225[] = {
2111:
                                                         XtNumber(seg0 2225), XtNumber(seg1 2225), XtNumber(seg2 2225),
2112: };
2113: static XPoint seg0 2226[] = {
2114:
                                                          \{-2, -16\}, \{0, -15\}, \{1, -14\}, \{2, -12\}, \{2, -10\}, \{1, -8\}, \{0, -7\}, 
2115:
                                                          \{-1, -5\}, \{-1, -3\}, \{1, -1\},
2116: };
```

```
2117: static XPoint seg1_2226[] = {
2118:
           \{0, -15\}, \{1, -13\}, \{1, -11\}, \{0, -9\}, \{-1, -8\}, \{-2, -6\}, \{-2, -4\}, \{-1, -2\}, 
2119:
           \{3,0\},\{-1,2\},\{-2,4\},\{-2,6\},\{-1,8\},\{0,9\},\{1,11\},\{1,13\},\{0,15\},
2120: };
2121: static XPoint seg2 2226[] = {
2122:
           \{1,1\},\{-1,3\},\{-1,5\},\{0,7\},\{1,8\},\{2,10\},\{2,12\},\{1,14\},
           {0,15},{-2,16},
2123:
2124: };
2125: static XPoint *char2226[] = {
2126:
           seg0_2226, seg1_2226, seg2_2226,
2127:
           NULL,
2128: };
2129: static int char p2226[] = {
2130:
           XtNumber(seg0 2226), XtNumber(seg1 2226), XtNumber(seg2 2226),
2131: };
2132: static XPoint seg0_2229[] = {
2133:
           {0,-16},
2134:
           \{0, 16\},\
2135: };
2136: static XPoint *char2229[] = {
2137:
           seg0 2229,
2138:
           NULL,
2139: };
2140: static int char p2229[] = {
2141:
           XtNumber(seg0 2229),
2142: };
2143: static XPoint seg0 2241[] = {
2144:
           \{8, -9\}, \{-8, 0\}, \{8, 9\}, \{
2145: };
2146: static XPoint *char2241[] = {
2147:
           seg0_2241,
2148:
           NULL,
2149: };
2150: static int char p2241[] = {
2151:
           XtNumber(seg0 2241),
2152: };
2153: static XPoint seg0 2242[] = {
2154:
           \{-8, -9\}, \{8, 0\}, \{-8, 9\},
2155: \};
2156: static XPoint *char2242[] = {
2157:
           seg0 2242,
2158:
           NULL,
2159: };
2160: static int char p2242[] = {
2161:
           XtNumber(seg0_2242),
2162: };
2163: static XPoint seg0_2246[] = {
2164:
           \{-9,3\},\{-9,1\},\{-8,-2\},\{-6,-3\},
2165:
           \{-4, -3\}, \{-2, -2\}, \{2, 1\}, \{4, 2\}, \{6, 2\}, \{8, 1\}, \{9, -1\},
2166: };
```

```
2167: static XPoint seg1 2246[] = {
2168:
                                                                                           \{-9,1\},\{-8,-1\},\{-6,-2\},\{-4,-2\},\{-2,-1\},\{2,2\},\{4,3\},\{6,3\},
2169:
                                                                                           \{8,2\},\{9,-1\},\{9,-3\},\
2170: };
2171: static XPoint *char2246[] = {
                                                                                          seg0 2246, seg1 2246,
2172:
2173:
                                                                                          NULL,
2174: };
2175: static int char p2246[] = {
2176:
                                                                                        XtNumber(seg0 2246),XtNumber(seg1 2246),
2177: };
2178: static XPoint seg0 2262[] = {
2179:
                                                                                       \{-2, -6\}, \{0, -9\}, \{2, -6\}, \{0, -9\}, \{2, -6\}, \{2, -6\}, \{2, -6\}, \{2, -6\}, \{2, -6\}, \{2, -6\}, \{2, -6\}, \{2, -6\}, \{3, -6\}, \{3, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, \{4, -6\}, 
2180: };
2181: static XPoint seg1 2262[] = {
2182:
                                                                                           \{-5, -3\}, \{0, -8\}, \{5, -3\}, \{0, -8\}, \{5, -3\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, 
2183: };
2184: static XPoint seg2 2262[] = {
2185:
                                                                                          \{0, -8\}, \{0, 9\},\
2186: };
2187: static XPoint *char2262[] = {
2188:
                                                                                          seg0 2262, seg1 2262, seg2 2262,
2189:
                                                                                          NULL,
2190: };
2191: static int char p2262[] = {
2192:
                                                                                          XtNumber(seg0_2262),XtNumber(seg1_2262),XtNumber(seg2_2262),
2193: };
2194: static XPoint seg0 2271[] = {
2195:
                                                                                          \{9, -12\}, \{-9, 9\}, 
2196: };
2197: static XPoint seg1 2271[] = {
2198:
                                                                                          \{-4, -12\}, \{-2, -10\}, \{-2, -8\}, \{-3, -6\}, \{-5, -5\}, \{-7, -5\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-9, -7\}, \{-
2199:
                                                                                           \{-9, -9\}, \{-8, -11\}, \{-6, -12\}, \{-4, -12\}, \{-2, -11\}, \{1, -10\}, \{4, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{
2200:
                                                                                           \{7, -11\}, \{9, -12\},\
2201: };
2202: static XPoint seg2 2271[] = {
2203:
                                                                                           \{5,2\},\{3,3\},\{2,5\},\{2,7\},\{4,9\},\{6,9\},\{8,8\},\{9,6\},
2204:
                                                                                          \{9,4\},\{7,2\},\{5,2\},
2205: };
2206: static XPoint *char2271[] = {
2207:
                                                                                           seg0 2271, seg1 2271, seg2 2271,
2208:
                                                                                          NULL,
2209: };
2210: static int char p2271[] = {
2211:
                                                                                          XtNumber(seg0 2271), XtNumber(seg1 2271), XtNumber(seg2 2271),
2212: };
2213: static XPoint seg0_2273[] = {
2214:
                                                                                          \{5, -4\}, \{4, -6\}, \{2, -7\}, \{-1, -7\}, \{-3, -6\}, \{-4, -5\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2
2215:
                                                                                           \{-5,1\},\{-4,3\},\{-2,4\},\{1,4\},\{3,3\},\{4,1\},
2216: };
2217: static XPoint seg1 2273[] = {
2218:
                                                                                           \{-1, -7\}, \{-3, -5\}, \{-4, -2\}, \{-4, 1\}, \{-3, 3\}, \{-2, 4\},
2219: };
```

```
2220: static XPoint seg2 2273[] = {
2221:
                                                                                                         \{5, -7\}, \{4, 1\}, \{4, 3\}, \{6, 4\}, \{8, 4\}, \{10, 2\}, \{11, -1\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11, -3\}, \{11,
2222:
                                                                                                           \{10, -6\}, \{9, -8\}, \{7, -10\}, \{5, -11\}, \{2, -12\}, \{-1, -12\}, \{-4, -11\}, \{-1, -12\}, \{-4, -11\}, \{-1, -12\}, \{-4, -11\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1,
2223:
                                                                                                           \{-6, -10\}, \{-8, -8\}, \{-9, -6\}, \{-10, -3\}, \{-10, 0\}, \{-9, 3\}, \{-8, 5\},
2224:
                                                                                                           \{-6,7\},\{-4,8\},\{-1,9\},\{2,9\},\{5,8\},\{7,7\},\{8,6\},
2225: };
2226: static XPoint seg3 2273[] = {
2227:
                                                                                                         \{6, -7\}, \{5, 1\}, \{5, 3\}, \{6, 4\},\
2228: };
2229: static XPoint *char2273[] = {
2230:
                                                                                                         seg0 2273, seg1 2273, seg2 2273, seg3 2273,
2231:
                                                                                                         NULL,
2232: };
2233: static int char p2273[] = {
2234:
                                                                                                         XtNumber(seg0_2273), XtNumber(seg1_2273), XtNumber(seg2_2273),
2235:
                                                                                                         XtNumber(seg3 2273),
2236: };
2237: static XPoint seg0 2275[] = {
2238:
                                                                                                         \{1, -12\}, \{-6, 16\},\
2239: };
2240: static XPoint seg1 2275[] = {
2241:
                                                                                                         \{7, -12\}, \{0, 16\},\
2242: };
2243: static XPoint seg2 2275[] = {
2244:
                                                                                                       \{-6, -1\}, \{8, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, 
2245: };
2246: static XPoint seg3 2275[] = {
2247:
                                                                                                           \{-7,5\},\{7,5\},
2248: };
2249: static XPoint *char2275[] = {
2250:
                                                                                                         seg0 2275, seg1 2275, seg2 2275, seg3 2275,
2251:
                                                                                                         NULL,
2252: };
2253: static int char p2275[] = {
2254:
                                                                                                         XtNumber(seg0 2275), XtNumber(seg1 2275), XtNumber(seg2 2275),
2255:
                                                                                                         XtNumber(seg3_2275),
2256: };
2257: static XPoint seg0_2750[] = {
2258:
                                                                                                           \{2, -12\}, \{-1, -11\}, \{-3, -9\}, \{-5, -6\}, \{-6, -3\}, \{-7, 1\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7, 4\}, \{-7,
2259:
                                                                                                           \{-6,7\},\{-5,8\},\{-3,9\},\{-1,9\},\{2,8\},\{4,6\},\{6,3\},\{7,0\},\{8,-4\},
2260:
                                                                                                           \{8, -7\}, \{7, -10\}, \{6, -11\}, \{4, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{3, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -
2261: };
2262: static XPoint seg1 2750[] = {
2263:
                                                                                                           \{2, -12\}, \{0, -11\}, \{-2, -9\}, \{-4, -6\}, \{-5, -3\}, \{-6, 1\}, \{-6, 4\}, \{-5, 7\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1\}, \{-6, -1
2264:
                                                                                                           \{-3,9\},\
2265: };
2266: static XPoint seg2_2750[] = {
2267:
                                                                                                         \{-1,9\},\{1,8\},\{3,6\},\{5,3\},\{6,0\},\{7,-4\},\{7,-7\},\{6,-10\},
2268:
                                                                                                           \{4, -12\},\
2269: }:
```

```
2270: static XPoint *char2750[] = {
2271:
                                                                         seg0_2750, seg1_2750, seg2_2750,
                                                                        NULL,
2272:
2273: };
2274: static int char_p2750[] = {
                                                                        XtNumber(seg0 2750), XtNumber(seg1 2750), XtNumber(seg2 2750),
2275:
2276: };
2277: static XPoint seg0 2751[] = {
2278:
                                                                         \{2, -8\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{
2279: };
2280: static XPoint seg1 2751[] = {
2281:
                                                                        \{4, -12\}, \{-2, 9\}, 
2282: };
2283: static XPoint seg2 2751[] = {
2284:
                                                                         \{4, -12\}, \{1, -9\}, \{-2, -7\}, \{-4, -6\},
2285: };
2286: static XPoint seg3 2751[] = {
2287:
                                                                        \{3, -9\}, \{-1, -7\}, \{-4, -6\},\
2288: };
2289: static XPoint *char2751[] = {
2290:
                                                                         seg0_2751, seg1_2751, seg2_2751, seg3_2751,
2291:
                                                                        NULL,
2292: };
2293: static int char p2751[] = {
2294:
                                                                        XtNumber(seg0 2751), XtNumber(seg1 2751), XtNumber(seg2 2751),
2295:
                                                                        XtNumber(seg3 2751),
2296: };
2297: static XPoint seg0 2752[] = {
2298:
                                                                        \{-3, -8\}, \{-2, -7\}, \{-3, -6\}, \{-4, -7\}, \{-4, -8\}, \{-3, -10\}, \{-2, -11\},
2299:
                                                                         \{1, -12\}, \{4, -12\}, \{7, -11\}, \{8, -9\}, \{8, -7\}, \{7, -5\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}
2300:
                                                                         \{2, -1\}, \{-2, 1\}, \{-5, 3\}, \{-7, 5\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{
2301: };
2302: static XPoint seg1 2752[] = {
2303:
                                                                        \{4, -12\}, \{6, -11\}, \{7, -9\}, \{7, -7\}, \{6, -5\}, \{4, -3\}, \{-2, 1\},
2304: };
2305: static XPoint seg2 2752[] = {
2306:
                                                                         \{-8,7\},\{-7,6\},\{-5,6\},\{0,8\},\{3,8\},\{5,7\},\{6,5\},
2307: };
2308: static XPoint seg3 2752[] = {
2309:
                                                                         \{-5,6\},\{0,9\},\{3,9\},\{5,8\},\{6,5\},
2310: };
2311: static XPoint *char2752[] = {
2312:
                                                                         seg0 2752, seg1 2752, seg2 2752, seg3 2752,
2313:
                                                                        NULL,
2314: };
2315: static int char p2752[] = {
2316:
                                                                        XtNumber(seg0_2752),XtNumber(seg1_2752),XtNumber(seg2_2752),
2317:
                                                                        XtNumber(seg3 2752),
2318: };
2319: static XPoint seg0 2753[] = {
2320:
                                                                         \{-3, -8\}, \{-2, -7\}, \{-3, -6\}, \{-4, -7\}, \{-4, -8\}, \{-3, -10\}, \{-2, -11\},
2321:
                                                                         \{1, -12\}, \{4, -12\}, \{7, -11\}, \{8, -9\}, \{8, -7\}, \{7, -5\}, \{4, -3\}, \{1, -2\}, \{4, -3\}, \{1, -2\}, \{4, -3\}, \{1, -2\}, \{4, -3\}, \{1, -2\}, \{4, -3\}, \{1, -2\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}
2322: };
```

```
2323: static XPoint seg1_2753[] = {
2324:
                                                                 \{4, -12\}, \{6, -11\}, \{7, -9\}, \{7, -7\}, \{6, -5\}, \{4, -3\},
2325: };
2326: static XPoint seg2_2753[] = {
2327:
                                                                  \{-1, -2\}, \{1, -2\}, \{4, -1\}, \{5, 0\}, \{6, 2\}, \{6, 5\}, \{5, 7\}, \{4, 8\},
2328:
                                                                  \{1,9\},\{-3,9\},\{-6,8\},\{-7,7\},\{-8,5\},\{-8,4\},\{-7,3\},\{-6,4\},\{-7,5\},
2329: };
2330: static XPoint seg3 2753[] = {
                                                                 \{1, -2\}, \{3, -1\}, \{4, 0\}, \{5, 2\}, \{5, 5\}, \{4, 7\}, \{3, 8\}, \{1, 9\},
2331:
2332: \};
2333: static XPoint *char2753[] = {
2334:
                                                                  seg0 2753, seg1 2753, seg2 2753, seg3 2753,
                                                                 NULL,
2335:
2336: };
2337: static int char p2753[] = {
2338:
                                                                 XtNumber(seg0 2753), XtNumber(seg1 2753), XtNumber(seg2 2753),
2339:
                                                                 XtNumber(seg3 2753),
2340: };
2341: static XPoint seg0_2754[] = {
2342:
                                                                 \{6, -11\}, \{0, 9\},\
2343: };
2344: static XPoint seg1 2754[] = {
2345:
                                                                 \{7, -12\}, \{1, 9\},\
2346: };
2347: static XPoint seg2 2754[] = {
2348:
                                                                 \{7, -12\}, \{-8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{
2349: };
2350: static XPoint *char2754[] = {
2351:
                                                                 seg0 2754, seg1 2754, seg2 2754,
2352:
                                                                 NULL,
2353: \};
2354: static int char p2754[] = {
                                                                 XtNumber(seg0 2754), XtNumber(seg1 2754), XtNumber(seg2 2754),
2355:
2356: };
2357: static XPoint seg0 2755[] = {
2358:
                                                                 \{-1, -12\}, \{-6, -2\},\
2359: };
2360: static XPoint seg1 2755[] = {
2361:
                                                               \{-1, -12\}, \{9, -12\}, 
2362: };
2363: static XPoint seg2 2755[] = {
2364:
                                                                  \{-1, -11\}, \{4, -11\}, \{9, -12\}, \{9, -12\}, \{1, -11\}, \{1, -11\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{3, -12\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4, -11\}, \{4,
2365: };
2366: static XPoint seg3 2755[] = {
2367:
                                                                  \{-6, -2\}, \{-5, -3\}, \{-2, -4\}, \{1, -4\}, \{4, -3\}, \{5, -2\}, \{6, 0\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6,
2368:
                                                                  \{5,6\},\{3,8\},\{0,9\},\{-3,9\},\{-6,8\},\{-7,7\},\{-8,5\},\{-8,4\},\{-7,3\},
2369:
                                                                  \{-6,4\},\{-7,5\},\
2370: };
```

```
2371: static XPoint seg4 2755[] = {
2372:
                                                                                                                            \{1, -4\}, \{3, -3\}, \{4, -2\}, \{5, 0\}, \{5, 3\}, \{4, 6\}, \{2, 8\}, \{0, 9\}, \{1, -4\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{3, -2\}, \{4, -2\}, \{3, -2\}, \{4, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2\}, \{5, -2
2373: };
2374: static XPoint *char2755[] = {
2375:
                                                                                                                            seg0_2755, seg1_2755, seg2_2755, seg3_2755, seg4_2755,
2376:
                                                                                                                          NULL,
2377: };
2378: static int char_p2755[] = {
2379:
                                                                                                                            XtNumber(seg0 2755), XtNumber(seg1 2755), XtNumber(seg2 2755),
2380:
                                                                                                                          XtNumber(seg3 2755), XtNumber(seg4 2755),
2381: };
2382: static XPoint seg0_2756[] = {
2383:
                                                                                                                          \{7, -9\}, \{6, -8\}, \{7, -7\}, \{8, -8\}, \{8, -9\}, \{7, -11\}, \{5, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{3, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\},
2384:
                                                                                                                            \{-1, -11\}, \{-3, -9\}, \{-5, -6\}, \{-6, -3\}, \{-7, 1\}, \{-7, 5\}, \{-6, 7\}, \{-5, 8\}, \{-1, -11\}, \{-7, 5\}, \{-6, 7\}, \{-5, 8\}, \{-1, -11\}, \{-7, 5\}, \{-6, 7\}, \{-1, -11\}, \{-7, 5\}, \{-6, 7\}, \{-1, -11\}, \{-7, 5\}, \{-7, 1\}, \{-7, 5\}, \{-7, 1\}, \{-7, 5\}, \{-7, 1\}, \{-7, 5\}, \{-7, 1\}, \{-7, 5\}, \{-7, 1\}, \{-7, 5\}, \{-7, 1\}, \{-7, 5\}, \{-7, 1\}, \{-7, 5\}, \{-7, 1\}, \{-7, 5\}, \{-7, 1\}, \{-7, 5\}, \{-7, 1\}, \{-7, 5\}, \{-7, 1\}, \{-7, 5\}, \{-7, 1\}, \{-7, 5\}, \{-7, 1\}, \{-7, 5\}, \{-7, 1\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5\}, \{-7, 5
2385:
                                                                                                                            \{-3,9\},\{0,9\},\{3,8\},\{5,6\},\{6,4\},\{6,1\},\{5,-1\},\{4,-2\},\{2,-3\},
2386:
                                                                                                                            \{-1, -3\}, \{-3, -2\}, \{-5, 0\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\}, \{-6, 2\},
2387: };
2388: static XPoint seg1 2756[] = {
2389:
                                                                                                                            \{2, -12\}, \{0, -11\}, \{-2, -9\}, \{-4, -6\}, \{-5, -3\}, \{-6, 1\}, \{-6, 6\}, \{-5, 8\}, 
2390: };
2391: static XPoint seg2 2756[] = {
2392:
                                                                                                                          \{0,9\},\{2,8\},\{4,6\},\{5,4\},\{5,0\},\{4,-2\},
2393: };
2394: static XPoint *char2756[] = {
2395:
                                                                                                                          seg0 2756, seg1 2756, seg2 2756,
2396:
                                                                                                                          NULL,
2397: };
2398: static int char p2756[] = {
2399:
                                                                                                                          XtNumber(seg0 2756), XtNumber(seg1 2756), XtNumber(seg2 2756),
2400: };
2401: static XPoint seg0 2757[] = {
2402:
                                                                                                                          \{-4, -12\}, \{-6, -6\},
2403: };
2404: static XPoint seg1 2757[] = {
2405:
                                                                                                                          \{9, -12\}, \{8, -9\}, \{6, -6\}, \{1, 0\}, \{-1, 3\}, \{-2, 5\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{
2406: };
2407: static XPoint seg2 2757[] = {
2408:
                                                                                                                          \{6, -6\}, \{0, 0\}, \{-2, 3\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-3, 5\}, \{-4, 9\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-3, 5\}, \{-
2409: };
2410: static XPoint seg3 2757[] = {
2411:
                                                                                                                          \{-5, -9\}, \{-2, -12\}, \{0, -12\}, \{5, -9\}, \{-5, -9\}, \{-2, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1
2412: };
2413: static XPoint seg4 2757[] = {
2414:
                                                                                                                            \{-4, -10\}, \{-2, -11\}, \{0, -11\}, \{5, -9\}, \{7, -9\}, \{8, -10\}, \{9, -12\}, \{9, -12\}, \{1, -10\}, \{1, -10\}, \{1, -10\}, \{2, -11\}, \{2, -11\}, \{2, -11\}, \{3, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, 
2415: };
2416: static XPoint *char2757[] = {
2417:
                                                                                                                            seg0_2757, seg1_2757, seg2_2757, seg3_2757, seg4_2757,
2418:
                                                                                                                          NULL,
2419: };
2420: static int char p2757[] = {
2421:
                                                                                                                            XtNumber(seg0 2757), XtNumber(seg1 2757), XtNumber(seg2 2757),
2422:
                                                                                                                            XtNumber(seg3 2757), XtNumber(seg4 2757),
2423: };
```

```
C
```

```
2424: static XPoint seg0 2758[] = {
2425:
                                                                                                                     \{1, -12\}, \{-2, -11\}, \{-3, -10\}, \{-4, -8\}, \{-4, -5\}, \{-3, -3\}, \{-1, -2\}, \{-3, -3\}, \{-1, -2\}, \{-3, -3\}, \{-1, -2\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-
2426:
                                                                                                                       \{2, -2\}, \{6, -3\}, \{7, -4\}, \{8, -6\}, \{8, -9\}, \{7, -11\}, \{4, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{2, -2\}, \{2, -2\}, \{3, -2\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4
2427: \};
2428: static XPoint seg1 2758[] = {
2429:
                                                                                                                     \{1, -12\}, \{-1, -11\}, \{-2, -10\}, \{-3, -8\}, \{-3, -5\}, \{-2, -3\}, \{-1, -2\}, \{-2, -3\}, \{-1, -2\}, \{-2, -3\}, \{-2, -3\}, \{-3, -2\}, \{-2, -3\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-
2430: };
2431: static XPoint seg2 2758[] = {
2432:
                                                                                                                     \{2, -2\}, \{5, -3\}, \{6, -4\}, \{7, -6\}, \{7, -9\}, \{6, -11\}, \{4, -12\}, \{6, -11\}, \{4, -12\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\},
2433: };
2434: static XPoint seg3 2758[] = {
2435:
                                                                                                                       \{-1, -2\}, \{-5, -1\}, \{-7, 1\}, \{-8, 3\}, \{-8, 6\}, \{-7, 8\}, \{-4, 9\}, \{0, 9\}, \{0, 9\}, \{-1, -2\}, \{-2, -1\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3
2436:
                                                                                                                       \{4,8\},\{5,7\},\{6,5\},\{6,2\},\{5,0\},\{4,-1\},\{2,-2\},
2437: }:
2438: static XPoint seg4 2758[] = {
2439:
                                                                                                                     \{-1, -2\}, \{-4, -1\}, \{-6, 1\}, \{-7, 3\}, \{-7, 6\}, \{-6, 8\}, \{-4, 9\},
2440: };
2441: static XPoint seg5 2758[] = {
2442:
                                                                                                                       \{0,9\},\{3,8\},\{4,7\},\{5,5\},\{5,1\},\{4,-1\},
2443: };
2444: static XPoint *char2758[] = {
2445:
                                                                                                                       seg0 2758, seg1 2758, seg2 2758, seg3 2758, seg4 2758, seg5 2758,
                                                                                                                     NULL,
2446:
2447: };
2448: static int char_p2758[] = {
2449:
                                                                                                                     XtNumber(seg0 2758), XtNumber(seg1 2758), XtNumber(seg2 2758),
2450:
                                                                                                                     XtNumber(seg3 2758), XtNumber(seg4 2758), XtNumber(seg5 2758),
2451: }:
2452: static XPoint seg0 2759[] = {
2453:
                                                                                                                     \{7, -5\}, \{6, -3\}, \{4, -1\}, \{2, 0\}, \{-1, 0\}, \{-3, -1\}, \{-4, -2\}, \{-5, -4\},
2454:
                                                                                                                       \{-5, -7\}, \{-4, -9\}, \{-2, -11\}, \{1, -12\}, \{4, -12\}, \{6, -11\}, \{7, -10\}, \{-5, -7\}, \{-4, -9\}, \{-2, -11\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -
2455:
                                                                                                                     \{8, -8\}, \{8, -4\}, \{7, 0\}, \{6, 3\}, \{4, 6\}, \{2, 8\}, \{-1, 9\}, \{-4, 9\}, \{-6, 8\},
2456:
                                                                                                                       \{-7,6\},\{-7,5\},\{-6,4\},\{-5,5\},\{-6,6\},
2457: \};
2458: static XPoint seg1 2759[] = {
2459:
                                                                                                                     \{-3, -1\}, \{-4, -3\}, \{-4, -7\}, \{-3, -9\}, \{-1, -11\}, \{1, -12\}, \{-3, -1\}, \{-1, -11\}, \{-1, -12\}, \{-3, -1\}, \{-3, -1\}, \{-3, -1\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{
2460: };
2461: static XPoint seg2 2759[] = {
2462:
                                                                                                                     \{6, -11\}, \{7, -9\}, \{7, -4\}, \{6, 0\}, \{5, 3\}, \{3, 6\}, \{1, 8\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1,
2463: };
2464: static XPoint *char2759[] = {
2465:
                                                                                                                       seg0 2759, seg1 2759, seg2 2759,
2466:
                                                                                                                     NULL,
2467: };
2468: static int char p2759[] = {
                                                                                                                     XtNumber(seg0_2759),XtNumber(seg1_2759),XtNumber(seg2_2759),
2469:
2470: };
2471: static XPoint seg0 2761[] = {
2472:
                                                                                                                     \{-2,9\},\{-3,8\},\{-2,7\},\{-1,8\},
2473:
                                                                                                                       \{-1,9\},\{-2,11\},\{-4,13\},
2474: };
```

```
2475: static XPoint *char2761[] = {
2476:
                                                                    seg0_2761,
2477:
                                                                   NULL,
2478: };
2479: static int char_p2761[] = {
                                                                 XtNumber(seg0 2761),
2480:
2481: };
2482: static XPoint seg0 2762[] = {
2483:
                                                                    \{1, -5\}, \{0, -4\}, \{1, -3\}, \{2, -4\}, \{1, -5\}, \{1, -5\}, \{2, -4\}, \{1, -5\}, \{2, -4\}, \{1, -5\}, \{2, -4\}, \{1, -5\}, \{2, -4\}, \{1, -5\}, \{2, -4\}, \{2, -4\}, \{3, -5\}, \{3, -4\}, \{3, -5\}, \{3, -4\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{
2484: };
2485: static XPoint seg1 2762[] = {
2486:
                                                                   \{-2,7\},\{-3,8\},\{-2,9\},\{-1,8\},
2487: };
2488: static XPoint *char2762[] = {
2489:
                                                                   seg0 2762, seg1 2762,
2490:
                                                                   NULL,
2491: };
2492: static int char p2762[] = {
2493:
                                                                   XtNumber(seg0_2762),XtNumber(seg1_2762),
2494: };
2495: static XPoint seg0 2763[] = {
2496:
                                                                   \{1, -5\}, \{0, -4\}, \{1, -3\}, \{2, -4\}, \{1, -3\}, \{2, -4\}, \{2, -4\}, \{2, -4\}, \{2, -4\}, \{2, -4\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{
2497:
                                                                   {1,-5},
2498: };
2499: static XPoint seg1 2763[] = {
2500:
                                                                   \{-2,9\},\{-3,8\},\{-2,7\},\{-1,8\},\{-1,9\},\{-2,11\},\{-4,13\},
2501: };
2502: static XPoint *char2763[] = {
2503:
                                                                   seg0 2763, seg1 2763,
2504:
                                                                   NULL,
2505: };
2506: static int char_p2763[] = {
2507:
                                                                 XtNumber(seg0 2763),XtNumber(seg1 2763),
2508: };
2509: static XPoint seg0_2764[] = {
2510:
                                                                   \{3, -12\},\
2511:
                                                                   \{2, -11\}, \{0, 1\}, 
2512: };
2513: static XPoint seg1 2764[] = {
2514:
                                                                   \{3, -11\}, \{0, 1\},\
2515: };
2516: static XPoint seg2 2764[] = {
2517:
                                                                   \{3, -12\}, \{4, -11\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}, 
2518: };
2519: static XPoint seg3 2764[] = {
2520:
                                                                   \{-2,7\},\{-3,8\},\{-2,9\},\{-1,8\},\{-2,7\},
2521: };
2522: static XPoint *char2764[] = {
2523:
                                                                   seg0_2764, seg1_2764, seg2_2764, seg3_2764,
2524:
                                                                   NULL,
2525: };
2526: static int char p2764[] = {
2527:
                                                                   XtNumber(seg0_2764), XtNumber(seg1_2764), XtNumber(seg2_2764),
```

```
Listing C.4 Continued
```

```
2528:
                                                                                                                XtNumber(seg3_2764),
2529: };
2530: static XPoint seg0_2765[] = {
2531:
                                                                                                                  \{-3, -8\}, \{-2, -7\}, \{-3, -6\}, \{-4, -7\}, \{-4, -8\}, \{-3, -10\}, \{-2, -11\},
2532:
                                                                                                                  \{1, -12\}, \{5, -12\}, \{8, -11\}, \{9, -9\}, \{9, -7\}, \{8, -5\}, \{7, -4\}, \{1, -2\}, \{1, -2\}, \{2, -12\}, \{3, -12\}, \{3, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, 
2533:
                                                                                                                  \{-1, -1\}, \{-1, 1\}, \{0, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{
2534: };
2535: static XPoint seg1_2765[] = {
2536:
                                                                                                                  \{5, -12\}, \{7, -11\}, \{8, -9\}, \{8, -7\}, \{7, -5\}, \{6, -4\}, \{4, -3\}, \{6, -4\}, \{4, -3\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\},
2537: \};
2538: static XPoint seg2 2765[] = {
2539:
                                                                                                                \{-2,7\},\{-3,8\},\{-2,9\},\{-1,8\},\{-2,7\},
2540: };
2541: static XPoint *char2765[] = {
2542:
                                                                                                                  seg0_2765, seg1_2765, seg2_2765,
                                                                                                                NULL,
2543:
2544: };
2545: static int char p2765[] = {
2546:
                                                                                                              XtNumber(seg0_2765),XtNumber(seg1_2765),XtNumber(seg2_2765),
2547: };
2548: static XPoint seg0 2766[] = {
2549:
                                                                                                                \{4, -12\}, \{2, -10\}, \{1, -8\},\
2550:
                                                                                                                  \{1, -7\}, \{2, -6\}, \{3, -7\}, \{2, -8\}, \{2, -8\}, \{2, -8\}, \{2, -8\}, \{2, -8\}, \{3, -7\}, \{2, -8\}, \{3, -7\}, \{2, -8\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{
2551: };
2552: static XPoint *char2766[] = {
2553:
                                                                                                                  seg0 2766,
2554:
                                                                                                                NULL,
2555: };
2556: static int char p2766[] = {
2557:
                                                                                                                XtNumber(seg0_2766),
2558: };
2559: static XPoint seg0 2767[] = {
2560:
                                                                                                                  \{3, -10\}, \{2, -11\}, \{3, -12\}, \{4, -11\}, \{4, -10\},
2561:
                                                                                                                  \{3, -8\}, \{1, -6\},\
2562: };
2563: static XPoint *char2767[] = {
2564:
                                                                                                                  seg0 2767,
2565:
                                                                                                                NULL,
2566: \};
2567: static int char p2767[] = {
2568:
                                                                                                                XtNumber(seg0 2767),
2569: };
2570: static XPoint seg0 2768[] = {
2571:
                                                                                                                  \{10, -4\}, \{9, -3\}, \{10, -2\}, \{11, -3\}, \{11, -4\}, \{10, -5\}, \{9, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -
2572:
                                                                                                                  \{7, -4\}, \{5, -2\}, \{0, 6\}, \{-2, 8\}, \{-4, 9\}, \{-7, 9\}, \{-10, 8\}, \{-11, 6\},
2573:
                                                                                                                  \{-11,4\},\{-10,2\},\{-9,1\},\{-7,0\},\{-2,-2\},\{0,-3\},\{2,-5\},\{3,-7\},
2574:
                                                                                                                  \{3, -9\}, \{2, -11\}, \{0, -12\}, \{-2, -11\}, \{-3, -9\}, \{-3, -6\}, \{-2, 0\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-1, 3\}, \{-2, 3\}, \{-1, 3\}, \{-2, 3\}, \{-2, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3, 3\}, \{-3,
2575:
                                                                                                                  \{1,6\},\{3,8\},\{5,9\},\{7,9\},\{8,7\},\{8,6\},
2576: };
```

```
2577: static XPoint seg1 2768[] = {
                                                                                       \{-7,9\},\{-9,8\},\{-10,6\},\{-10,4\},\{-9,2\},\{-8,1\},\{-2,-2\},
2578:
2579: };
2580: static XPoint seg2 2768[] = {
2581:
                                                                                      \{-3, -6\}, \{-2, -1\}, \{-1, 2\}, \{1, 5\}, \{3, 7\}, \{5, 8\}, \{7, 8\}, \{8, 7\}, \{1, 5\}, \{3, 7\}, \{5, 8\}, \{7, 8\}, \{8, 7\}, \{1, 5\}, \{1, 5\}, \{2, 5\}, \{3, 7\}, \{3, 7\}, \{5, 8\}, \{1, 5\}, \{2, 5\}, \{3, 7\}, \{3, 7\}, \{5, 8\}, \{1, 5\}, \{2, 5\}, \{3, 7\}, \{5, 8\}, \{1, 5\}, \{3, 7\}, \{5, 8\}, \{1, 5\}, \{1, 5\}, \{2, 5\}, \{3, 7\}, \{5, 8\}, \{1, 5\}, \{2, 5\}, \{3, 7\}, \{5, 8\}, \{1, 5\}, \{2, 5\}, \{3, 7\}, \{5, 8\}, \{1, 5\}, \{2, 5\}, \{3, 7\}, \{5, 8\}, \{5, 8\}, \{5, 8\}, \{1, 5\}, \{2, 5\}, \{3, 7\}, \{3, 7\}, \{5, 8\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\}, \{3, 7\},
2582: };
2583: static XPoint *char2768[] = {
2584:
                                                                                      seg0 2768, seg1 2768, seg2 2768,
2585:
                                                                                      NULL,
2586: };
2587: static int char p2768[] = {
2588:
                                                                                      XtNumber(seg0 2768), XtNumber(seg1 2768), XtNumber(seg2 2768),
2589: };
2590: static XPoint seg0 2769[] = {
2591:
                                                                                      \{2, -16\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6, 13\}, \{-6,
2592: };
2593: static XPoint seg1 2769[] = {
2594:
                                                                                      \{7, -16\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1,
2595: };
2596: static XPoint seg2 2769[] = {
2597:
                                                                                       \{8, -8\}, \{7, -7\}, \{8, -6\}, \{9, -7\}, \{9, -8\}, \{8, -10\}, \{7, -11\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\},
2598:
                                                                                       \{0, -12\}, \{-3, -11\}, \{-5, -9\}, \{-5, -7\}, \{-4, -5\}, \{-3, -4\}, \{4, 0\}, \{6, 2\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4\}, \{-3, -4
2599: };
2600: static XPoint seg3 2769[] = {
2601:
                                                                                       \{-5, -7\}, \{-3, -5\}, \{4, -1\}, \{5, 0\}, \{6, 2\}, \{6, 5\}, \{5, 7\}, \{4, 8\},
2602:
                                                                                       \{1,9\},\{-3,9\},\{-6,8\},\{-7,7\},\{-8,5\},\{-8,4\},\{-7,3\},\{-6,4\},\{-7,5\},
2603: };
2604: static XPoint *char2769[] = {
2605:
                                                                                       seg0 2769, seg1 2769, seg2 2769, seg3 2769,
2606:
                                                                                      NULL,
2607: };
2608: static int char_p2769[] = {
2609:
                                                                                      XtNumber(seg0 2769), XtNumber(seg1 2769), XtNumber(seg2 2769),
2610:
                                                                                      XtNumber(seg3 2769),
2611: };
2612: static XPoint seg0 2770[] = {
2613:
                                                                                       \{13, -16\}, \{-13, 16\},\
2614: };
2615: static XPoint *char2770[] = {
2616:
                                                                                       seg0 2770,
2617:
                                                                                      NULL,
2618: };
2619: static int char p2770[] = {
2620:
                                                                                      XtNumber(seg0 2770),
2621: };
2622: static XPoint seg0 2771[] = {
2623:
                                                                                       \{8, -16\}, \{4, -13\}, \{1, -10\}, \{-1, -7\}, \{-3, -3\}, \{-4, 2\}, \{-4, 6\},
2624:
                                                                                       \{-3,11\},\{-2,14\},\{-1,16\},
2625: };
2626: static XPoint seg1 2771[] = {
2627:
                                                                                       \{4, -13\}, \{1, -9\}, \{-1, -5\}, \{-2, -2\}, \{-3, 3\}, \{-3, 8\}, \{-2, 13\}, \{-1, 16\},
2628: };
2629: static XPoint *char2771[] = {
2630:
                                                                                      seg0 2771, seg1 2771,
```

```
2631:
                                                 NULL,
2632: };
2633: static int char p2771[] = {
2634:
                                                XtNumber(seg0_2771), XtNumber(seg1_2771),
2635: };
2636: static XPoint seg0 2772[] = {
2637:
                                                 \{1, -16\}, \{2, -14\}, \{3, -11\}, \{4, -6\}, \{4, -2\}, \{3, 3\}, \{1, 7\},
2638:
                                                  \{-1,10\},\{-4,13\},\{-8,16\},
2639: };
2640: static XPoint seg1 2772[] = {
2641:
                                                \{1, -16\}, \{2, -13\}, \{3, -8\}, \{3, -3\}, \{2, 2\}, \{1, 5\}, \{-1, 9\}, \{-4, 13\}, \{-1, 9\}, \{-4, 13\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\},
2642: };
2643: static XPoint *char2772[] = {
2644:
                                                 seg0_2772, seg1_2772,
2645:
                                                 NULL,
2646: };
2647: static int char p2772[] = {
2648:
                                                XtNumber(seg0 2772), XtNumber(seg1 2772),
2649: };
2650: static XPoint seg0 2773[] = {
2651:
                                                 \{2, -12\}, \{2, 0\},\
2652: };
2653: static XPoint seg1_2773[] = {
2654:
                                                 \{-3, -9\}, \{7, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, 
2655: };
2656: static XPoint seg2 2773[] = {
2657:
                                                 \{7, -9\}, \{-3, -3\},\
2658: };
2659: static XPoint *char2773[] = {
2660:
                                                 seg0_2773, seg1_2773, seg2_2773,
2661:
                                                 NULL,
2662: };
2663: static int char p2773[] = {
                                                 XtNumber(seg0_2773),XtNumber(seg1_2773),XtNumber(seg2_2773),
2664:
2665: };
2666: static XPoint seg0 2778[] = {
2667:
                                                \{-2, -12\}, \{-4, -5\},
2668: };
2669: static XPoint seg1 2778[] = {
2670:
                                                 \{-1, -12\}, \{-4, -5\},
2671: };
2672: static XPoint seg2 2778[] = {
2673:
                                                 \{7, -12\}, \{5, -5\},\
2674: };
2675: static XPoint seg3 2778[] = {
2676:
                                                 \{8, -12\}, \{5, -5\},\
2677: };
2678: static XPoint *char2778[] = {
2679:
                                                  seg0_2778, seg1_2778, seg2_2778, seg3_2778,
2680:
                                                 NULL,
2681: };
```

```
2682: static int char p2778[] = {
2683:
                             XtNumber(seg0_2778),XtNumber(seg1_2778),XtNumber(seg2_2778),
2684:
                             XtNumber(seg3 2778),
2685: };
2686: static XPoint seg0_3001[] = {
2687:
                             \{0, -12\}, \{-7, 8\},\
2688: };
2689: static XPoint seg1 3001[] = {
2690:
                             \{-1, -9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5, 9\}, \{5
2691: };
2692: static XPoint seg2 3001[] = {
2693:
                             \{0, -9\}, \{6, 9\}, 
2694: };
2695: static XPoint seg3 3001[] = {
2696:
                             \{0, -12\}, \{7, 9\},\
2697: };
2698: static XPoint seg4 3001[] = {
2699:
                             \{-5,3\},\{4,3\},
2700: };
2701: static XPoint seg5 3001[] = {
2702:
                             \{-9,9\},\{-3,9\},
2703: };
2704: static XPoint seg6 3001[] = {
2705:
                             \{2,9\},\{9,9\},
2706: };
2707: static XPoint seg7 3001[] = {
2708:
                            \{-7,8\},\{-8,9\},
2709: };
2710: static XPoint seg8 3001[] = {
2711:
                             \{-7,8\},\{-5,9\},
2712: };
2713: static XPoint seg9_3001[] = {
2714:
                             \{5,8\},\{3,9\},\
2715: };
2716: static XPoint seg10_3001[] = {
2717:
                             \{5,7\},\{4,9\},\
2718: };
2719: static XPoint seg11 3001[] = {
2720:
                             \{6,7\},\{8,9\},\
2721: };
2722: static XPoint *char3001[] = {
                             seg0 3001, seg1 3001, seg2 3001, seg3 3001, seg4 3001,
2723:
2724:
                             seg5 3001, seg6 3001, seg7 3001, seg8 3001, seg9 3001,
2725:
                             seg10 3001, seg11 3001,
2726:
                             NULL,
2727: };
2728: static int char_p3001[] = {
                             XtNumber(seg0 3001), XtNumber(seg1 3001), XtNumber(seg2 3001),
2729:
2730:
                             XtNumber(seg3_3001), XtNumber(seg4_3001), XtNumber(seg5_3001),
2731:
                             XtNumber(seg6 3001), XtNumber(seg7 3001), XtNumber(seg8 3001),
                             XtNumber(seg9_3001),XtNumber(seg10_3001),XtNumber(seg11_3001),
2732:
2733: };
```

```
Listing C.4 Continued
```

```
2734: static XPoint seg0 3002[] = {
                                                 \{-6, -12\}, \{-6, 9\},
2735:
2736: };
2737: static XPoint seg1_3002[] = {
2738:
                                                   \{-5, -11\}, \{-5, 8\},\
2739: };
2740: static XPoint seg2 3002[] = {
2741:
                                                   \{-4, -12\}, \{-4, 9\},\
2742: };
2743: static XPoint seg3 3002[] = {
2744:
                                                \{-9, -12\}, \{3, -12\}, \{6, -11\}, \{7, -10\}, \{8, -8\}, \{8, -6\}, \{7, -4\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -
2745:
                                                   \{3, -2\},\
2746: };
2747: static XPoint seq4 3002[] = {
2748:
                                                  \{6, -10\}, \{7, -8\}, \{7, -6\}, \{6, -4\},\
2749: };
2750: static XPoint seg5_3002[] = {
2751:
                                                 \{3, -12\}, \{5, -11\}, \{6, -9\}, \{6, -5\}, \{5, -3\}, \{3, -2\},\
2752: };
2753: static XPoint seg6 3002[] = {
2754:
                                                   \{-4, -2\}, \{3, -2\}, \{6, -1\}, \{7, 0\}, \{8, 2\}, \{8, 5\}, \{7, 7\}, \{6, 8\},
2755:
                                                   \{3,9\},\{-9,9\},\
2756: };
2757: static XPoint seg7 3002[] = {
2758:
                                                 \{6,0\},\{7,2\},\{7,5\},\{6,7\},
2759: };
2760: static XPoint seg8 3002[] = {
2761:
                                                   \{3, -2\}, \{5, -1\}, \{6, 1\}, \{6, 6\}, \{5, 8\}, \{3, 9\}, \{3, 9\}, \{5, 8\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3
2762: };
2763: static XPoint seg9 3002[] = {
                                                   {-8,-12},{-6,-11},
2764:
2765: };
2766: static XPoint seg10 3002[] = {
2767:
                                                   \{-7, -12\}, \{-6, -10\},\
2768: };
2769: static XPoint seg11 3002[] = {
2770:
                                                 \{-3, -12\}, \{-4, -10\},
2771: };
2772: static XPoint seg12 3002[] = {
2773:
                                                   \{-2, -12\}, \{-4, -11\},
2774: };
2775: static XPoint seg13 3002[] = {
2776:
                                                  \{-6,8\},\{-8,9\},
2777: };
2778: static XPoint seg14 3002[] = {
2779:
                                                  \{-6,7\},\{-7,9\},
2780: };
2781: static XPoint seg15 3002[] = {
2782:
                                                   \{-4,7\},\{-3,9\},
2783: };
```

```
2784: static XPoint seg16 3002[] = {
2785:
                                      \{-4,8\},\{-2,9\},
2786: };
2787: static XPoint *char3002[] = {
2788:
                                      seg0_3002, seg1_3002, seg2_3002, seg3_3002, seg4_3002,
2789:
                                      seg5 3002, seg6 3002, seg7 3002, seg8 3002, seg9 3002,
2790:
                                      seg10 3002, seg11 3002, seg12 3002, seg13 3002, seg14 3002,
2791:
                                      seg15 3002, seg16 3002,
2792:
                                      NULL,
2793: };
2794: static int char p3002[] = {
                                      XtNumber(seg0 3002), XtNumber(seg1 3002), XtNumber(seg2 3002),
2795:
2796:
                                      XtNumber(seg3 3002), XtNumber(seg4 3002), XtNumber(seg5 3002),
2797:
                                      XtNumber(seg6 3002), XtNumber(seg7 3002), XtNumber(seg8 3002),
                                      XtNumber(seg9 3002), XtNumber(seg10 3002), XtNumber(seg11 3002),
2798:
2799:
                                      XtNumber(seg12 3002),XtNumber(seg13 3002),XtNumber(seg14 3002),
2800:
                                      XtNumber(seg15_3002),XtNumber(seg16_3002),
2801: };
2802: static XPoint seg0_3003[] = {
2803:
                                      \{6, -9\}, \{7, -12\}, \{7, -6\}, \{6, -9\}, \{4, -11\}, \{2, -12\}, \{-1, -12\}, \{-4, -11\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -1
2804:
                                      \{-6, -9\}, \{-7, -7\}, \{-8, -4\}, \{-8, 1\}, \{-7, 4\}, \{-6, 6\}, \{-4, 8\},
2805:
                                      \{-1,9\},\{2,9\},\{4,8\},\{6,6\},\{7,4\},
2806: };
2807: static XPoint seg1 3003[] = {
2808:
                                      \{-5, -9\}, \{-6, -7\}, \{-7, -4\}, \{-7, 1\}, \{-6, 4\}, \{-5, 6\},
2809: };
2810: static XPoint seg2 3003[] = {
2811:
                                      \{-1, -12\}, \{-3, -11\}, \{-5, -8\}, \{-6, -4\}, \{-6, 1\}, \{-5, 5\}, \{-3, 8\}, \{-1, 9\},
2812: };
2813: static XPoint *char3003[] = {
2814:
                                      seg0 3003, seg1 3003, seg2 3003,
2815:
                                      NULL,
2816: };
2817: static int char p3003[] = {
                                      XtNumber(seg0_3003),XtNumber(seg1_3003),XtNumber(seg2_3003),
2818:
2819: };
2820: static XPoint seg0 3004[] = {
2821:
                                      \{-6, -12\}, \{-6, 9\},
2822: };
2823: static XPoint seg1 3004[] = {
2824:
                                      \{-5, -11\}, \{-5, 8\},
2825: };
2826: static XPoint seg2 3004[] = {
2827:
                                      \{-4, -12\}, \{-4, 9\},
2828: };
2829: static XPoint seg3 3004[] = {
2830:
                                      \{-9, -12\}, \{1, -12\}, \{4, -11\}, \{6, -9\}, \{7, -7\}, \{8, -4\}, \{8, 1\}, \{7, 4\}, \{6, -9\}, \{7, -7\}, \{8, -4\}, \{8, 1\}, \{7, 4\}, \{7, 4\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8, -1\}, \{8
2831:
                                      \{6,6\},\{4,8\},\{1,9\},\{-9,9\},
2832: };
2833: static XPoint seg4 3004[] = {
2834:
                                      \{5, -9\}, \{6, -7\}, \{7, -4\}, \{7, 1\}, \{6, 4\}, \{5, 6\},\
2835: };
```

```
Listing C.4 Continued
```

```
2836: static XPoint seg5 3004[] = {
           \{1, -12\}, \{3, -11\}, \{5, -8\}, \{6, -4\}, \{6, 1\}, \{5, 5\}, \{3, 8\}, \{1, 9\},
2837:
2838: };
2839: static XPoint seg6 3004[] = {
2840:
           \{-8, -12\}, \{-6, -11\}, 
2841: };
2842: static XPoint seg7 3004[] = {
2843:
           \{-7, -12\}, \{-6, -10\},\
2844: };
2845: static XPoint seg8 3004[] = {
2846:
          \{-3, -12\}, \{-4, -10\},
2847: };
2848: static XPoint seg9 3004[] = {
2849:
           \{-2, -12\}, \{-4, -11\},
2850: };
2851: static XPoint seg10_3004[] = {
2852:
           \{-6,8\},\{-8,9\},
2853: };
2854: static XPoint seg11 3004[] = {
2855:
           \{-6,7\},\{-7,9\},
2856: };
2857: static XPoint seg12_3004[] = {
2858:
           \{-4,7\},\{-3,9\},
2859: };
2860: static XPoint seg13_3004[] = {
2861:
           \{-4,8\},\{-2,9\},
2862: };
2863: static XPoint *char3004[] = {
2864:
           seg0 3004, seg1_3004, seg2_3004, seg3_3004, seg4_3004,
2865:
           seg5_3004, seg6_3004, seg7_3004, seg8_3004, seg9_3004,
2866:
           seg10 3004, seg11 3004, seg12 3004, seg13 3004,
2867:
           NULL,
2868: };
2869: static int char_p3004[] = {
2870:
           XtNumber(seg0_3004), XtNumber(seg1_3004), XtNumber(seg2_3004),
2871:
           XtNumber(seg3 3004), XtNumber(seg4 3004), XtNumber(seg5 3004),
2872:
           XtNumber(seg6 3004), XtNumber(seg7 3004), XtNumber(seg8 3004),
2873:
           XtNumber(seg9_3004), XtNumber(seg10_3004), XtNumber(seg11_3004),
2874:
           XtNumber(seg12 3004), XtNumber(seg13 3004),
2875: };
2876: static XPoint seg0 3005[] = {
2877:
           \{-6, -12\}, \{-6, 9\},\
2878: };
2879: static XPoint seg1 3005[] = {
2880:
           \{-5, -11\}, \{-5, 8\},\
2881: };
2882: static XPoint seg2 3005[] = {
2883:
           \{-4, -12\}, \{-4, 9\}, 
2884: };
2885: static XPoint seg3 3005[] = {
2886:
           \{-9, -12\}, \{7, -12\}, \{7, -6\},\
2887: };
```

```
2888: static XPoint seg4 3005[] = {
2889:
                                                        \{-4, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, 
2890: };
2891: static XPoint seg5 3005[] = {
2892:
                                                       \{2, -6\}, \{2, 2\},\
2893: };
2894: static XPoint seg6 3005[] = {
2895:
                                                       \{-9,9\},\{7,9\},\{7,3\},
2896: };
2897: static XPoint seg7 3005[] = {
2898:
                                                       \{-8, -12\}, \{-6, -11\},
2899: };
2900: static XPoint seg8 3005[] = {
2901:
                                                       \{-7, -12\}, \{-6, -10\},\
2902: };
2903: static XPoint seg9 3005[] = {
2904:
                                                      \{-3, -12\}, \{-4, -10\},
2905: };
2906: static XPoint seg10 3005[] = {
2907:
                                                     \{-2, -12\}, \{-4, -11\},
2908: };
2909: static XPoint seg11 3005[] = {
2910:
                                                       \{2, -12\}, \{7, -11\},\
2911: };
2912: static XPoint seg12 3005[] = {
2913:
                                                       \{4, -12\}, \{7, -10\},\
2914: };
2915: static XPoint seg13 3005[] = {
2916:
                                                       \{5, -12\}, \{7, -9\},\
2917: };
2918: static XPoint seg14 3005[] = {
2919:
                                                       \{6, -12\}, \{7, -6\},\
2920: };
2921: static XPoint seg15 3005[] = {
2922:
                                                      \{2, -6\}, \{1, -2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2
2923: };
2924: static XPoint seg16 3005[] = {
2925:
                                                      \{2, -4\}, \{0, -2\}, \{2, 0\},\
2926: };
2927: static XPoint seg17 3005[] = {
2928:
                                                      \{2, -3\}, \{-2, -2\}, \{2, -1\}, 
2929: };
2930: static XPoint seg18 3005[] = {
2931:
                                                       \{-6,8\},\{-8,9\},
2932: };
2933: static XPoint seg19 3005[] = {
2934:
                                                       \{-6,7\},\{-7,9\},
2935: };
2936: static XPoint seg20_3005[] = {
2937:
                                                       \{-4,7\},\{-3,9\},
2938: };
```

```
Listing C.4 Continued
```

```
2939: static XPoint seg21_3005[] = {
2940:
                          \{-4,8\},\{-2,9\},\
2941: };
2942: static XPoint seg22 3005[] = {
2943:
                          \{2,9\},\{7,8\},\
2944: };
2945: static XPoint seg23_3005[] = {
2946:
                          \{4,9\},\{7,7\},\
2947: };
2948: static XPoint seg24 3005[] = {
2949:
                         \{5,9\},\{7,6\},\
2950: };
2951: static XPoint seg25 3005[] = {
2952:
                          \{6,9\},\{7,3\},\
2953: };
2954: static XPoint *char3005[] = {
                          seg0 3005, seg1 3005, seg2 3005, seg3 3005, seg4 3005,
2955:
2956:
                          seg5_3005, seg6_3005, seg7_3005, seg8_3005, seg9_3005,
2957:
                          seg10 3005, seg11 3005, seg12 3005, seg13 3005, seg14 3005,
                          seg15 3005, seg16 3005, seg17 3005, seg18 3005, seg19 3005,
2958:
2959:
                          seg20_3005, seg21_3005, seg22_3005, seg23_3005, seg24_3005,
2960:
                          seg25 3005,
                          NULL,
2961:
2962: };
2963: static int char p3005[] = {
2964:
                          XtNumber(seg0 3005), XtNumber(seg1 3005), XtNumber(seg2 3005),
2965:
                          XtNumber(seg3_3005), XtNumber(seg4_3005), XtNumber(seg5_3005),
2966:
                          XtNumber(seg6_3005), XtNumber(seg7_3005), XtNumber(seg8_3005),
2967:
                         XtNumber(seg9 3005), XtNumber(seg10 3005), XtNumber(seg11 3005),
2968:
                          XtNumber(seg12 3005), XtNumber(seg13 3005), XtNumber(seg14 3005),
2969:
                          XtNumber(seg15 3005),XtNumber(seg16 3005),XtNumber(seg17 3005),
2970:
                          XtNumber(seg18 3005), XtNumber(seg19 3005), XtNumber(seg20 3005),
2971:
                          XtNumber(seg21 3005),XtNumber(seg22 3005),XtNumber(seg23 3005),
2972:
                          XtNumber(seg24 3005), XtNumber(seg25 3005),
2973: };
2974: static XPoint seg0_3006[] = {
2975:
                          \{-6, -12\}, \{-6, 9\},
2976: };
2977: static XPoint seg1 3006[] = {
2978:
                          \{-5, -11\}, \{-5, 8\},\
2979: };
2980: static XPoint seg2 3006[] = {
2981:
                          \{-4, -12\}, \{-4, 9\},
2982: };
2983: static XPoint seg3 3006[] = {
2984:
                         \{-9, -12\}, \{7, -12\}, \{7, -6\},\
2985: };
2986: static XPoint seg4 3006[] = {
                          \{-4, -2\}, \{2, -2\},\
2987:
2988: };
2989: static XPoint seg5 3006[] = {
2990:
                          \{2, -6\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2,
2991: };
```

```
2992: static XPoint seg6 3006[] = {
2993:
                               \{-9,9\},\{-1,9\},
2994: };
2995: static XPoint seg7_3006[] = {
2996:
                               \{-8, -12\}, \{-6, -11\},
2997: };
2998: static XPoint seg8 3006[] = {
2999:
                               \{-7, -12\}, \{-6, -10\},
3000: };
3001: static XPoint seg9 3006[] = {
3002:
                              \{-3, -12\}, \{-4, -10\},
3003: };
3004: static XPoint seg10_3006[] = {
3005:
                               \{-2, -12\}, \{-4, -11\},
3006: };
3007: static XPoint seg11 3006[] = {
3008:
                               \{2, -12\}, \{7, -11\},\
3009: };
3010: static XPoint seg12 3006[] = {
3011:
                               \{4, -12\}, \{7, -10\},\
3012: };
3013: static XPoint seg13 3006[] = {
3014:
                               \{5, -12\}, \{7, -9\},\
3015: };
3016: static XPoint seg14_3006[] = {
3017:
                               \{6, -12\}, \{7, -6\},\
3018: };
3019: static XPoint seg15 3006[] = {
3020:
                               \{2, -6\}, \{1, -2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2, 2\}, \{2
3021: };
3022: static XPoint seg16 3006[] = {
3023:
                              \{2, -4\}, \{0, -2\}, \{2, 0\},\
3024: };
3025: static XPoint seg17 3006[] = {
3026:
                               \{2, -3\}, \{-2, -2\}, \{2, -1\},\
3027: };
3028: static XPoint seg18 3006[] = {
3029:
                               \{-6,8\},\{-8,9\},
3030: };
3031: static XPoint seg19_3006[] = {
3032:
                               \{-6,7\},\{-7,9\},\
3033: };
3034: static XPoint seg20_3006[] = {
3035:
                               \{-4,7\},\{-3,9\},
3036: };
3037: static XPoint seg21 3006[] = {
3038:
                               \{-4,8\},\{-2,9\},\
3039: };
3040: static XPoint *char3006[] = {
                               seg0 3006, seg1 3006, seg2 3006, seg3 3006, seg4 3006,
3041:
3042:
                               seg5 3006, seg6 3006, seg7 3006, seg8 3006, seg9 3006,
                               seg10 3006, seg11 3006, seg12 3006, seg13 3006, seg14 3006,
3043:
```

```
C
```

```
3044.
                                        seg15 3006, seg16 3006, seg17 3006, seg18 3006, seg19 3006,
3045:
                                        seg20 3006, seg21 3006,
3046:
                                       NULL,
3047: };
3048: static int char p3006[] = {
                                       XtNumber(seg0_3006), XtNumber(seg1_3006), XtNumber(seg2_3006),
3049:
3050:
                                        XtNumber(seg3_3006), XtNumber(seg4_3006), XtNumber(seg5_3006),
3051:
                                       XtNumber(seg6_3006), XtNumber(seg7_3006), XtNumber(seg8_3006),
3052:
                                        XtNumber(seg9 3006), XtNumber(seg10 3006), XtNumber(seg11 3006),
3053:
                                       XtNumber(seg12 3006), XtNumber(seg13 3006), XtNumber(seg14 3006),
3054:
                                        XtNumber(seg15_3006), XtNumber(seg16_3006), XtNumber(seg17_3006),
3055:
                                        XtNumber(seg18 3006), XtNumber(seg19 3006), XtNumber(seg20 3006),
3056:
                                       XtNumber(seg21_3006),
3057: };
3058: static XPoint seg0 3007[] = {
3059:
                                        \{6, -9\}, \{7, -12\}, \{7, -6\}, \{6, -9\}, \{4, -11\}, \{2, -12\}, \{-1, -12\}, \{-4, -11\}, \{-1, -12\}, \{-4, -11\}, \{-1, -12\}, \{-4, -11\}, \{-1, -12\}, \{-4, -11\}, \{-1, -12\}, \{-4, -11\}, \{-1, -12\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -11\}, \{-4, -1
3060:
                                        \{-6, -9\}, \{-7, -7\}, \{-8, -4\}, \{-8, 1\}, \{-7, 4\}, \{-6, 6\}, \{-4, 8\}, \{-6, 6\}, \{-4, 8\}, \{-6, 6\}, \{-4, 8\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}, \{-6, 6\}
3061:
                                        \{-1,9\},\{2,9\},\{4,8\},\{6,8\},\{7,9\},\{7,1\},
3062: };
3063: static XPoint seg1 3007[] = {
3064:
                                        \{-5, -9\}, \{-6, -7\}, \{-7, -4\}, \{-7, 1\}, \{-6, 4\}, \{-5, 6\},
3065: };
3066: static XPoint seg2 3007[] = {
3067:
                                        \{-1, -12\}, \{-3, -11\}, \{-5, -8\}, \{-6, -4\}, \{-6, 1\}, \{-5, 5\}, \{-3, 8\}, \{-1, 9\}, 
3068: };
3069: static XPoint seg3 3007[] = {
3070:
                                        \{6,2\},\{6,7\},\
3071: };
3072: static XPoint seg4 3007[] = {
3073:
                                       \{5,1\},\{5,7\},\{4,8\},
3074: };
3075: static XPoint seg5 3007[] = {
3076:
                                        \{2,1\},\{10,1\},\
3077: };
3078: static XPoint seg6 3007[] = {
3079:
                                        \{3,1\},\{5,2\},\
3080: };
3081: static XPoint seg7 3007[] = {
3082:
                                       \{4,1\},\{5,3\},\
3083: };
3084: static XPoint seg8 3007[] = {
3085:
                                        \{8,1\},\{7,3\},\
3086: };
3087: static XPoint seq9 3007[] = {
3088:
                                        \{9,1\},\{7,2\},\
3089: };
3090: static XPoint *char3007[] = {
                                        seg0 3007, seg1 3007, seg2 3007, seg3 3007, seg4 3007, seg5 3007,
3091:
3092:
                                        seg6_3007, seg7_3007, seg8_3007, seg9_3007,
3093:
                                       NULL,
3094: };
```

```
3095: static int char p3007[] = {
3096:
           XtNumber(seg0_3007),XtNumber(seg1_3007),XtNumber(seg2_3007),
3097:
           XtNumber(seg3 3007), XtNumber(seg4 3007), XtNumber(seg5 3007),
3098:
           XtNumber(seg6 3007), XtNumber(seg7 3007), XtNumber(seg8 3007),
3099:
           XtNumber(seg9_3007),
3100: };
3101: static XPoint seg0 3008[] = {
3102:
           \{-7, -12\}, \{-7, 9\},\
3103: };
3104: static XPoint seg1 3008[] = {
3105:
           \{-6, -11\}, \{-6, 8\},\
3106: };
3107: static XPoint seg2 3008[] = {
3108:
          \{-5, -12\}, \{-5, 9\},
3109: };
3110: static XPoint seg3 3008[] = {
3111:
          {5,-12},{5,9},
3112: };
3113: static XPoint seg4_3008[] = {
3114:
          \{6, -11\}, \{6, 8\},\
3115: };
3116: static XPoint seg5 3008[] = {
3117:
           \{7, -12\}, \{7, 9\},\
3118: };
3119: static XPoint seg6 3008[] = {
3120:
           \{-10, -12\}, \{-2, -12\}, 
3121: };
3122: static XPoint seg7 3008[] = {
3123:
           \{2, -12\}, \{10, -12\}, 
3124: };
3125: static XPoint seg8 3008[] = {
3126:
           \{-5, -2\}, \{5, -2\}, 
3127: };
3128: static XPoint seg9 3008[] = {
3129:
          \{-10,9\},\{-2,9\},
3130: };
3131: static XPoint seg10 3008[] = {
3132:
          {2,9},{10,9},
3133: };
3134: static XPoint seg11 3008[] = {
3135:
          \{-9, -12\}, \{-7, -11\},
3136: };
3137: static XPoint seg12 3008[] = {
3138:
           \{-8, -12\}, \{-7, -10\},
3139: };
3140: static XPoint seg13 3008[] = {
3141:
           \{-4, -12\}, \{-5, -10\},
3142: };
3143: static XPoint seg14_3008[] = {
3144:
           \{-3, -12\}, \{-5, -11\},
3145: };
```

```
Listing C.4 Continued
```

```
3146: static XPoint seg15 3008[] = {
3147:
          \{3, -12\}, \{5, -11\},\
3148: };
3149: static XPoint seg16 3008[] = {
3150:
          \{4, -12\}, \{5, -10\},\
3151: };
3152: static XPoint seg17 3008[] = {
3153:
          \{8, -12\}, \{7, -10\},\
3154: };
3155: static XPoint seg18 3008[] = {
3156:
          \{9, -12\}, \{7, -11\},\
3157: };
3158: static XPoint seg19 3008[] = {
3159:
          \{-7,8\},\{-9,9\},
3160: };
3161: static XPoint seg20 3008[] = {
3162:
          \{-7,7\},\{-8,9\},
3163: };
3164: static XPoint seg21 3008[] = {
3165:
          \{-5,7\},\{-4,9\},
3166: };
3167: static XPoint seg22 3008[] = {
3168:
          \{-5,8\},\{-3,9\},
3169: };
3170: static XPoint seg23 3008[] = {
3171:
          \{5,8\},\{3,9\},\
3172: };
3173: static XPoint seg24 3008[] = {
3174:
          \{5,7\},\{4,9\},\
3175: };
3176: static XPoint seg25 3008[] = {
3177:
          \{7,7\},\{8,9\},\
3178: };
3179: static XPoint seg26 3008[] = {
3180:
          \{7,8\},\{9,9\},\
3181: };
3182: static XPoint *char3008[] = {
3183:
           seg0_3008, seg1_3008, seg2_3008, seg3_3008, seg4_3008,
3184:
           seg5 3008, seg6 3008, seg7 3008, seg8 3008, seg9 3008,
3185:
           seg10 3008, seg11 3008, seg12 3008, seg13 3008, seg14 3008,
           seg15 3008, seg16 3008, seg17 3008, seg18 3008, seg19 3008,
3186:
           seg20 3008, seg21 3008, seg22 3008, seg23 3008, seg24 3008,
3187:
3188:
           seg25 3008, seg26 3008,
          NULL,
3189:
3190: };
3191: static int char p3008[] = {
3192:
          XtNumber(seg0_3008), XtNumber(seg1_3008), XtNumber(seg2_3008),
3193:
           XtNumber(seg3 3008), XtNumber(seg4 3008), XtNumber(seg5 3008),
3194:
           XtNumber(seg6 3008), XtNumber(seg7 3008), XtNumber(seg8 3008),
3195:
          XtNumber(seg9_3008), XtNumber(seg10_3008), XtNumber(seg11_3008),
3196:
          XtNumber(seg12_3008), XtNumber(seg13_3008), XtNumber(seg14_3008),
```

```
3197:
                           XtNumber(seg15 3008), XtNumber(seg16 3008), XtNumber(seg17 3008),
3198:
                           XtNumber(seg18_3008), XtNumber(seg19_3008), XtNumber(seg20_3008),
3199:
                           XtNumber(seg21 3008), XtNumber(seg22 3008), XtNumber(seg23 3008),
3200:
                           XtNumber(seg24 3008), XtNumber(seg25 3008), XtNumber(seg26 3008),
3201: };
3202: static XPoint seg0 3009[] = {
3203:
                           \{-1, -12\}, \{-1, 9\},
3204: };
3205: static XPoint seg1 3009[] = {
3206:
                           \{0, -11\}, \{0, 8\},\
3207: };
3208: static XPoint seg2 3009[] = {
3209:
                          \{1, -12\}, \{1, 9\},\
3210: };
3211: static XPoint seg3 3009[] = {
3212:
                           \{-4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4,
3213: };
3214: static XPoint seg4 3009[] = {
3215:
                           \{-4,9\},\{4,9\},
3216: };
3217: static XPoint seg5 3009[] = {
3218:
                           \{-3, -12\}, \{-1, -11\},
3219: };
3220: static XPoint seg6 3009[] = {
3221:
                           \{-2, -12\}, \{-1, -10\}, 
3222: };
3223: static XPoint seg7 3009[] = {
3224:
                           \{2, -12\}, \{1, -10\}, 
3225: };
3226: static XPoint seg8 3009[] = {
3227:
                           \{3, -12\}, \{1, -11\},\
3228: };
3229: static XPoint seg9 3009[] = {
3230:
                           \{-1,8\},\{-3,9\},
3231: };
3232: static XPoint seg10 3009[] = {
3233:
                           \{-1,7\},\{-2,9\},
3234: };
3235: static XPoint seg11 3009[] = {
3236:
                           \{1,7\},\{2,9\},\
3237: };
3238: static XPoint seg12 3009[] = {
3239:
                           \{1,8\},\{3,9\},
3240: };
3241: static XPoint *char3009[] = {
                           seg0 3009, seg1 3009, seg2 3009, seg3 3009, seg4 3009,
3242:
3243:
                           seg5_3009, seg6_3009, seg7_3009, seg8_3009, seg9_3009,
3244:
                           seg10 3009, seg11 3009, seg12 3009,
3245:
                           NULL,
3246: };
3247: static int char p3009[] = {
3248:
                           XtNumber(seg0_3009), XtNumber(seg1_3009), XtNumber(seg2_3009),
3249:
                           XtNumber(seg3_3009), XtNumber(seg4_3009), XtNumber(seg5_3009),
```

```
3250:
                                             XtNumber(seg6 3009), XtNumber(seg7 3009), XtNumber(seg8 3009),
3251:
                                             XtNumber(seg9_3009),XtNumber(seg10_3009),XtNumber(seg11_3009),
3252:
                                             XtNumber(seq12 3009),
3253: };
3254: static XPoint seg0 3010[] = {
3255:
                                             \{1, -12\}, \{1, 5\}, \{0, 8\}, \{-1, 9\}, \{0, 8\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1
3256: };
3257: static XPoint seg1 3010[] = {
3258:
                                             \{2, -11\}, \{2, 5\}, \{1, 8\}, \{2, 5\}, \{1, 8\}, \{2, 5\}, \{1, 8\}, \{2, 5\}, \{1, 8\}, \{2, 5\}, \{1, 8\}, \{2, 5\}, \{2, 5\}, \{2, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3, 5\}, \{3
3259: };
3260: static XPoint seg2_3010[] = {
3261:
                                             \{3, -12\}, \{3, 5\}, \{2, 8\}, \{-1, 9\}, \{-3, 9\}, \{-5, 8\}, \{-6, 6\}, \{-6, 4\},
3262:
                                             \{-5,3\},\{-4,3\},\{-3,4\},\{-3,5\},\{-4,6\},\{-5,6\},
3263: };
3264: static XPoint seg3 3010[] = {
3265:
                                             \{-5,4\},\{-5,5\},\{-4,5\},\{-4,4\},\{-5,4\},
3266: };
3267: static XPoint seg4 3010[] = {
3268:
                                             \{-2, -12\}, \{6, -12\}, 
3269: };
3270: static XPoint seg5 3010[] = {
3271:
                                             \{-1, -12\}, \{1, -11\}, 
3272: };
3273: static XPoint seg6 3010[] = {
3274:
                                             \{0, -12\}, \{1, -10\},\
3275: };
3276: static XPoint seg7 3010[] = {
3277:
                                             \{4, -12\}, \{3, -10\},\
3278: };
3279: static XPoint seg8 3010[] = {
3280:
                                             \{5, -12\}, \{3, -11\},\
3281: };
3282: static XPoint *char3010[] = {
                                             seg0 3010, seg1 3010, seg2 3010, seg3 3010, seg4 3010, seg5 3010,
3283:
3284:
                                             seg6_3010, seg7_3010, seg8_3010,
3285:
                                             NULL,
3286: };
3287: static int char_p3010[] = {
3288:
                                             XtNumber(seg0 3010), XtNumber(seg1 3010), XtNumber(seg2 3010),
3289:
                                             XtNumber(seg3 3010),XtNumber(seg4 3010),XtNumber(seg5 3010),
3290:
                                             XtNumber(seg6_3010),XtNumber(seg7_3010),XtNumber(seg8_3010),
3291: };
3292: static XPoint seg0 3011[] = {
3293:
                                             \{-7, -12\}, \{-7, 9\},\
3294: };
3295: static XPoint seg1_3011[] = {
3296:
                                             \{-6, -11\}, \{-6, 8\},
3297: };
3298: static XPoint seg2 3011[] = {
3299:
                                             \{-5, -12\}, \{-5, 9\},
3300: };
```

```
3301: static XPoint seg3 3011[] = {
3302:
                                 \{6, -11\}, \{-5, 0\},\
3303: };
3304: static XPoint seg4 3011[] = {
3305:
                                \{-2, -2\}, \{5, 9\},\
3306: };
3307: static XPoint seg5 3011[] = {
3308:
                                \{-1, -2\}, \{6, 9\},\
3309: };
3310: static XPoint seg6 3011[] = {
3311:
                                \{-1, -4\}, \{7, 9\},\
3312: };
3313: static XPoint seg7 3011[] = {
3314:
                                \{-10, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12
3315: };
3316: static XPoint seg8 3011[] = {
3317:
                               \{3, -12\}, \{9, -12\},\
3318: };
3319: static XPoint seg9_3011[] = {
3320:
                               \{-10,9\},\{-2,9\},
3321: };
3322: static XPoint seg10 3011[] = {
3323:
                                \{2,9\},\{9,9\},
3324: };
3325: static XPoint seg11 3011[] = {
3326:
                                \{-9, -12\}, \{-7, -11\},
3327: };
3328: static XPoint seg12 3011[] = {
3329:
                                \{-8, -12\}, \{-7, -10\},
3330: };
3331: static XPoint seg13 3011[] = {
3332:
                                \{-4, -12\}, \{-5, -10\},\
3333: };
3334: static XPoint seg14 3011[] = {
3335:
                                \{-3, -12\}, \{-5, -11\},
3336: };
3337: static XPoint seg15_3011[] = {
3338:
                                \{5, -12\}, \{6, -11\},\
3339: };
3340: static XPoint seg16 3011[] = {
3341:
                                \{8, -12\}, \{6, -11\},\
3342: };
3343: static XPoint seg17 3011[] = {
3344:
                                \{-7,8\},\{-9,9\},
3345: };
3346: static XPoint seg18_3011[] = {
3347:
                                \{-7,7\},\{-8,9\},\
3348: };
3349: static XPoint seg19_3011[] = {
3350:
                                \{-5,7\},\{-4,9\},
3351: };
```
```
Listing C.4 Continued
```

```
3352: static XPoint seg20 3011[] = {
3353:
                                        \{-5,8\},\{-3,9\},
3354: };
3355: static XPoint seg21 3011[] = {
3356:
                                         \{5,7\},\{3,9\},\
3357: };
3358: static XPoint seg22 3011[] = {
3359:
                                        \{5,7\},\{8,9\},\
3360: };
3361: static XPoint *char3011[] = {
3362:
                                         seg0_3011, seg1_3011, seg2_3011, seg3_3011, seg4_3011,
3363:
                                         seg5 3011, seg6 3011, seg7 3011, seg8 3011, seg9 3011,
                                         seg10_3011, seg11_3011, seg12_3011, seg13_3011, seg14_3011,
3364:
3365:
                                         seg15_3011, seg16_3011, seg17_3011, seg18_3011, seg19_3011,
3366:
                                         seg20 3011, seg21 3011, seg22 3011,
3367:
                                        NULL.
3368: };
3369: static int char p3011[] = {
3370:
                                         XtNumber(seg0 3011), XtNumber(seg1 3011), XtNumber(seg2 3011),
                                         XtNumber(seg3_3011), XtNumber(seg4_3011), XtNumber(seg5_3011),
3371:
3372:
                                         XtNumber(seg6_3011), XtNumber(seg7_3011), XtNumber(seg8_3011),
3373:
                                         XtNumber(seg9 3011), XtNumber(seg10 3011), XtNumber(seg11 3011),
3374:
                                         XtNumber(seg12_3011), XtNumber(seg13_3011), XtNumber(seg14_3011),
3375:
                                         XtNumber(seg15 3011),XtNumber(seg16 3011),XtNumber(seg17 3011),
                                        XtNumber(seg18_3011),XtNumber(seg19_3011),XtNumber(seg20_3011),
3376:
3377:
                                        XtNumber(seg21 3011), XtNumber(seg22 3011),
3378: };
3379: static XPoint seg0 3012[] = {
3380:
                                        \{-4, -12\}, \{-4, 9\},
3381: };
3382: static XPoint seg1 3012[] = {
3383:
                                        \{-3, -11\}, \{-3, 8\},
3384: };
3385: static XPoint seg2 3012[] = {
3386:
                                        \{-2, -12\}, \{-2, 9\}, 
3387: };
3388: static XPoint seg3 3012[] = {
3389:
                                        \{-7, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1,
3390: };
3391: static XPoint seg4 3012[] = {
3392:
                                         \{-7,9\},\{8,9\},\{8,3\},
3393: };
3394: static XPoint seg5 3012[] = {
3395:
                                        \{-6, -12\}, \{-4, -11\},
3396: };
3397: static XPoint seg6_3012[] = {
3398:
                                         \{-5, -12\}, \{-4, -10\}, \{-5, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}, \{-1, -10\}
3399: };
3400: static XPoint seg7 3012[] = {
3401:
                                         \{-1, -12\}, \{-2, -10\},
3402: };
```

```
3403: static XPoint seg8 3012[] = {
3404:
           \{0, -12\}, \{-2, -11\}, 
3405: };
3406: static XPoint seg9 3012[] = {
3407:
          \{-4,8\},\{-6,9\},
3408: };
3409: static XPoint seg10 3012[] = {
3410:
          \{-4,7\},\{-5,9\},
3411: };
3412: static XPoint seg11_3012[] = {
3413:
          \{-2,7\},\{-1,9\},
3414: };
3415: static XPoint seg12 3012[] = {
3416:
          \{-2,8\},\{0,9\},
3417: };
3418: static XPoint seg13 3012[] = {
3419:
          \{3,9\},\{8,8\},\
3420: };
3421: static XPoint seg14 3012[] = {
3422:
          \{5,9\},\{8,7\},\
3423: };
3424: static XPoint seg15 3012[] = {
3425:
           \{6,9\},\{8,6\},\
3426: };
3427: static XPoint seg16 3012[] = {
3428:
          \{7,9\},\{8,3\},\
3429: };
3430: static XPoint *char3012[] = {
           seg0 3012, seg1 3012, seg2 3012, seg3 3012, seg4 3012,
3431:
           seg5 3012, seg6 3012, seg7 3012, seg8 3012, seg9 3012,
3432:
3433:
           seg10 3012, seg11 3012, seg12 3012, seg13 3012, seg14 3012,
           seg15_3012, seg16_3012,
3434:
3435:
          NULL,
3436: };
3437: static int char_p3012[] = {
3438:
          XtNumber(seg0 3012), XtNumber(seg1 3012), XtNumber(seg2 3012),
          XtNumber(seg3_3012), XtNumber(seg4_3012), XtNumber(seg5_3012),
3439:
          XtNumber(seg6_3012), XtNumber(seg7_3012), XtNumber(seg8_3012),
3440:
3441:
          XtNumber(seg9 3012), XtNumber(seg10 3012), XtNumber(seg11 3012),
3442:
          XtNumber(seg12 3012), XtNumber(seg13 3012), XtNumber(seg14 3012),
          XtNumber(seg15 3012), XtNumber(seg16 3012),
3443:
3444: };
3445: static XPoint seg0 3013[] = {
3446:
           \{-8, -12\}, \{-8, 8\},
3447: };
3448: static XPoint seg1 3013[] = {
3449:
           \{-8, -12\}, \{-1, 9\},\
3450: };
3451: static XPoint seg2_3013[] = {
3452:
          \{-7, -12\}, \{-1, 6\},
3453: };
3454: static XPoint seg3 3013[] = {
3455:
           \{-6, -12\}, \{0, 6\},\
3456: };
```

```
Listing C.4 Continued
```

```
3457: static XPoint seg4 3013[] = {
3458:
           \{6, -12\}, \{-1, 9\},\
3459: };
3460: static XPoint seg5_3013[] = {
3461:
           \{6, -12\}, \{6, 9\},\
3462: };
3463: static XPoint seg6 3013[] = {
3464:
           \{7, -11\}, \{7, 8\},\
3465: };
3466: static XPoint seg7_3013[] = {
3467:
          {8,-12},{8,9},
3468: };
3469: static XPoint seg8 3013[] = {
3470:
           \{-11, -12\}, \{-6, -12\}, 
3471: };
3472: static XPoint seg9 3013[] = {
3473:
           \{6, -12\}, \{11, -12\}, 
3474: };
3475: static XPoint seg10_3013[] = {
3476:
           \{-11,9\},\{-5,9\},
3477: };
3478: static XPoint seg11_3013[] = {
3479:
           \{3,9\},\{11,9\},\
3480: };
3481: static XPoint seg12_3013[] = {
3482:
           \{-10, -12\}, \{-8, -11\},
3483: };
3484: static XPoint seg13 3013[] = {
3485:
           \{9, -12\}, \{8, -10\},\
3486: };
3487: static XPoint seg14 3013[] = {
3488:
           \{10, -12\}, \{8, -11\},\
3489: };
3490: static XPoint seg15_3013[] = {
3491:
           \{-8,8\},\{-10,9\},
3492: };
3493: static XPoint seg16_3013[] = {
3494:
           \{-8,8\},\{-6,9\},\
3495: };
3496: static XPoint seg17_3013[] = {
3497:
           \{6,8\},\{4,9\},\
3498: };
3499: static XPoint seg18_3013[] = {
3500:
           \{6,7\},\{5,9\},\
3501: };
3502: static XPoint seg19 3013[] = {
3503:
           \{8,7\},\{9,9\},\
3504: };
```

```
3505: static XPoint seg20 3013[] = {
           {8,8},{10,9},
3506:
3507: };
3508: static XPoint *char3013[] = {
3509:
           seg0_3013, seg1_3013, seg2_3013, seg3_3013, seg4_3013,
3510:
           seg5 3013, seg6 3013, seg7 3013, seg8 3013, seg9 3013,
3511:
           seg10 3013, seg11 3013, seg12 3013, seg13 3013, seg14 3013,
3512:
           seg15 3013, seg16 3013, seg17 3013, seg18 3013, seg19 3013,
3513:
           seg20 3013,
3514:
          NULL,
3515: };
3516: static int char p3013[] = {
3517:
          XtNumber(seg0 3013), XtNumber(seg1 3013), XtNumber(seg2 3013),
3518:
          XtNumber(seg3 3013), XtNumber(seg4 3013), XtNumber(seg5 3013),
           XtNumber(seg6 3013), XtNumber(seg7 3013), XtNumber(seg8 3013),
3519:
3520:
          XtNumber(seg9 3013), XtNumber(seg10 3013), XtNumber(seg11 3013),
3521:
          XtNumber(seg12_3013), XtNumber(seg13_3013), XtNumber(seg14_3013),
3522:
          XtNumber(seg15 3013), XtNumber(seg16 3013), XtNumber(seg17 3013),
3523:
          XtNumber(seg18 3013), XtNumber(seg19 3013), XtNumber(seg20 3013),
3524: };
3525: static XPoint seg0 3014[] = {
3526:
           \{-7, -12\}, \{-7, 8\},\
3527: };
3528: static XPoint seg1 3014[] = {
3529:
           \{-7, -12\}, \{7, 9\},\
3530: };
3531: static XPoint seg2 3014[] = {
3532:
          \{-6, -12\}, \{6, 6\}, 
3533: };
3534: static XPoint seg3 3014[] = {
3535:
          \{-5, -12\}, \{7, 6\},\
3536: };
3537: static XPoint seg4 3014[] = {
3538:
          \{7, -11\}, \{7, 9\},\
3539: };
3540: static XPoint seq5 3014[] = {
3541:
           \{-10, -12\}, \{-5, -12\}, 
3542: };
3543: static XPoint seg6 3014[] = {
3544:
           \{4, -12\}, \{10, -12\},\
3545: };
3546: static XPoint seg7 3014[] = {
3547:
           \{-10,9\},\{-4,9\},
3548: };
3549: static XPoint seg8 3014[] = {
3550:
           \{-9, -12\}, \{-7, -11\},
3551: };
3552: static XPoint seg9 3014[] = {
3553:
          \{5, -12\}, \{7, -11\},\
3554: };
3555: static XPoint seg10 3014[] = {
3556:
           \{9, -12\}, \{7, -11\},\
3557: };
```

```
Listing C.4 Continued
```

```
3558: static XPoint seg11 3014[] = {
3559:
           \{-7,8\},\{-9,9\},\
3560: };
3561: static XPoint seg12 3014[] = {
3562:
           \{-7,8\},\{-5,9\},
3563: };
3564: static XPoint *char3014[] = {
3565:
           seg0_3014, seg1_3014, seg2_3014, seg3_3014, seg4_3014,
3566:
           seg5 3014, seg6 3014, seg7 3014, seg8 3014, seg9 3014,
3567:
           seg10 3014, seg11 3014, seg12 3014,
3568:
           NULL,
3569: };
3570: static int char p3014[] = {
3571:
           XtNumber(seg0_3014), XtNumber(seg1_3014), XtNumber(seg2_3014),
3572:
           XtNumber(seg3 3014),XtNumber(seg4 3014), XtNumber(seg5 3014),
3573:
           XtNumber(seg6_3014), XtNumber(seg7_3014), XtNumber(seg8_3014),
3574:
           XtNumber(seg9 3014), XtNumber(seg10 3014), XtNumber(seg11 3014),
3575:
           XtNumber(seq12 3014),
3576: };
3577: static XPoint seg0 3015[] = {
3578:
           \{-1, -12\}, \{-4, -11\}, \{-6, -9\}, \{-7, -7\}, \{-8, -3\}, \{-8, 0\},
3579:
           \{-7,4\},\{-6,6\},\{-4,8\},\{-1,9\},\{1,9\},\{4,8\},\{6,6\},\{7,4\},\{8,0\},
3580:
           \{8, -3\}, \{7, -7\}, \{6, -9\}, \{4, -11\}, \{1, -12\}, \{-1, -12\}, 
3581: };
3582: static XPoint seg1_3015[] = {
3583:
           \{-5, -9\}, \{-6, -7\}, \{-7, -4\}, \{-7, 1\}, \{-6, 4\}, \{-5, 6\},
3584: };
3585: static XPoint seg2 3015[] = {
3586:
           \{5,6\},\{6,4\},\{7,1\},\{7,-4\},\{6,-7\},\{5,-9\},
3587: };
3588: static XPoint seg3 3015[] = {
3589:
           \{-1, -12\}, \{-3, -11\}, \{-5, -8\}, \{-6, -4\}, \{-6, 1\}, \{-5, 5\}, \{-3, 8\}, \{-1, 9\},
3590: };
3591: static XPoint seg4 3015[] = {
3592:
           \{1,9\},\{3,8\},\{5,5\},\{6,1\},\{6,-4\},\{5,-8\},\{3,-11\},\{1,-12\},
3593: };
3594: static XPoint *char3015[] = {
3595:
           seg0_3015, seg1_3015, seg2_3015, seg3_3015, seg4_3015,
3596:
           NULL,
3597: };
3598: static int char_p3015[] = {
3599:
           XtNumber(seg0 3015), XtNumber(seg1 3015), XtNumber(seg2 3015),
3600:
           XtNumber(seg3_3015),XtNumber(seg4_3015),
3601: };
3602: static XPoint seg0 3016[] = {
3603:
           \{-6, -12\}, \{-6, 9\},\
3604: };
3605: static XPoint seg1 3016[] = {
3606:
           \{-5, -11\}, \{-5, 8\},\
3607: };
```

```
3608: static XPoint seg2 3016[] = {
3609:
                                          \{-4, -12\}, \{-4, 9\},
3610: };
3611: static XPoint seg3 3016[] = {
3612:
                                          \{-9, -12\}, \{3, -12\}, \{6, -11\}, \{7, -10\}, \{8, -8\}, \{8, -5\}, \{7, -3\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -
3613:
                                          \{3, -1\}, \{-4, -1\}, 
3614: };
3615: static XPoint seg4 3016[] = {
3616:
                                          \{6, -10\}, \{7, -8\}, \{7, -5\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, 
3617: };
3618: static XPoint seg5 3016[] = {
3619:
                                          \{3, -12\}, \{5, -11\}, \{6, -9\}, \{6, -4\}, \{5, -2\}, \{3, -1\},\
3620: };
3621: static XPoint seg6 3016[] = {
3622:
                                          \{-9,9\},\{-1,9\},
3623: };
3624: static XPoint seg7_3016[] = {
3625:
                                          \{-8, -12\}, \{-6, -11\},
3626: };
3627: static XPoint seg8 3016[] = {
3628:
                                          \{-7, -12\}, \{-6, -10\},\
3629: };
3630: static XPoint seg9_3016[] = {
3631:
                                          \{-3, -12\}, \{-4, -10\},
3632: };
3633: static XPoint seg10 3016[] = {
                                          \{-2, -12\}, \{-4, -11\},
3634:
3635: };
3636: static XPoint seg11 3016[] = {
3637:
                                          \{-6,8\},\{-8,9\},
3638: };
3639: static XPoint seg12_3016[] = {
3640:
                                          \{-6,7\},\{-7,9\},\
3641: };
3642: static XPoint seg13_3016[] = {
3643:
                                          \{-4,7\},\{-3,9\},
3644: };
3645: static XPoint seg14 3016[] = {
3646:
                                          \{-4,8\},\{-2,9\},\
3647: };
3648: static XPoint *char3016[] = {
                                          seg0 3016, seg1 3016, seg2 3016, seg3 3016, seg4 3016,
3649:
3650:
                                          seg5 3016, seg6 3016, seg7 3016, seg8 3016, seg9 3016,
3651:
                                          seg10 3016, seg11 3016, seg12 3016, seg13 3016, seg14 3016,
3652:
                                          NULL,
3653: };
3654: static int char_p3016[] = {
                                          XtNumber(seg0_3016), XtNumber(seg1_3016), XtNumber(seg2_3016),
3655:
3656:
                                          XtNumber(seg3_3016), XtNumber(seg4_3016), XtNumber(seg5_3016),
3657:
                                          XtNumber(seg6 3016), XtNumber(seg7 3016), XtNumber(seg8 3016),
3658:
                                          XtNumber(seg9 3016), XtNumber(seg10 3016), XtNumber(seg11 3016),
3659:
                                          XtNumber(seg12 3016), XtNumber(seg13 3016), XtNumber(seg14 3016),
3660: };
```

```
3661: static XPoint seg0 3017[] = {
3662:
                                          \{-1, -12\}, \{-4, -11\}, \{-6, -9\}, \{-7, -7\}, \{-8, -3\}, \{-8, 0\},
3663:
                                          \{-7,4\},\{-6,6\},\{-4,8\},\{-1,9\},\{1,9\},\{4,8\},\{6,6\},\{7,4\},\{8,0\},
3664:
                                          \{8, -3\}, \{7, -7\}, \{6, -9\}, \{4, -11\}, \{1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -
3665: };
3666: static XPoint seq1 3017[] = {
3667:
                                          \{-5, -9\}, \{-6, -7\}, \{-7, -4\}, \{-7, 1\}, \{-6, 4\}, \{-5, 6\},
3668: };
3669: static XPoint seg2 3017[] = {
3670:
                                          \{5,6\},\{6,4\},\{7,1\},\{7,-4\},\{6,-7\},\{5,-9\},
3671: };
3672: static XPoint seg3 3017[] = {
3673:
                                         \{-1, -12\}, \{-3, -11\}, \{-5, -8\}, \{-6, -4\}, \{-6, 1\}, \{-5, 5\}, \{-3, 8\}, \{-1, 9\},
3674: };
3675: static XPoint seq4 3017[] = {
3676:
                                         \{1,9\},\{3,8\},\{5,5\},\{6,1\},\{6,-4\},\{5,-8\},\{3,-11\},\{1,-12\},
3677: };
3678: static XPoint seg5 3017[] = {
3679:
                                          \{-4,6\},\{-3,4\},\{-1,3\},\{0,3\},\{2,4\},\{3,6\},\{4,12\},\{5,14\},
3680:
                                          \{7, 14\}, \{8, 12\}, \{8, 10\}, 
3681: };
3682: static XPoint seg6 3017[] = {
3683:
                                          \{4,10\},\{5,12\},\{6,13\},\{7,13\},
3684: };
3685: static XPoint seg7 3017[] = {
                                         \{3,6\},\{5,11\},\{6,12\},\{7,12\},\{8,11\},
3686:
3687: };
3688: static XPoint *char3017[] = {
3689:
                                          seg0 3017, seg1 3017, seg2 3017, seg3 3017, seg4 3017, seg5 3017,
3690:
                                          seg6 3017, seg7 3017,
3691:
                                         NULL,
3692: };
3693: static int char p3017[] = {
3694:
                                         XtNumber(seg0 3017), XtNumber(seg1 3017), XtNumber(seg2 3017),
3695:
                                          XtNumber(seg3 3017), XtNumber(seg4 3017), XtNumber(seg5 3017),
3696:
                                         XtNumber(seq6 3017), XtNumber(seq7 3017),
3697: };
3698: static XPoint seg0 3018[] = {
3699:
                                          \{-6, -12\}, \{-6, 9\},
3700: };
3701: static XPoint seg1 3018[] = {
3702:
                                          \{-5, -11\}, \{-5, 8\},\
3703: };
3704: static XPoint seg2 3018[] = {
3705:
                                          \{-4, -12\}, \{-4, 9\}, 
3706: };
3707: static XPoint seg3 3018[] = {
3708:
                                          \{-9, -12\}, \{3, -12\}, \{6, -11\}, \{7, -10\}, \{8, -8\}, \{8, -6\}, \{7, -4\}, \{6, -3\}, \{6, -3\}, \{7, -4\}, \{6, -3\}, \{7, -4\}, \{6, -3\}, \{7, -4\}, \{8, -6\}, \{7, -4\}, \{8, -6\}, \{7, -4\}, \{8, -3\}, \{8, -6\}, \{8, -6\}, \{8, -6\}, \{8, -6\}, \{8, -6\}, \{8, -6\}, \{8, -6\}, \{8, -6\}, \{8, -6\}, \{8, -6\}, \{8, -6\}, \{8, -6\}, \{8, -6\}, \{8, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -6\}, \{9, -
3709:
                                          \{3, -2\}, \{-4, -2\}, 
3710: };
```

```
3711: static XPoint seg4 3018[] = {
3712:
                          \{6, -10\}, \{7, -8\}, \{7, -6\}, \{6, -4\},\
3713: };
3714: static XPoint seg5 3018[] = {
3715:
                          \{3, -12\}, \{5, -11\}, \{6, -9\}, \{6, -5\}, \{5, -3\}, \{3, -2\}, \{3, -2\}, \{5, -3\}, \{3, -2\}, \{5, -3\}, \{3, -2\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\},
3716: };
3717: static XPoint seg6 3018[] = {
3718:
                          \{0, -2\}, \{2, -1\}, \{3, 1\}, \{5, 7\}, \{6, 9\}, \{8, 9\}, \{9, 7\}, \{9, 5\},\
3719: };
3720: static XPoint seg7 3018[] = {
3721:
                          \{5,5\},\{6,7\},\{7,8\},\{8,8\},
3722: };
3723: static XPoint seg8 3018[] = {
3724:
                          \{2, -1\}, \{3, 0\}, \{6, 6\}, \{7, 7\}, \{8, 7\}, \{9, 6\},\
3725: };
3726: static XPoint seg9 3018[] = {
3727:
                          \{-9,9\},\{-1,9\},
3728: };
3729: static XPoint seg10 3018[] = {
3730:
                          \{-8, -12\}, \{-6, -11\},
3731: };
3732: static XPoint seg11 3018[] = {
3733:
                          \{-7, -12\}, \{-6, -10\},\
3734: };
3735: static XPoint seg12 3018[] = {
3736:
                          \{-3, -12\}, \{-4, -10\},
3737: };
3738: static XPoint seg13 3018[] = {
3739:
                          \{-2, -12\}, \{-4, -11\},
3740: };
3741: static XPoint seg14 3018[] = {
3742:
                          \{-6,8\},\{-8,9\},
3743: };
3744: static XPoint seg15 3018[] = {
3745:
                          \{-6,7\},\{-7,9\},
3746: };
3747: static XPoint seg16 3018[] = {
3748:
                          \{-4,7\},\{-3,9\},
3749: };
3750: static XPoint seg17 3018[] = {
3751:
                          \{-4,8\},\{-2,9\},
3752: };
3753: static XPoint *char3018[] = {
3754:
                          seg0 3018, seg1 3018, seg2 3018, seg3 3018, seg4 3018,
                          seg5_3018, seg6_3018, seg7_3018, seg8_3018, seg9_3018,
3755:
3756:
                          seg10 3018, seg11 3018, seg12 3018, seg13 3018,
3757:
                          seg14_3018, seg15_3018, seg16_3018, seg17_3018,
3758:
                          NULL,
3759: };
3760: static int char p3018[] = {
3761:
                          XtNumber(seg0 3018), XtNumber(seg1 3018), XtNumber(seg2 3018),
3762:
                          XtNumber(seg3_3018), XtNumber(seg4_3018), XtNumber(seg5_3018),
3763:
                          XtNumber(seg6_3018), XtNumber(seg7_3018), XtNumber(seg8_3018),
3764:
                          XtNumber(seg9 3018), XtNumber(seg10 3018), XtNumber(seg11 3018),
```

```
3765:
                             XtNumber(seg12 3018), XtNumber(seg13 3018), XtNumber(seg14 3018),
3766:
                            XtNumber(seg15 3018), XtNumber(seg16 3018), XtNumber(seg17 3018),
3767: };
3768: static XPoint seg0 3019[] = {
3769:
                             \{6, -9\}, \{7, -12\}, \{7, -6\}, \{6, -9\}, \{4, -11\}, \{1, -12\},
3770:
                             \{-2, -12\}, \{-5, -11\}, \{-7, -9\}, \{-7, -6\}, \{-6, -4\}, \{-3, -2\}, \{3, 0\}, \{5, 1\},
3771:
                             \{6,3\},\{6,6\},\{5,8\},
3772: };
3773: static XPoint seg1 3019[] = {
3774:
                             \{-6, -6\}, \{-5, -4\}, \{-3, -3\}, \{3, -1\}, \{5, 0\}, \{6, 2\},
3775: };
3776: static XPoint seg2 3019[] = {
3777:
                             \{-5, -11\}, \{-6, -9\}, \{-6, -7\}, \{-5, -5\}, \{-3, -4\}, \{3, -2\}, \{6, 0\}, \{7, 2\}, \{-5, -11\}, \{-6, -9\}, \{-6, -7\}, \{-5, -5\}, \{-3, -4\}, \{-2, -2\}, \{-3, -4\}, \{-2, -2\}, \{-3, -4\}, \{-3, -2\}, \{-3, -4\}, \{-3, -2\}, \{-3, -2\}, \{-3, -4\}, \{-3, -2\}, \{-3, -2\}, \{-3, -4\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2
3778:
                             \{7,5\},\{6,7\},\{5,8\},\{2,9\},\{-1,9\},\{-4,8\},\{-6,6\},\{-7,3\},\{-7,9\},
3779:
                             \{-6, 6\},\
3780: };
3781: static XPoint *char3019[] = {
3782:
                            seg0 3019, seg1 3019, seg2 3019,
3783:
                            NULL,
3784: };
3785: static int char_p3019[] = {
3786:
                            XtNumber(seg0 3019), XtNumber(seg1 3019), XtNumber(seg2 3019),
3787: };
3788: static XPoint seg0 3020[] = {
3789:
                            \{-8, -12\}, \{-8, -6\},
3790: };
3791: static XPoint seg1 3020[] = {
3792:
                            \{-1, -12\}, \{-1, 9\},
3793: };
3794: static XPoint seg2 3020[] = {
3795:
                            \{0, -11\}, \{0, 8\},\
3796: };
3797: static XPoint seg3 3020[] = {
3798:
                             \{1, -12\}, \{1, 9\}, 
3799: };
3800: static XPoint seg4 3020[] = {
3801:
                            \{8, -12\}, \{8, -6\},\
3802: };
3803: static XPoint seg5_3020[] = {
3804:
                             \{-8, -12\}, \{8, -12\}, 
3805: };
3806: static XPoint seg6 3020[] = {
3807:
                             \{-4,9\},\{4,9\},
3808: };
3809: static XPoint seg7 3020[] = {
3810:
                            \{-7, -12\}, \{-8, -6\},
3811: };
3812: static XPoint seg8 3020[] = {
3813:
                             \{-6, -12\}, \{-8, -9\},
3814: };
```

```
3815: static XPoint seg9 3020[] = {
3816:
                          \{-5, -12\}, \{-8, -10\},
3817: };
3818: static XPoint seg10 3020[] = {
3819:
                          \{-3, -12\}, \{-8, -11\},
3820: };
3821: static XPoint seg11 3020[] = {
3822:
                         \{3, -12\}, \{8, -11\},\
3823: };
3824: static XPoint seg12 3020[] = {
3825:
                         \{5, -12\}, \{8, -10\},\
3826: };
3827: static XPoint seg13 3020[] = {
3828:
                         \{6, -12\}, \{8, -9\},\
3829: };
3830: static XPoint seg14 3020[] = {
3831:
                         \{7, -12\}, \{8, -6\},\
3832: };
3833: static XPoint seg15_3020[] = {
3834:
                         \{-1,8\},\{-3,9\},
3835: };
3836: static XPoint seg16 3020[] = {
3837:
                          \{-1,7\},\{-2,9\},
3838: };
3839: static XPoint seg17 3020[] = {
3840:
                          \{1,7\},\{2,9\},\
3841: };
3842: static XPoint seg18 3020[] = {
3843:
                          \{1,8\},\{3,9\},\
3844: };
3845: static XPoint *char3020[] = {
3846:
                          seg0_3020, seg1_3020, seg2_3020, seg3_3020, seg4_3020,
3847:
                          seg5 3020, seg6 3020, seg7 3020, seg8 3020, seg9 3020,
3848:
                          seg10 3020, seg11 3020, seg12 3020, seg13 3020, seg14 3020,
3849:
                          seg15 3020, seg16_3020, seg17_3020, seg18_3020,
3850:
                          NULL,
3851: };
3852: static int char p3020[] = {
3853:
                         XtNumber(seg0 3020), XtNumber(seg1 3020), XtNumber(seg2 3020),
3854:
                          XtNumber(seg3_3020), XtNumber(seg4_3020), XtNumber(seg5_3020),
                          XtNumber(seg6 3020), XtNumber(seg7 3020), XtNumber(seg8 3020),
3855:
                         XtNumber(seg9 3020), XtNumber(seg10 3020), XtNumber(seg11 3020),
3856
3857:
                          XtNumber(seg12 3020),XtNumber(seg13 3020),XtNumber(seg14 3020),
                          XtNumber(seg15 3020),XtNumber(seg16 3020),XtNumber(seg17 3020),
3858:
3859:
                         XtNumber(seg18 3020),
3860: };
3861: static XPoint seg0_3021[] = {
3862:
                          \{-7, -12\}, \{-7, 3\}, \{-6, 6\}, \{-4, 8\}, \{-1, 9\}, \{1, 9\}, \{1, 9\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{
3863:
                          \{4,8\},\{6,6\},\{7,3\},\{7,-11\},
3864: };
3865: static XPoint seg1_3021[] = {
3866:
                          \{-6, -11\}, \{-6, 4\}, \{-5, 6\},\
3867: };
```

```
Listing C.4 Continued
```

```
3868: static XPoint seg2_3021[] = {
3869:
                                               \{-5, -12\}, \{-5, 4\}, \{-4, 7\}, \{-3, 8\}, \{-1, 9\},
3870: };
3871: static XPoint seg3_3021[] = {
3872:
                                                \{-10, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12
3873: };
3874: static XPoint seg4 3021[] = {
3875:
                                               \{4, -12\}, \{10, -12\},\
3876: };
3877: static XPoint seg5 3021[] = {
3878:
                                              \{-9, -12\}, \{-7, -11\},\
3879: };
3880: static XPoint seg6 3021[] = {
                                                \{-8, -12\}, \{-7, -10\},\
3881:
3882: };
3883: static XPoint seg7_3021[] = {
3884:
                                                \{-4, -12\}, \{-5, -10\},
3885: };
3886: static XPoint seg8_3021[] = {
3887:
                                                \{-3, -12\}, \{-5, -11\},
3888: };
3889: static XPoint seg9_3021[] = {
3890:
                                               \{5, -12\}, \{7, -11\},\
3891: };
3892: static XPoint seg10 3021[] = {
3893:
                                                \{9, -12\}, \{7, -11\}, 
3894: };
3895: static XPoint *char3021[] = {
3896:
                                               seg0 3021,seg1 3021,seg2 3021,seg3 3021,seg4 3021,seg5 3021,
3897:
                                                seg6 3021, seg7 3021, seg8 3021, seg9 3021, seg10 3021,
3898:
                                               NULL,
3899: };
3900: static int char p3021[] = {
3901:
                                               XtNumber(seg0_3021), XtNumber(seg1_3021), XtNumber(seg2_3021),
3902:
                                                XtNumber(seg3 3021),XtNumber(seg4 3021),XtNumber(seg5 3021),
3903:
                                               XtNumber(seg6_3021),XtNumber(seg7_3021),XtNumber(seg8_3021),
3904:
                                               XtNumber(seg9 3021), XtNumber(seg10 3021),
3905: };
3906: static XPoint seg0 3022[] = {
3907:
                                                \{-7, -12\}, \{0, 9\},\
3908: };
3909: static XPoint seg1 3022[] = {
3910:
                                               \{-6, -12\}, \{0, 6\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{
3911: };
3912: static XPoint seg2 3022[] = {
3913:
                                               \{-5, -12\}, \{1, 6\},
3914: };
3915: static XPoint seg3 3022[] = {
3916:
                                                \{7, -11\}, \{0, 9\},\
3917: };
```

```
3918: static XPoint seg4 3022[] = {
3919:
                             \{-9, -12\}, \{-2, -12\}, 
3920: };
3921: static XPoint seg5 3022[] = {
3922:
                            \{3, -12\}, \{9, -12\},\
3923: };
3924: static XPoint seg6 3022[] = {
3925:
                            \{-8, -12\}, \{-6, -10\},\
3926: };
3927: static XPoint seg7 3022[] = {
3928:
                             \{-4, -12\}, \{-5, -10\},\
3929: };
3930: static XPoint seg8 3022[] = {
3931:
                            \{-3, -12\}, \{-5, -11\},
3932: };
3933: static XPoint seg9 3022[] = {
3934:
                            \{5, -12\}, \{7, -11\},\
3935: };
3936: static XPoint seg10_3022[] = {
3937:
                            \{8, -12\}, \{7, -11\},\
3938: };
3939: static XPoint *char3022[] = {
3940:
                             seg0_3022, seg1_3022, seg2_3022, seg3_3022, seg4_3022, seg5_3022,
3941:
                             seg6 3022, seg7 3022, seg8 3022, seg9 3022, seg10 3022,
3942:
                            NULL,
3943: };
3944: static int char p3022[] = {
3945:
                            XtNumber(seg0 3022), XtNumber(seg1 3022), XtNumber(seg2 3022),
3946:
                             XtNumber(seg3 3022), XtNumber(seg4 3022), XtNumber(seg5 3022),
                            XtNumber(seg6 3022), XtNumber(seg7 3022), XtNumber(seg8 3022),
3947:
3948:
                            XtNumber(seg9 3022), XtNumber(seg10 3022),
3949: };
3950: static XPoint seg0_3023[] = {
3951:
                            \{-8, -12\}, \{-4, 9\},
3952: };
3953: static XPoint seq1 3023[] = {
3954:
                             \{-7, -12\}, \{-4, 4\}, \{-4, 9\},
3955: };
3956: static XPoint seg2 3023[] = {
3957:
                             \{-6, -12\}, \{-3, 4\},\
3958: };
3959: static XPoint seg3 3023[] = {
3960:
                             \{0, -12\}, \{-3, 4\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, 
3961: };
3962: static XPoint seg4_3023[] = {
3963:
                             \{0, -12\}, \{4, 9\}, 
3964: };
3965: static XPoint seg5 3023[] = {
3966:
                            \{1, -12\}, \{4, 4\}, \{4, 9\},\
3967: };
3968: static XPoint seg6_3023[] = {
3969:
                             \{2, -12\}, \{5, 4\},\
3970: };
```

```
3971: static XPoint seg7 3023[] = {
3972:
                          \{8, -11\}, \{5, 4\}, \{4, 9\},
3973: };
3974: static XPoint seg8 3023[] = {
3975:
                           \{-11, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12\}, \{-3, -12
3976: };
3977: static XPoint seg9 3023[] = {
3978:
                          \{0, -12\}, \{2, -12\}, 
3979: };
3980: static XPoint seg10 3023[] = {
3981:
                          \{5, -12\}, \{11, -12\}, 
3982: };
3983: static XPoint seg11 3023[] = {
3984:
                          \{-10, -12\}, \{-7, -11\},
3985: };
3986: static XPoint seg12 3023[] = {
3987:
                          \{-9, -12\}, \{-7, -10\}, 
3988: };
3989: static XPoint seg13 3023[] = {
3990:
                           \{-5, -12\}, \{-6, -10\},
3991: };
3992: static XPoint seg14 3023[] = {
3993:
                          \{-4, -12\}, \{-6, -11\},
3994: };
3995: static XPoint seg15_3023[] = {
3996:
                          \{6, -12\}, \{8, -11\},\
3997: };
3998: static XPoint seg16 3023[] = {
3999:
                          \{10, -12\}, \{8, -11\},\
4000: };
4001: static XPoint *char3023[] = {
4002:
                           seg0_3023, seg1_3023, seg2_3023, seg3_3023, seg4_3023,
4003:
                           seq5 3023, seq6 3023, seq7 3023, seq8 3023, seq9 3023,
4004:
                           seg10 3023, seg11 3023, seg12 3023, seg13 3023, seg14 3023,
                          seg15_3023, seg16_3023,
4005:
4006:
                          NULL,
4007: };
4008: static int char_p3023[] = {
4009:
                          XtNumber(seg0 3023), XtNumber(seg1 3023), XtNumber(seg2 3023),
4010:
                           XtNumber(seg3 3023), XtNumber(seg4 3023), XtNumber(seg5 3023),
4011:
                          XtNumber(seg6_3023), XtNumber(seg7_3023), XtNumber(seg8_3023),
4012:
                          XtNumber(seg9_3023), XtNumber(seg10_3023), XtNumber(seg11_3023),
4013:
                          XtNumber(seg12 3023), XtNumber(seg13 3023), XtNumber(seg14 3023),
4014:
                          XtNumber(seg15_3023),XtNumber(seg16_3023),
4015: };
4016: static XPoint seg0_3024[] = {
4017:
                           \{-7, -12\}, \{5, 9\}, 
4018: };
4019: static XPoint seg1 3024[] = {
4020:
                           \{-6, -12\}, \{6, 9\},\
4021: };
```

```
4022: static XPoint seg2 3024[] = {
4023:
           \{-5, -12\}, \{7, 9\},
4024: };
4025: static XPoint seg3 3024[] = {
4026:
           \{6, -11\}, \{-6, 8\},\
4027: };
4028: static XPoint seg4 3024[] = {
4029:
           \{-9, -12\}, \{-2, -12\}, 
4030: };
4031: static XPoint seg5 3024[] = {
4032:
           \{3, -12\}, \{9, -12\},\
4033: };
4034: static XPoint seg6 3024[] = {
4035:
           \{-9,9\},\{-3,9\},
4036: };
4037: static XPoint seg7 3024[] = {
4038:
           \{2,9\},\{9,9\},
4039: };
4040: static XPoint seg8 3024[] = {
4041:
           \{-8, -12\}, \{-5, -10\},
4042: };
4043: static XPoint seg9 3024[] = {
4044:
           \{-4, -12\}, \{-5, -10\},
4045: };
4046: static XPoint seg10 3024[] = {
4047:
           \{-3, -12\}, \{-5, -11\},
4048: };
4049: static XPoint seg11 3024[] = {
4050:
           \{4, -12\}, \{6, -11\},\
4051: };
4052: static XPoint seg12 3024[] = {
4053:
           \{8, -12\}, \{6, -11\},\
4054: };
4055: static XPoint seg13 3024[] = {
4056:
           \{-6,8\},\{-8,9\},
4057: };
4058: static XPoint seg14 3024[] = {
4059:
           \{-6,8\},\{-4,9\},
4060: };
4061: static XPoint seg15 3024[] = {
4062:
           \{5,8\},\{3,9\},\
4063: };
4064: static XPoint seg16 3024[] = {
4065:
           \{5,7\},\{4,9\},\
4066: };
4067: static XPoint seg17 3024[] = {
4068:
           \{5,7\},\{8,9\},\
4069: };
4070: static XPoint *char3024[] = {
4071:
           seg0 3024, seg1 3024, seg2 3024, seg3 3024, seg4 3024,
4072:
           seg5 3024, seg6 3024, seg7 3024, seg8 3024, seg9 3024,
4073:
           seg10 3024, seg11 3024, seg12 3024, seg13 3024, seg14 3024,
```

```
4074:
           seg15 3024, seg16 3024, seg17 3024,
4075:
           NULL,
4076: };
4077: static int char_p3024[] = {
4078:
           XtNumber(seg0 3024), XtNumber(seg1 3024), XtNumber(seg2 3024),
4079:
           XtNumber(seg3_3024), XtNumber(seg4_3024), XtNumber(seg5_3024),
4080:
           XtNumber(seg6_3024), XtNumber(seg7_3024), XtNumber(seg8_3024),
4081:
           XtNumber(seg9_3024), XtNumber(seg10_3024), XtNumber(seg11_3024),
4082:
           XtNumber(seg12 3024), XtNumber(seg13 3024), XtNumber(seg14 3024),
4083:
           XtNumber(seg15 3024), XtNumber(seg16 3024), XtNumber(seg17 3024),
4084: };
4085: static XPoint seg0_3025[] = {
4086:
           \{-8, -12\}, \{-1, -1\}, \{-1, 9\}, 
4087: };
4088: static XPoint seg1 3025[] = {
4089:
           \{-7, -12\}, \{0, -1\}, \{0, 8\},\
4090: };
4091: static XPoint seg2 3025[] = {
4092:
           \{-6, -12\}, \{1, -1\}, \{1, 9\}, 
4093: };
4094: static XPoint seg3 3025[] = {
4095:
           \{7, -11\}, \{1, -1\}, 
4096: };
4097: static XPoint seg4 3025[] = {
4098:
           \{-10, -12\}, \{-3, -12\},\
4099: };
4100: static XPoint seg5 3025[] = {
4101:
           \{4, -12\}, \{10, -12\}, 
4102: };
4103: static XPoint seg6 3025[] = {
4104:
           \{-4,9\},\{4,9\},
4105: };
4106: static XPoint seg7 3025[] = {
4107:
           \{-9, -12\}, \{-7, -11\},
4108: };
4109: static XPoint seg8 3025[] = {
4110:
           \{-4, -12\}, \{-6, -11\},
4111: };
4112: static XPoint seg9 3025[] = {
4113:
           \{5, -12\}, \{7, -11\},\
4114: };
4115: static XPoint seg10 3025[] = {
4116:
           \{9, -12\}, \{7, -11\},\
4117: };
4118: static XPoint seg11 3025[] = {
4119:
           \{-1,8\},\{-3,9\},
4120: };
4121: static XPoint seg12 3025[] = {
4122:
           \{-1,7\},\{-2,9\},
4123: };
```

```
4124: static XPoint seg13 3025[] = {
4125:
           \{1,7\},\{2,9\},\
4126: };
4127: static XPoint seg14 3025[] = {
4128:
           \{1,8\},\{3,9\},\
4129: };
4130: static XPoint *char3025[] = {
           seg0 3025, seg1_3025, seg2_3025, seg3_3025, seg4_3025,
4131:
4132:
           seg5 3025, seg6 3025, seg7 3025, seg8 3025, seg9 3025,
4133:
           seg10 3025, seg11 3025, seg12 3025, seg13 3025, seg14 3025,
4134:
           NULL,
4135: };
4136: static int char p3025[] = {
4137:
           XtNumber(seg0 3025), XtNumber(seg1 3025), XtNumber(seg2 3025),
           XtNumber(seg3 3025), XtNumber(seg4 3025), XtNumber(seg5 3025),
4138:
4139:
           XtNumber(seg6 3025), XtNumber(seg7 3025), XtNumber(seg8 3025),
4140:
           XtNumber(seg9_3025), XtNumber(seg10_3025), XtNumber(seg11_3025),
4141:
           XtNumber(seg12 3025), XtNumber(seg13 3025), XtNumber(seg14 3025),
4142: };
4143: static XPoint seg0 3026[] = {
4144:
           \{7, -12\}, \{-7, -12\}, \{-7, -6\},\
4145: };
4146: static XPoint seg1 3026[] = {
4147:
           \{5, -12\}, \{-7, 9\},\
4148: };
4149: static XPoint seg2 3026[] = {
4150:
          \{6, -12\}, \{-6, 9\},\
4151: };
4152: static XPoint seg3 3026[] = {
4153:
           \{7, -12\}, \{-5, 9\},\
4154: };
4155: static XPoint seg4 3026[] = {
4156:
           \{-7,9\},\{7,9\},\{7,3\},
4157: };
4158: static XPoint seg5 3026[] = {
4159:
           \{-6, -12\}, \{-7, -6\},\
4160: };
4161: static XPoint seg6 3026[] = {
4162:
           \{-5, -12\}, \{-7, -9\},\
4163: };
4164: static XPoint seg7 3026[] = {
4165:
           \{-4, -12\}, \{-7, -10\},
4166: };
4167: static XPoint seg8 3026[] = {
4168:
           \{-2, -12\}, \{-7, -11\},\
4169: };
4170: static XPoint seg9_3026[] = {
4171:
          \{2,9\},\{7,8\},\
4172: };
4173: static XPoint seg10 3026[] = {
4174:
           \{4,9\},\{7,7\},\
4175: };
```

```
Listing C.4 Continued
```

```
4176: static XPoint seg11 3026[] = {
4177:
                            \{5,9\},\{7,6\},\
4178: };
4179: static XPoint seg12 3026[] = {
4180:
                             \{6,9\},\{7,3\},\
4181: };
4182: static XPoint *char3026[] = {
4183:
                             seg0_3026, seg1_3026, seg2_3026, seg3_3026, seg4_3026,
4184:
                             seg5 3026, seg6 3026, seg7 3026, seg8 3026, seg9 3026,
4185:
                             seg10 3026, seg11 3026, seg12 3026,
4186:
                             NULL,
4187: };
4188: static int char_p3026[] = {
4189:
                             XtNumber(seg0_3026), XtNumber(seg1_3026), XtNumber(seg2_3026),
4190:
                             XtNumber(seg3 3026), XtNumber(seg4 3026), XtNumber(seg5 3026),
4191:
                             XtNumber(seg6_3026),XtNumber(seg7_3026), XtNumber(seg8_3026),
4192:
                             XtNumber(seg9 3026), XtNumber(seg10 3026), XtNumber(seg11 3026),
4193:
                             XtNumber(seg12 3026),
4194: };
4195: static XPoint seg0 3051[] = {
4196:
                             \{3, -12\}, \{-9, 8\},\
4197: };
4198: static XPoint seg1_3051[] = {
4199:
                             \{1, -8\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3,
4200: };
4201: static XPoint seg2 3051[] = {
4202:
                             \{2, -10\}, \{3, 8\},\
4203: };
4204: static XPoint seg3 3051[] = {
4205:
                            \{3, -12\}, \{3, -10\}, \{4, 7\}, \{4, 9\},\
4206: };
4207: static XPoint seg4 3051[] = {
4208:
                             \{-6,3\},\{2,3\},
4209: };
4210: static XPoint seg5 3051[] = {
4211:
                             \{-12,9\},\{-6,9\},
4212: };
4213: static XPoint seg6_3051[] = {
4214:
                            \{-1,9\},\{6,9\},
4215: };
4216: static XPoint seg7_3051[] = {
4217:
                             \{-9,8\},\{-11,9\},
4218: };
4219: static XPoint seg8 3051[] = {
4220:
                             \{-9,8\},\{-7,9\},\
4221: };
4222: static XPoint seg9 3051[] = {
4223:
                             \{2,8\},\{0,9\},\
4224: };
4225: static XPoint seg10 3051[] = {
4226:
                             \{2,7\},\{1,9\},\
4227: };
```

```
4228: static XPoint seg11 3051[] = {
4229:
                                            \{4,7\},\{5,9\},\
4230: };
4231: static XPoint *char3051[] = {
4232:
                                            seg0 3051, seg1 3051, seg2 3051, seg3 3051, seg4 3051,
4233:
                                            seg5 3051, seg6 3051, seg7 3051, seg8 3051, seg9 3051,
                                           seg10_3051, seg11_3051,
4234:
4235:
                                           NULL,
4236: };
4237: static int char p3051[] = {
4238:
                                           XtNumber(seg0 3051), XtNumber(seg1 3051), XtNumber(seg2 3051),
4239:
                                           XtNumber(seg3 3051), XtNumber(seg4 3051), XtNumber(seg5 3051),
                                           XtNumber(seg6_3051),XtNumber(seg7_3051), XtNumber(seg8_3051),
4240:
4241:
                                           XtNumber(seg9 3051), XtNumber(seg10 3051), XtNumber(seg11 3051),
4242: };
4243: static XPoint seg0 3052[] = {
4244:
                                            \{-3, -12\}, \{-9, 9\},
4245: };
4246: static XPoint seg1 3052[] = {
4247:
                                           \{-2, -12\}, \{-8, 9\}, 
4248: }:
4249: static XPoint seg2 3052[] = {
4250:
                                           \{-1, -12\}, \{-7, 9\},
4251: };
4252: static XPoint seg3 3052[] = {
4253:
                                            \{-6, -12\}, \{5, -12\}, \{8, -11\}, \{9, -9\}, \{9, -7\}, \{8, -4\}, \{7, -3\}, \{4, -2\}, \{9, -12\}, \{1, -2\}, \{2, -2\}, \{2, -2\}, \{3, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -
4254: };
4255: static XPoint seg4 3052[] = {
4256:
                                           \{7, -11\}, \{8, -9\}, \{8, -7\}, \{7, -4\}, \{6, -3\},\
4257: };
4258: static XPoint seg5 3052[] = {
4259:
                                          \{5, -12\}, \{6, -11\}, \{7, -9\}, \{7, -7\}, \{6, -4\}, \{4, -2\},\
4260: };
4261: static XPoint seg6 3052[] = {
4262:
                                            \{-4, -2\}, \{4, -2\}, \{6, -1\}, \{7, 1\}, \{7, 3\}, \{6, 6\}, \{4, 8\}, \{0, 9\},
4263:
                                            \{-12,9\},\
4264: };
4265: static XPoint seg7 3052[] = {
4266:
                                          \{5, -1\}, \{6, 1\}, \{6, 3\}, \{5, 6\}, \{3, 8\}, \{5, 6\}, \{3, 8\}, \{5, 6\}, \{3, 8\}, \{5, 6\}, \{3, 8\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5, 6\}, \{5,
4267: };
4268: static XPoint seg8 3052[] = {
4269:
                                           \{4, -2\}, \{5, 0\}, \{5, 3\}, \{4, 6\}, \{2, 8\}, \{0, 9\},
4270: };
4271: static XPoint seg9 3052[] = {
4272:
                                            \{-5, -12\}, \{-2, -11\},
4273: };
4274: static XPoint seg10 3052[] = {
4275:
                                            \{-4, -12\}, \{-3, -10\},
4276: };
4277: static XPoint seg11 3052[] = {
4278:
                                           \{0, -12\}, \{-2, -10\},\
4279: };
```

```
Listing C.4 Continued
```

```
4280: static XPoint seg12 3052[] = {
4281:
                                        \{1, -12\}, \{-2, -11\},\
4282: };
4283: static XPoint seg13 3052[] = {
4284:
                                       \{-8,8\},\{-11,9\},
4285: };
4286: static XPoint seg14 3052[] = {
4287:
                                       \{-8,7\},\{-10,9\},
4288: };
4289: static XPoint seg15 3052[] = {
4290:
                                       \{-7,7\},\{-6,9\},
4291: };
4292: static XPoint seg16 3052[] = {
4293:
                                        \{-8,8\},\{-5,9\},\
4294: };
4295: static XPoint *char3052[] = {
4296:
                                        seg0_3052, seg1_3052, seg2_3052, seg3_3052, seg4_3052,
4297:
                                        seg5 3052, seg6 3052, seg7 3052, seg8 3052, seg9 3052,
4298:
                                        seg10_3052, seg11_3052, seg12_3052, seg13_3052, seg14_3052,
4299:
                                        seq15 3052, seq16 3052,
4300:
                                       NULL,
4301: };
4302: static int char p3052[] = {
4303:
                                       XtNumber(seg0_3052), XtNumber(seg1_3052), XtNumber(seg2_3052),
                                       XtNumber(seg3 3052), XtNumber(seg4 3052), XtNumber(seg5 3052),
4304:
4305:
                                       XtNumber(seg6 3052), XtNumber(seg7 3052), XtNumber(seg8 3052),
4306:
                                        XtNumber(seg9 3052), XtNumber(seg10 3052), XtNumber(seg11 3052),
                                       XtNumber(seg12_3052),XtNumber(seg13_3052),XtNumber(seg14_3052),
4307:
4308:
                                       XtNumber(seg15 3052), XtNumber(seg16 3052),
4309: };
4310: static XPoint seg0_3053[] = {
4311:
                                        \{8, -10\}, \{9, -10\}, \{10, -12\}, \{9, -6\}, \{9, -8\}, \{8, -10\},
4312:
                                        \{7, -11\}, \{5, -12\}, \{2, -12\}, \{-1, -11\}, \{-3, -9\}, \{-5, -6\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6
4313:
                                        \{-7,1\},\{-7,4\},\{-6,7\},\{-5,8\},\{-2,9\},\{1,9\},\{3,8\},\{5,6\},\{6,4\},
4314: };
4315: static XPoint seg1 3053[] = {
4316:
                                        \{-1, -10\}, \{-3, -8\}, \{-4, -6\}, \{-5, -3\}, \{-6, 1\}, \{-6, 5\}, \{-5, 7\},
4317: };
4318: static XPoint seg2_3053[] = {
4319:
                                        \{2, -12\}, \{0, -11\}, \{-2, -8\}, \{-3, -6\}, \{-4, -3\}, \{-5, 1\}, \{-5, 6\}, \{-4, 8\}, \{-3, -6\}, \{-4, -3\}, \{-5, -1\}, \{-5, -6\}, \{-4, -2\}, \{-2, -2\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6\}, \{-3, -6
4320:
                                        \{-2,9\},\
4321: };
4322: static XPoint *char3053[] = {
4323:
                                        seg0 3053, seg1 3053, seg2 3053,
4324:
                                       NULL,
4325: };
4326: static int char_p3053[] = {
4327:
                                       XtNumber(seg0 3053), XtNumber(seg1 3053), XtNumber(seg2 3053),
4328: };
```

```
4329: static XPoint seg0 3054[] = {
4330:
                                          \{-3, -12\}, \{-9, 9\},
4331: };
4332: static XPoint seg1 3054[] = {
4333:
                                          \{-2, -12\}, \{-8, 9\},
4334: };
4335: static XPoint seg2 3054[] = {
4336:
                                          \{-1, -12\}, \{-7, 9\},\
4337: };
4338: static XPoint seg3 3054[] = {
4339:
                                          \{-6, -12\}, \{3, -12\}, \{6, -11\}, \{7, -10\}, \{8, -7\}, \{8, -3\}, \{7, 1\}, \{5, 5\},
4340:
                                          \{3,7\},\{1,8\},\{-3,9\},\{-12,9\},
4341: };
4342: static XPoint seg4 3054[] = {
4343:
                                          \{5, -11\}, \{6, -10\}, \{7, -7\}, \{7, -3\}, \{6, 1\}, \{4, 5\}, \{2, 7\}, \{6, -10\}, \{7, -7\}, \{7, -3\}, \{6, -1\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -10\}, \{7, -
4344: };
4345: static XPoint seg5_3054[] = {
4346:
                                          \{3, -12\}, \{5, -10\}, \{6, -7\}, \{6, -3\}, \{5, 1\}, \{3, 5\}, \{0, 8\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-
4347: };
4348: static XPoint seg6 3054[] = {
4349:
                                         \{-5, -12\}, \{-2, -11\},
4350: };
4351: static XPoint seg7 3054[] = {
4352:
                                          \{-4, -12\}, \{-3, -10\},\
4353: };
4354: static XPoint seg8 3054[] = {
4355:
                                         \{0, -12\}, \{-2, -10\},\
4356: };
4357: static XPoint seg9 3054[] = {
4358:
                                          \{1, -12\}, \{-2, -11\}, 
4359: };
4360: static XPoint seg10_3054[] = {
4361:
                                          \{-8,8\},\{-11,9\},
4362: };
4363: static XPoint seg11_3054[] = {
4364:
                                          {-8,7},{-10,9},
4365: };
4366: static XPoint seg12 3054[] = {
4367:
                                         \{-7,7\},\{-6,9\},\
4368: };
4369: static XPoint seg13 3054[] = {
4370:
                                         \{-8,8\},\{-5,9\},\
4371: };
4372: static XPoint *char3054[] = {
4373:
                                          seg0 3054, seg1 3054, seg2 3054, seg3 3054, seg4 3054,
4374:
                                          seg5 3054, seg6 3054, seg7 3054, seg8 3054, seg9 3054,
4375:
                                          seg10_3054, seg11_3054, seg12_3054, seg13_3054,
4376:
                                          NULL,
4377: };
4378: static int char p3054[] = {
                                          XtNumber(seg0 3054), XtNumber(seg1 3054), XtNumber(seg2 3054),
4379:
4380:
                                          XtNumber(seg3_3054), XtNumber(seg4_3054), XtNumber(seg5_3054),
4381:
                                         XtNumber(seg6_3054), XtNumber(seg7_3054), XtNumber(seg8_3054),
4382:
                                         XtNumber(seg9 3054), XtNumber(seg10 3054), XtNumber(seg11 3054),
```

```
Listing C.4 Continued
```

```
4383:
                                                       XtNumber(seg12 3054), XtNumber(seg13 3054),
4384: };
4385: static XPoint seq0 3055[] = {
4386:
                                                      \{-3, -12\}, \{-9, 9\},
4387: };
4388: static XPoint seg1 3055[] = {
4389:
                                                      \{-2, -12\}, \{-8, 9\},
4390: };
4391: static XPoint seg2 3055[] = {
4392:
                                                       \{-1, -12\}, \{-7, 9\},\
4393: };
4394: static XPoint seg3 3055[] = {
4395:
                                                       \{3, -6\}, \{1, 2\}, 
4396: };
4397: static XPoint seg4 3055[] = {
                                                      \{-6, -12\}, \{9, -12\}, \{8, -6\},\
4398:
4399: };
4400: static XPoint seg5 3055[] = {
4401:
                                                       \{-4, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, 
4402: };
4403: static XPoint seg6 3055[] = {
4404:
                                                       \{-12,9\},\{3,9\},\{5,4\},
4405: };
4406: static XPoint seg7 3055[] = {
4407:
                                                     \{-5, -12\}, \{-2, -11\},\
4408: };
4409: static XPoint seg8 3055[] = {
4410:
                                                      \{-4, -12\}, \{-3, -10\},
4411: };
4412: static XPoint seg9 3055[] = {
4413:
                                                       \{0, -12\}, \{-2, -10\},\
4414: };
4415: static XPoint seg10 3055[] = {
4416:
                                                       \{1, -12\}, \{-2, -11\},\
4417: };
4418: static XPoint seg11 3055[] = {
4419:
                                                      \{5, -12\}, \{8, -11\},\
4420: };
4421: static XPoint seg12 3055[] = {
4422:
                                                       \{6, -12\}, \{8, -10\},\
4423: };
4424: static XPoint seg13 3055[] = {
4425:
                                                       \{7, -12\}, \{8, -9\},\
4426: };
4427: static XPoint seg14 3055[] = {
4428:
                                                      \{8, -12\}, \{8, -6\},\
4429: };
4430: static XPoint seg15 3055[] = {
4431:
                                                       \{3, -6\}, \{1, -2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1
4432: };
```

```
4433: static XPoint seg16 3055[] = {
4434:
                          \{2, -4\}, \{0, -2\}, \{1, 0\}, \{2, -4\}, \{0, -2\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, \{1, 0\}, 
4435: };
4436: static XPoint seg17 3055[] = {
4437:
                         \{2, -3\}, \{-1, -2\}, \{1, -1\},\
4438: };
4439: static XPoint seg18 3055[] = {
4440:
                         \{-8,8\},\{-11,9\},
4441: };
4442: static XPoint seg19 3055[] = {
4443:
                         \{-8,7\},\{-10,9\},
4444: };
4445: static XPoint seg20 3055[] = {
4446:
                         \{-7,7\},\{-6,9\},\
4447: };
4448: static XPoint seg21 3055[] = {
4449:
                         \{-8,8\},\{-5,9\},\
4450: };
4451: static XPoint seg22 3055[] = {
4452:
                         \{-2,9\},\{3,8\},
4453: };
4454: static XPoint seg23 3055[] = {
4455:
                          \{0,9\},\{3,7\},\
4456: };
4457: static XPoint seg24 3055[] = {
4458:
                         \{3,7\},\{5,4\},\
4459: };
4460: static XPoint *char3055[] = {
                          seg0 3055, seg1 3055, seg2 3055, seg3 3055, seg4 3055,
4461:
                          seg5 3055, seg6 3055, seg7 3055, seg8 3055, seg9 3055,
4462:
4463:
                          seg10_3055, seg11_3055, seg12_3055, seg13_3055, seg14_3055,
4464:
                          seg15_3055, seg16_3055, seg17_3055, seg18_3055, seg19_3055,
4465:
                          seg20 3055, seg21 3055, seg22 3055, seg23 3055, seg24 3055,
4466:
                          NULL,
4467: };
4468: static int char p3055[] = {
4469:
                         XtNumber(seg0 3055), XtNumber(seg1 3055), XtNumber(seg2 3055),
4470:
                          XtNumber(seg3_3055), XtNumber(seg4_3055), XtNumber(seg5_3055),
4471:
                         XtNumber(seg6 3055), XtNumber(seg7 3055), XtNumber(seg8 3055),
4472:
                         XtNumber(seg9 3055), XtNumber(seg10 3055), XtNumber(seg11 3055),
                         XtNumber(seg12 3055), XtNumber(seg13 3055), XtNumber(seg14 3055),
4473:
                         XtNumber(seg15 3055), XtNumber(seg16 3055), XtNumber(seg17 3055),
4474 .
4475:
                         XtNumber(seg18 3055), XtNumber(seg19 3055), XtNumber(seg20 3055),
4476:
                         XtNumber(seg21 3055), XtNumber(seg22 3055), XtNumber(seg23 3055),
4477:
                         XtNumber(seg24 3055),
4478: };
4479: static XPoint seg0_3056[] = {
4480:
                         \{-3, -12\}, \{-9, 9\},
4481: };
4482: static XPoint seg1 3056[] = {
4483:
                          \{-2, -12\}, \{-8, 9\},\
4484: };
```

```
Listing C.4 Continued
```

```
4485: static XPoint seg2 3056[] = {
4486:
                                                   \{-1, -12\}, \{-7, 9\},
4487: };
4488: static XPoint seg3 3056[] = {
4489:
                                                      \{3, -6\}, \{1, 2\}, 
4490: };
4491: static XPoint seg4 3056[] = {
4492:
                                                     \{-6, -12\}, \{9, -12\}, \{8, -6\},\
4493: };
4494: static XPoint seg5 3056[] = {
4495:
                                                   \{-4, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, 
4496: };
4497: static XPoint seg6 3056[] = {
4498:
                                                    {-12,9},{-4,9},
4499: };
4500: static XPoint seg7_3056[] = {
4501:
                                                     \{-5, -12\}, \{-2, -11\}, 
4502: };
4503: static XPoint seg8 3056[] = {
4504:
                                                      \{-4, -12\}, \{-3, -10\},
4505: };
4506: static XPoint seg9 3056[] = {
4507:
                                                   \{0, -12\}, \{-2, -10\},\
4508: };
4509: static XPoint seg10_3056[] = {
4510:
                                                     \{1, -12\}, \{-2, -11\},\
4511: };
4512: static XPoint seg11 3056[] = {
4513:
                                                    \{5, -12\}, \{8, -11\},\
4514: };
4515: static XPoint seg12 3056[] = {
4516:
                                                    \{6, -12\}, \{8, -10\},\
4517: };
4518: static XPoint seg13_3056[] = {
4519:
                                                   \{7, -12\}, \{8, -9\},\
4520: };
4521: static XPoint seg14 3056[] = {
4522:
                                                     \{8, -12\}, \{8, -6\},\
4523: };
4524: static XPoint seg15 3056[] = {
4525:
                                                      \{3, -6\}, \{1, -2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1
4526: };
4527: static XPoint seg16 3056[] = {
4528:
                                                    \{2, -4\}, \{0, -2\}, \{1, 0\},\
4529: };
4530: static XPoint seg17_3056[] = {
4531:
                                                     \{2, -3\}, \{-1, -2\}, \{1, -1\},\
4532: };
4533: static XPoint seg18 3056[] = {
4534:
                                                      \{-8,8\},\{-11,9\},
4535: };
```

```
4536: static XPoint seg19 3056[] = {
4537:
                                     \{-8,7\},\{-10,9\},
4538: };
4539: static XPoint seg20 3056[] = {
4540:
                                     \{-7,7\},\{-6,9\},\
4541: };
4542: static XPoint seg21 3056[] = {
4543:
                                     \{-8,8\},\{-5,9\},\
4544: };
4545: static XPoint *char3056[] = {
4546:
                                     seg0 3056, seg1 3056, seg2 3056, seg3 3056, seg4 3056,
                                     seg5 3056, seg6 3056, seg7 3056, seg8 3056, seg9 3056,
4547.
4548:
                                     seg10 3056, seg11 3056, seg12 3056, seg13 3056, seg14 3056,
4549:
                                     seg15 3056, seg16 3056, seg17 3056, seg18 3056, seg19 3056,
4550:
                                     seg20 3056, seg21 3056,
4551:
                                     NULL,
4552: };
4553: static int char p3056[] = {
4554:
                                     XtNumber(seg0_3056), XtNumber(seg1_3056), XtNumber(seg2_3056),
4555:
                                     XtNumber(seg3_3056), XtNumber(seg4_3056), XtNumber(seg5_3056),
4556:
                                     XtNumber(seg6_3056), XtNumber(seg7_3056), XtNumber(seg8_3056),
4557:
                                     XtNumber(seg9 3056), XtNumber(seg10 3056), XtNumber(seg11 3056),
4558:
                                     XtNumber(seg12 3056), XtNumber(seg13 3056), XtNumber(seg14 3056),
4559:
                                     XtNumber(seg15 3056), XtNumber(seg16 3056), XtNumber(seg17 3056),
4560:
                                     XtNumber(seg18 3056), XtNumber(seg19 3056), XtNumber(seg20 3056),
4561:
                                     XtNumber(seg21_3056),
4562: };
4563: static XPoint seg0 3057[] = {
                                     \{8, -10\}, \{9, -10\}, \{10, -12\}, \{9, -6\}, \{9, -8\}, \{8, -10\}, \{9, -10\}, \{10, -12\}, \{9, -6\}, \{9, -8\}, \{10, -10\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\},
4564:
4565:
                                     \{7, -11\}, \{5, -12\}, \{2, -12\}, \{-1, -11\}, \{-3, -9\}, \{-5, -6\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6, -3\}, \{-6
4566:
                                     \{-7,1\},\{-7,4\},\{-6,7\},\{-5,8\},\{-2,9\},\{0,9\},\{3,8\},\{5,6\},\{7,2\},
4567: };
4568: static XPoint seg1 3057[] = {
4569:
                                     \{-1, -10\}, \{-3, -8\}, \{-4, -6\}, \{-5, -3\}, \{-6, 1\}, \{-6, 5\}, \{-5, 7\},
4570: };
4571: static XPoint seg2 3057[] = {
4572:
                                     \{4,6\},\{5,5\},\{6,2\},
4573: };
4574: static XPoint seg3_3057[] = {
                                     \{2, -12\}, \{0, -11\}, \{-2, -8\}, \{-3, -6\}, \{-4, -3\}, \{-5, 1\}, \{-5, 6\}, \{-4, 8\},
4575:
4576:
                                     \{-2,9\},\
4577: };
4578: static XPoint seg4 3057[] = {
4579:
                                     \{0,9\},\{2,8\},\{4,5\},\{5,2\},
4580: };
4581: static XPoint seg5 3057[] = {
4582:
                                     \{2,2\},\{10,2\},\
4583: };
4584: static XPoint seg6_3057[] = {
4585:
                                     \{3,2\},\{5,3\},\
4586: };
4587: static XPoint seg7 3057[] = {
4588:
                                     \{4,2\},\{5,5\},\
4589: };
```

```
Listing C.4 Continued
```

```
4590: static XPoint seg8 3057[] = {
4591:
                              \{8,2\},\{6,4\},\
4592: };
4593: static XPoint seg9 3057[] = {
4594:
                              \{9,2\},\{6,3\},\
4595: };
4596: static XPoint *char3057[] = {
                              seg0_3057, seg1_3057, seg2_3057, seg3_3057, seg4_3057, seg5_3057,
4597:
4598:
                               seg6 3057, seg7 3057, seg8 3057, seg9 3057,
4599:
                              NULL,
4600: };
4601: static int char p3057[] = {
4602:
                              XtNumber(seg0_3057),XtNumber(seg1_3057),XtNumber(seg2_3057),
4603:
                              XtNumber(seg3_3057),XtNumber(seg4_3057),XtNumber(seg5_3057),
4604:
                              XtNumber(seg6 3057), XtNumber(seg7 3057), XtNumber(seg8 3057),
4605:
                              XtNumber(seg9 3057),
4606: };
4607: static XPoint seg0 3058[] = {
4608:
                               \{-4, -12\}, \{-10, 9\},
4609: };
4610: static XPoint seg1 3058[] = {
4611:
                              \{-3, -12\}, \{-9, 9\},
4612: };
4613: static XPoint seg2 3058[] = {
4614:
                              \{-2, -12\}, \{-8, 9\},
4615: };
4616: static XPoint seg3 3058[] = {
4617:
                              \{8, -12\}, \{2, 9\},\
4618: };
4619: static XPoint seg4 3058[] = {
4620:
                              \{9, -12\}, \{3, 9\},\
4621: };
4622: static XPoint seq5 3058[] = {
4623:
                               \{10, -12\}, \{4, 9\},\
4624: };
4625: static XPoint seg6 3058[] = {
4626:
                              \{-7, -12\}, \{1, -12\}, 
4627: };
4628: static XPoint seg7_3058[] = {
4629:
                              \{5, -12\}, \{13, -12\}, 
4630: };
4631: static XPoint seg8 3058[] = {
4632:
                              \{-6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, \{6, -2\}, 
4633: };
4634: static XPoint seg9 3058[] = {
4635:
                              \{-13,9\},\{-5,9\},
4636: };
4637: static XPoint seg10 3058[] = {
4638:
                               \{-1,9\},\{7,9\},
4639: };
```

```
4640: static XPoint seg11 3058[] = {
4641:
           \{-6, -12\}, \{-3, -11\},
4642: };
4643: static XPoint seg12 3058[] = {
4644:
           \{-5, -12\}, \{-4, -10\},
4645: };
4646: static XPoint seg13 3058[] = {
4647:
           \{-1, -12\}, \{-3, -10\},\
4648: };
4649: static XPoint seg14 3058[] = {
4650:
           \{0, -12\}, \{-3, -11\},\
4651: };
4652: static XPoint seg15 3058[] = {
4653:
           \{6, -12\}, \{9, -11\},\
4654: };
4655: static XPoint seg16 3058[] = {
4656:
           \{7, -12\}, \{8, -10\},\
4657: };
4658: static XPoint seg17 3058[] = {
4659:
           \{11, -12\}, \{9, -10\},\
4660: };
4661: static XPoint seg18 3058[] = {
4662:
           \{12, -12\}, \{9, -11\},\
4663: };
4664: static XPoint seg19 3058[] = {
4665:
           \{-9,8\},\{-12,9\},
4666: };
4667: static XPoint seg20 3058[] = {
4668:
           \{-9,7\},\{-11,9\},
4669: };
4670: static XPoint seg21 3058[] = {
4671:
           \{-8,7\},\{-7,9\},\
4672: };
4673: static XPoint seg22 3058[] = {
4674:
           \{-9,8\},\{-6,9\},\
4675: };
4676: static XPoint seg23 3058[] = {
4677:
           \{3,8\},\{0,9\},\
4678: };
4679: static XPoint seg24 3058[] = {
4680:
           \{3,7\},\{1,9\},\
4681: };
4682: static XPoint seg25 3058[] = {
4683:
           \{4,7\},\{5,9\},\
4684: };
4685: static XPoint seg26_3058[] = {
4686:
           \{3,8\},\{6,9\},\
4687: };
4688: static XPoint *char3058[] = {
4689:
           seg0 3058, seg1 3058, seg2 3058, seg3 3058, seg4 3058,
4690:
           seg5 3058, seg6 3058, seg7 3058, seg8 3058, seg9 3058,
4691:
           seg10 3058, seg11 3058, seg12 3058, seg13 3058, seg14 3058,
```

```
4692:
           seg15 3058, seg16 3058, seg17 3058, seg18 3058, seg19 3058,
4693:
           seg20 3058, seg21 3058, seg22 3058, seg23 3058, seg24 3058,
4694:
           seg25 3058, seg26 3058,
4695:
           NULL,
4696: };
4697: static int char_p3058[] = {
4698:
           XtNumber(seg0_3058), XtNumber(seg1_3058), XtNumber(seg2_3058),
4699:
           XtNumber(seg3_3058), XtNumber(seg4_3058), XtNumber(seg5_3058),
4700:
           XtNumber(seg6 3058), XtNumber(seg7 3058), XtNumber(seg8 3058),
          XtNumber(seg9 3058), XtNumber(seg10_3058),XtNumber(seg11_3058),
4701:
4702:
           XtNumber(seg12_3058), XtNumber(seg13_3058), XtNumber(seg14_3058),
4703:
          XtNumber(seg15 3058), XtNumber(seg16 3058), XtNumber(seg17 3058),
4704:
          XtNumber(seg18_3058), XtNumber(seg19_3058), XtNumber(seg20_3058),
4705:
           XtNumber(seg21 3058), XtNumber(seg22 3058), XtNumber(seg23 3058),
4706:
          XtNumber(seg24 3058), XtNumber(seg25 3058), XtNumber(seg26 3058),
4707: };
4708: static XPoint seg0 3059[] = {
4709:
          \{2, -12\}, \{-4, 9\},\
4710: };
4711: static XPoint seg1 3059[] = {
4712:
           \{3, -12\}, \{-3, 9\},\
4713: };
4714: static XPoint seg2 3059[] = {
4715:
           \{4, -12\}, \{-2, 9\},\
4716: };
4717: static XPoint seg3 3059[] = {
4718:
           \{-1, -12\}, \{7, -12\}, 
4719: };
4720: static XPoint seg4 3059[] = {
4721:
          \{-7,9\},\{1,9\},
4722: };
4723: static XPoint seg5 3059[] = {
4724:
           \{0, -12\}, \{3, -11\},\
4725: };
4726: static XPoint seg6 3059[] = {
4727:
           \{1, -12\}, \{2, -10\},\
4728: };
4729: static XPoint seg7 3059[] = {
4730:
          \{5, -12\}, \{3, -10\},\
4731: };
4732: static XPoint seg8 3059[] = {
4733:
           \{6, -12\}, \{3, -11\},\
4734: };
4735: static XPoint seg9 3059[] = {
4736:
           \{-3,8\},\{-6,9\},\
4737: };
4738: static XPoint seg10 3059[] = {
4739:
           \{-3,7\},\{-5,9\},
4740: };
```

```
4741: static XPoint seg11 3059[] = {
4742:
           \{-2,7\},\{-1,9\},
4743: };
4744: static XPoint seg12 3059[] = {
4745:
           \{-3,8\},\{0,9\},
4746: };
4747: static XPoint *char3059[] = {
4748:
           seg0 3059, seg1 3059, seg2 3059, seg3 3059, seg4 3059,
           seg5 3059, seg6 3059, seg7 3059, seg8 3059, seg9 3059,
4749:
4750:
           seg10 3059, seg11 3059, seg12 3059,
4751:
           NULL,
4752: };
4753: static int char p3059[] = {
4754:
           XtNumber(seg0 3059), XtNumber(seg1 3059), XtNumber(seg2 3059),
4755:
           XtNumber(seg3 3059), XtNumber(seg4 3059), XtNumber(seg5 3059),
4756:
           XtNumber(seg6 3059), XtNumber(seg7 3059), XtNumber(seg8 3059),
4757:
           XtNumber(seg9_3059), XtNumber(seg10_3059), XtNumber(seg11_3059),
4758:
           XtNumber(seg12 3059),
4759: };
4760: static XPoint seg0 3060[] = {
4761:
           \{5, -12\}, \{0, 5\}, \{-1, 7\}, \{-3, 9\},\
4762: };
4763: static XPoint seg1 3060[] = {
4764:
           \{6, -12\}, \{2, 1\}, \{1, 4\}, \{0, 6\},\
4765: };
4766: static XPoint seg2 3060[] = {
4767:
           \{7, -12\}, \{3, 1\}, \{1, 6\}, \{-1, 8\}, \{-3, 9\}, \{-5, 9\}, \{-7, 8\}, \{-8, 6\},
4768:
           \{-8,4\},\{-7,3\},\{-6,3\},\{-5,4\},\{-5,5\},\{-6,6\},\{-7,6\},
4769: };
4770: static XPoint seg3 3060[] = {
4771:
           \{-7,4\},\{-7,5\},\{-6,5\},\{-6,4\},\{-7,4\},
4772: };
4773: static XPoint seg4 3060[] = {
4774:
           \{2, -12\}, \{10, -12\}, 
4775: };
4776: static XPoint seg5 3060[] = {
4777:
           \{3, -12\}, \{6, -11\},\
4778: };
4779: static XPoint seg6 3060[] = {
4780:
           \{4, -12\}, \{5, -10\},\
4781: };
4782: static XPoint seg7 3060[] = {
4783:
           \{8, -12\}, \{6, -10\},\
4784: };
4785: static XPoint seg8 3060[] = {
4786:
           \{9, -12\}, \{6, -11\},\
4787: };
4788: static XPoint *char3060[] = {
4789:
           seg0_3060, seg1_3060, seg2_3060, seg3_3060, seg4_3060, seg5_3060,
4790:
           seg6 3060, seg7 3060, seg8 3060,
4791:
           NULL,
4792: };
4793: static int char p3060[] = {
4794:
           XtNumber(seg0 3060), XtNumber(seg1 3060), XtNumber(seg2 3060),
```

```
Listing C.4 Continued
```

```
4795:
                                                    XtNumber(seg3 3060), XtNumber(seg4 3060), XtNumber(seg5 3060),
4796:
                                                    XtNumber(seg6 3060),XtNumber(seg7_3060),XtNumber(seg8_3060),
4797: };
4798: static XPoint seg0 3061[] = {
4799:
                                                     \{-3, -12\}, \{-9, 9\},
4800: };
4801: static XPoint seg1 3061[] = {
4802:
                                                    \{-2, -12\}, \{-8, 9\},\
4803: };
4804: static XPoint seg2 3061[] = {
4805:
                                                   \{-1, -12\}, \{-7, 9\},\
4806: };
4807: static XPoint seg3_3061[] = {
4808:
                                                    \{10, -11\}, \{-5, 0\},\
4809: };
4810: static XPoint seg4 3061[] = {
4811:
                                                    \{-1, -3\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3
4812: };
4813: static XPoint seg5 3061[] = {
4814:
                                                     \{0, -3\}, \{4, 9\},\
4815: };
4816: static XPoint seg6 3061[] = {
4817:
                                                    \{1, -4\}, \{5, 8\},\
4818: };
4819: static XPoint seg7_3061[] = {
4820:
                                                    \{-6, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2,
4821: };
4822: static XPoint seg8 3061[] = {
4823:
                                                    \{7, -12\}, \{13, -12\}, 
4824: };
4825: static XPoint seg9 3061[] = {
4826:
                                                   \{-12,9\},\{-4,9\},
4827: };
4828: static XPoint seg10 3061[] = {
4829:
                                                    \{0,9\},\{7,9\},\
4830: };
4831: static XPoint seg11 3061[] = {
4832:
                                                    \{-5, -12\}, \{-2, -11\},
4833: };
4834: static XPoint seg12 3061[] = {
4835:
                                                     \{-4, -12\}, \{-3, -10\},\
4836: };
4837: static XPoint seg13 3061[] = {
4838:
                                                    \{0, -12\}, \{-2, -10\},\
4839: };
4840: static XPoint seg14 3061[] = {
4841:
                                                    \{1, -12\}, \{-2, -11\}, 
4842: };
4843: static XPoint seg15 3061[] = {
4844:
                                                     \{8, -12\}, \{10, -11\},\
4845: };
```

```
4846: static XPoint seg16 3061[] = {
4847:
                         \{12, -12\}, \{10, -11\}, 
4848: };
4849: static XPoint seg17 3061[] = {
4850:
                         \{-8,8\},\{-11,9\},\
4851: };
4852: static XPoint seg18 3061[] = {
4853:
                         \{-8,7\},\{-10,9\},\
4854: };
4855: static XPoint seg19 3061[] = {
4856:
                         \{-7,7\},\{-6,9\},
4857: };
4858: static XPoint seg20 3061[] = {
4859:
                         \{-8,8\},\{-5,9\},\
4860: };
4861: static XPoint seg21 3061[] = {
4862:
                         \{3,8\},\{1,9\},\
4863: };
4864: static XPoint seg22_3061[] = {
4865:
                         \{3,7\},\{2,9\},\
4866: };
4867: static XPoint seg23 3061[] = {
4868:
                         \{4,7\},\{6,9\},\
4869: };
4870: static XPoint *char3061[] = {
                         seg0 3061, seg1_3061, seg2_3061, seg3_3061, seg4_3061,
4871:
4872:
                         seg5 3061, seg6 3061, seg7 3061, seg8 3061, seg9 3061,
4873:
                         seg10 3061, seg11 3061, seg12 3061, seg13 3061, seg14 3061,
4874:
                         seg15 3061, seg16 3061, seg17 3061, seg18 3061, seg19 3061,
                         seg20 3061, seg21 3061, seg22 3061, seg23 3061,
4875:
4876:
                         NULL,
4877: };
4878: static int char_p3061[] = {
4879:
                         XtNumber(seg0 3061), XtNumber(seg1 3061), XtNumber(seg2 3061),
4880:
                         XtNumber(seg3_3061), XtNumber(seg4_3061), XtNumber(seg5_3061),
4881:
                         XtNumber(seg6 3061), XtNumber(seg7 3061), XtNumber(seg8 3061),
4882:
                         XtNumber(seg9 3061), XtNumber(seg10 3061), XtNumber(seg11 3061),
                         XtNumber(seg12 3061),XtNumber(seg13_3061),XtNumber(seg14_3061),
4883:
4884:
                         XtNumber(seg15 3061),XtNumber(seg16 3061),XtNumber(seg17 3061),
4885:
                         XtNumber(seg18 3061), XtNumber(seg19 3061), XtNumber(seg20 3061),
                         XtNumber(seg21 3061),XtNumber(seg22 3061),XtNumber(seg23 3061),
4886:
4887: };
4888: static XPoint seg0 3062[] = {
4889:
                         \{-1, -12\}, \{-7, 9\},
4890: };
4891: static XPoint seg1 3062[] = {
4892:
                         \{0, -12\}, \{-6, 9\},\
4893: };
4894: static XPoint seg2_3062[] = {
4895:
                         \{1, -12\}, \{-5, 9\},\
4896: };
4897: static XPoint seg3 3062[] = {
4898:
                         \{-4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4,
4899: };
```

```
4900: static XPoint seg4 3062[] = {
4901:
           \{-10,9\},\{5,9\},\{7,3\},
4902: };
4903: static XPoint seg5_3062[] = {
4904:
           \{-3, -12\}, \{0, -11\}, 
4905: };
4906: static XPoint seg6 3062[] = {
4907:
           \{-2, -12\}, \{-1, -10\},
4908: };
4909: static XPoint seg7 3062[] = {
4910:
          \{2, -12\}, \{0, -10\},\
4911: };
4912: static XPoint seg8 3062[] = {
          \{3, -12\}, \{0, -11\},\
4913:
4914: };
4915: static XPoint seg9 3062[] = {
4916:
           \{-6,8\},\{-9,9\},
4917: };
4918: static XPoint seg10_3062[] = {
4919:
           \{-6,7\},\{-8,9\},
4920: };
4921: static XPoint seg11_3062[] = {
4922:
          \{-5,7\},\{-4,9\},
4923: };
4924: static XPoint seg12 3062[] = {
4925:
           \{-6,8\},\{-3,9\},
4926: };
4927: static XPoint seg13 3062[] = {
4928:
           \{0,9\},\{5,8\},\
4929: };
4930: static XPoint seg14 3062[] = {
4931:
          \{2,9\},\{6,6\},\
4932: };
4933: static XPoint seg15_3062[] = {
4934:
           \{4,9\},\{7,3\},\
4935: };
4936: static XPoint *char3062[] = {
4937:
           seg0 3062, seg1 3062, seg2 3062, seg3 3062, seg4 3062,
4938:
           seg5_3062, seg6_3062, seg7_3062, seg8_3062, seg9_3062,
4939:
           seg10_3062, seg11_3062, seg12_3062, seg13_3062, seg14_3062,
4940:
           seg15 3062,
4941:
          NULL,
4942: };
4943: static int char p3062[] = {
          XtNumber(seg0 3062), XtNumber(seg1 3062), XtNumber(seg2 3062),
4944:
4945:
          XtNumber(seg3_3062), XtNumber(seg4_3062), XtNumber(seg5_3062),
4946:
          XtNumber(seg6 3062), XtNumber(seg7 3062), XtNumber(seg8 3062),
4947:
           XtNumber(seg9 3062), XtNumber(seg10 3062), XtNumber(seg11 3062),
4948:
          XtNumber(seg12_3062), XtNumber(seg13_3062), XtNumber(seg14_3062),
4949:
          XtNumber(seg15 3062),
```

```
4950: };
4951: static XPoint seg0_3063[] = {
4952:
           \{-5, -12\}, \{-11, 8\},
4953: };
4954: static XPoint seg1_3063[] = {
4955:
           \{-5, -11\}, \{-4, 7\}, \{-4, 9\},
4956: };
4957: static XPoint seg2 3063[] = {
4958:
           \{-4, -12\}, \{-3, 7\},
4959: };
4960: static XPoint seg3 3063[] = {
4961:
           \{-3, -12\}, \{-2, 6\},
4962: };
4963: static XPoint seg4 3063[] = {
4964:
           \{9, -12\}, \{-2, 6\}, \{-4, 9\},\
4965: };
4966: static XPoint seg5_3063[] = {
4967:
           \{9, -12\}, \{3, 9\},\
4968: };
4969: static XPoint seg6 3063[] = {
4970:
           \{10, -12\}, \{4, 9\},\
4971: };
4972: static XPoint seg7_3063[] = {
4973:
           \{11, -12\}, \{5, 9\},\
4974: };
4975: static XPoint seg8 3063[] = {
4976:
           \{-8, -12\}, \{-3, -12\},
4977: };
4978: static XPoint seg9 3063[] = {
4979:
           \{9, -12\}, \{14, -12\}, 
4980: };
4981: static XPoint seg10_3063[] = {
4982:
           \{-14,9\},\{-8,9\},
4983: };
4984: static XPoint seg11_3063[] = {
4985:
           \{0,9\},\{8,9\},\
4986: };
4987: static XPoint seg12 3063[] = {
           \{-7, -12\}, \{-5, -11\},\
4988:
4989: };
4990: static XPoint seg13 3063[] = {
4991:
           \{-6, -12\}, \{-5, -10\},
4992: };
4993: static XPoint seg14 3063[] = {
4994:
           \{12, -12\}, \{10, -10\},\
4995: };
4996: static XPoint seg15_3063[] = {
4997:
           \{13, -12\}, \{10, -11\}, 
4998: };
4999: static XPoint seg16 3063[] = {
5000:
           \{-11,8\},\{-13,9\},
5001: };
```

```
Listing C.4 Continued
```

```
5002: static XPoint seg17_3063[] = {
5003:
                          \{-11,8\},\{-9,9\},
5004: };
5005: static XPoint seg18 3063[] = {
5006:
                          \{4,8\},\{1,9\},\
5007: };
5008: static XPoint seg19 3063[] = {
5009:
                          \{4,7\},\{2,9\},\
5010: };
5011: static XPoint seg20 3063[] = {
5012:
                          \{5,7\},\{6,9\},\
5013: };
5014: static XPoint seg21 3063[] = {
5015:
                          \{4,8\},\{7,9\},\
5016: };
5017: static XPoint *char3063[] = {
                          seg0 3063, seg1 3063, seg2 3063, seg3 3063, seg4 3063,
5018:
5019:
                          seg5_3063, seg6_3063, seg7_3063, seg8_3063, seg9_3063,
5020:
                          seg10 3063, seg11 3063, seg12 3063, seg13 3063, seg14 3063,
                          seg15 3063, seg16 3063, seg17 3063, seg18 3063, seg19 3063,
5021:
5022:
                          seg20 3063, seg21 3063,
5023:
                          NULL,
5024: };
5025: static int char_p3063[] = {
5026:
                          XtNumber(seg0_3063), XtNumber(seg1_3063), XtNumber(seg2_3063),
5027:
                          XtNumber(seg3 3063), XtNumber(seg4 3063), XtNumber(seg5 3063),
5028:
                          XtNumber(seg6_3063), XtNumber(seg7_3063), XtNumber(seg8_3063),
5029:
                          XtNumber(seg9 3063), XtNumber(seg10 3063), XtNumber(seg11 3063),
5030:
                          XtNumber(seg12 3063), XtNumber(seg13 3063), XtNumber(seg14 3063),
5031:
                          XtNumber(seg15 3063),XtNumber(seg16 3063),XtNumber(seg17 3063),
5032:
                          XtNumber(seg18 3063), XtNumber(seg19 3063), XtNumber(seg20 3063),
5033:
                          XtNumber(seg21 3063),
5034: };
5035: static XPoint seg0 3064[] = {
5036:
                          \{-3, -12\}, \{-9, 8\},
5037: };
5038: static XPoint seg1 3064[] = {
5039:
                          \{-3, -12\}, \{4, 9\},\
5040: };
5041: static XPoint seg2 3064[] = {
5042:
                          \{-2, -12\}, \{4, 6\},\
5043: };
5044: static XPoint seg3 3064[] = {
5045:
                          \{-1, -12\}, \{5, 6\},\
5046: };
5047: static XPoint seg4 3064[] = {
5048:
                          \{10, -11\}, \{5, 6\}, \{4, 9\}, 
5049: };
5050: static XPoint seg5 3064[] = {
5051:
                          \{-6, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}
5052: };
```

```
5053: static XPoint seg6 3064[] = {
                                         \{7, -12\}, \{13, -12\}, 
5054:
5055: };
5056: static XPoint seg7 3064[] = {
5057:
                                         \{-12,9\},\{-6,9\},
5058: };
5059: static XPoint seg8 3064[] = {
5060:
                                         \{-5, -12\}, \{-2, -11\},
5061: };
5062: static XPoint seg9 3064[] = {
5063:
                                         \{-4, -12\}, \{-2, -10\}, 
5064: };
5065: static XPoint seg10 3064[] = {
5066:
                                         \{8, -12\}, \{10, -11\},\
5067: };
5068: static XPoint seg11 3064[] = {
5069:
                                        \{12, -12\}, \{10, -11\},\
5070: };
5071: static XPoint seg12 3064[] = {
5072:
                                        \{-9,8\},\{-11,9\},
5073: };
5074: static XPoint seg13 3064[] = {
5075:
                                         \{-9,8\},\{-7,9\},
5076: };
5077: static XPoint *char3064[] = {
                                         seg0 3064, seg1_3064, seg2_3064, seg3_3064, seg4_3064,
5078:
5079:
                                         seq5 3064, seq6 3064, seq7 3064, seq8 3064, seq9 3064,
                                         seg10_3064, seg11_3064, seg12_3064, seg13_3064,
5080:
5081:
                                        NULL,
5082: };
5083: static int char p3064[] = {
                                        XtNumber(seg0_3064), XtNumber(seg1_3064), XtNumber(seg2_3064),
5084:
5085:
                                         XtNumber(seg3 3064), XtNumber(seg4 3064), XtNumber(seg5 3064),
5086:
                                         XtNumber(seg6 3064), XtNumber(seg7 3064), XtNumber(seg8 3064),
5087:
                                        XtNumber(seg9_3064), XtNumber(seg10_3064), XtNumber(seg11_3064),
5088:
                                        XtNumber(seg12 3064), XtNumber(seg13 3064),
5089: };
5090: static XPoint seg0 3065[] = {
5091:
                                         \{1, -12\}, \{-2, -11\}, \{-4, -9\}, \{-6, -6\}, \{-7, -3\}, \{-8, 1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8, -1\}, \{-8,
5092:
                                         \{-8,4\},\{-7,7\},\{-6,8\},\{-4,9\},\{-1,9\},\{2,8\},\{4,6\},\{6,3\},\{7,0\},
5093:
                                         \{8, -4\}, \{8, -7\}, \{7, -10\}, \{6, -11\}, \{4, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{2, -12\}, \{3, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -1
5094: };
5095: static XPoint seg1 3065[] = {
5096:
                                         \{-3, -9\}, \{-5, -6\}, \{-6, -3\}, \{-7, 1\}, \{-7, 5\}, \{-6, 7\},
5097: };
5098: static XPoint seg2 3065[] = {
5099:
                                         \{3,6\},\{5,3\},\{6,0\},\{7,-4\},\{7,-8\},\{6,-10\},
5100: };
5101: static XPoint seg3 3065[] = {
5102:
                                         \{1, -12\}, \{-1, -11\}, \{-3, -8\}, \{-4, -6\}, \{-5, -3\}, \{-6, 1\}, \{-6, 6\}, \{-5, 8\},
5103:
                                         \{-4,9\},\
5104: };
```

```
5105: static XPoint seg4 3065[] = {
5106:
                            \{-1,9\},\{1,8\},\{3,5\},\{4,3\},\{5,0\},\{6,-4\},\{6,-9\},\{5,-11\},
5107:
                             \{4, -12\},\
5108: };
5109: static XPoint *char3065[] = {
5110:
                            seg0 3065, seg1 3065, seg2 3065, seg3 3065, seg4 3065,
5111:
                            NULL,
5112: };
5113: static int char p3065[] = {
5114:
                            XtNumber(seg0 3065), XtNumber(seg1 3065), XtNumber(seg2 3065),
5115:
                            XtNumber(seg3_3065),XtNumber(seg4_3065),
5116: };
5117: static XPoint seg0 3066[] = {
5118:
                            \{-3, -12\}, \{-9, 9\},
5119: };
5120: static XPoint seg1 3066[] = {
5121:
                            \{-2, -12\}, \{-8, 9\},\
5122: };
5123: static XPoint seg2 3066[] = {
5124:
                            \{-1, -12\}, \{-7, 9\},
5125: };
5126: static XPoint seg3 3066[] = {
5127:
                            \{-6, -12\}, \{6, -12\}, \{9, -11\}, \{10, -9\}, \{10, -7\}, \{9, -4\}, \{7, -2\}, \{3, -1\}, \{10, -7\}, \{9, -4\}, \{7, -2\}, \{3, -1\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\}, \{10, -7\},
5128:
                             \{-5, -1\},\
5129: };
5130: static XPoint seg4_3066[] = {
5131:
                            \{8, -11\}, \{9, -9\}, \{9, -7\}, \{8, -4\}, \{6, -2\},
5132: };
5133: static XPoint seg5 3066[] = {
5134:
                            \{6, -12\}, \{7, -11\}, \{8, -9\}, \{8, -7\}, \{7, -4\}, \{5, -2\}, \{3, -1\},
5135: };
5136: static XPoint seg6 3066[] = {
5137:
                            \{-12,9\},\{-4,9\},
5138: };
5139: static XPoint seg7_3066[] = {
5140:
                            \{-5, -12\}, \{-2, -11\},
5141: };
5142: static XPoint seg8 3066[] = {
5143:
                            \{-4, -12\}, \{-3, -10\},
5144: };
5145: static XPoint seg9_3066[] = {
5146:
                            \{0, -12\}, \{-2, -10\},\
5147: };
5148: static XPoint seg10_3066[] = {
5149:
                            \{1, -12\}, \{-2, -11\},\
5150: };
5151: static XPoint seg11_3066[] = {
5152:
                            \{-8,8\},\{-11,9\},
5153: };
5154: static XPoint seg12 3066[] = {
5155:
                            \{-8,7\},\{-10,9\},
5156: };
```

```
5157: static XPoint seg13 3066[] = {
                         \{-7,7\},\{-6,9\},
5158:
5159: };
5160: static XPoint seg14 3066[] = {
5161:
                         \{-8,8\},\{-5,9\},\
5162: };
5163: static XPoint *char3066[] = {
5164:
                         seg0 3066, seg1 3066, seg2 3066, seg3 3066, seg4 3066,
                         seg5 3066, seg6 3066, seg7 3066, seg8 3066, seg9 3066,
5165:
5166:
                         seg10 3066, seg11 3066, seg12 3066, seg13 3066, seg14 3066,
5167:
                         NULL,
5168: };
5169: static int char p3066[] = {
5170:
                         XtNumber(seg0 3066), XtNumber(seg1 3066), XtNumber(seg2 3066),
5171:
                         XtNumber(seg3 3066),XtNumber(seg4 3066),XtNumber(seg5 3066),
5172:
                         XtNumber(seg6 3066), XtNumber(seg7 3066), XtNumber(seg8 3066),
5173:
                         XtNumber(seg9_3066), XtNumber(seg10_3066), XtNumber(seg11_3066),
5174:
                         XtNumber(seg12 3066), XtNumber(seg13 3066), XtNumber(seg14 3066),
5175: };
5176: static XPoint seg0 3067[] = {
5177:
                         \{1, -12\}, \{-2, -11\}, \{-4, -9\}, \{-6, -6\}, \{-7, -3\}, \{-8, 1\},
5178:
                         \{-8,4\},\{-7,7\},\{-6,8\},\{-4,9\},\{-1,9\},\{2,8\},\{4,6\},\{6,3\},\{7,0\},
5179:
                         \{8, -4\}, \{8, -7\}, \{7, -10\}, \{6, -11\}, \{4, -12\}, \{1, -12\},
5180: };
5181: static XPoint seq1 3067[] = {
5182:
                         \{-3, -9\}, \{-5, -6\}, \{-6, -3\}, \{-7, 1\}, \{-7, 5\}, \{-6, 7\},
5183: };
5184: static XPoint seg2 3067[] = {
5185:
                         \{3,6\},\{5,3\},\{6,0\},\{7,-4\},\{7,-8\},\{6,-10\},
5186: };
5187: static XPoint seg3 3067[] = {
5188:
                         \{1, -12\}, \{-1, -11\}, \{-3, -8\}, \{-4, -6\}, \{-5, -3\}, \{-6, 1\}, \{-6, 6\}, \{-5, 8\},
5189:
                         \{-4,9\},\
5190: };
5191: static XPoint seg4_3067[] = {
                         \{-1,9\},\{1,8\},\{3,5\},\{4,3\},\{5,0\},\{6,-4\},\{6,-9\},\{5,-11\},
5192:
5193:
                         \{4, -12\},\
5194: };
5195: static XPoint seg5 3067[] = {
5196:
                         \{-6,6\},\{-5,4\},\{-3,3\},\{-2,3\},\{0,4\},\{1,6\},\{2,11\},\{3,12\},
5197:
                         \{4, 12\}, \{5, 11\}, 
5198: };
5199: static XPoint seg6 3067[] = {
5200:
                         \{2, 12\}, \{3, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{4, 13\}, \{
5201: };
5202: static XPoint seg7 3067[] = {
5203:
                         \{1,6\},\{1,13\},\{2,14\},\{4,14\},\{5,11\},\{5,10\},
5204: };
5205: static XPoint *char3067[] = {
5206:
                         seg0 3067, seg1 3067, seg2 3067, seg3 3067, seg4 3067, seg5 3067,
5207:
                         seg6 3067, seg7 3067,
5208:
                         NULL,
5209: };
```
```
5210: static int char p3067[] = {
                                                XtNumber(seg0 3067), XtNumber(seg1 3067), XtNumber(seg2 3067),
5211:
5212:
                                                XtNumber(seg3 3067), XtNumber(seg4 3067), XtNumber(seg5 3067),
5213:
                                                XtNumber(seg6_3067),XtNumber(seg7_3067),
5214: };
5215: static XPoint seg0 3068[] = {
5216:
                                                \{-3, -12\}, \{-9, 9\},
5217: };
5218: static XPoint seg1 3068[] = {
5219:
                                                \{-2, -12\}, \{-8, 9\},
5220: };
5221: static XPoint seg2 3068[] = {
5222:
                                                \{-1, -12\}, \{-7, 9\},
5223: };
5224: static XPoint seg3_3068[] = {
5225:
                                                \{-6, -12\}, \{5, -12\}, \{8, -11\}, \{9, -9\}, \{9, -7\}, \{8, -4\}, \{7, -3\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2
5226:
                                                 \{-4, -2\},\
5227: };
5228: static XPoint seg4_3068[] = {
5229:
                                                \{7, -11\}, \{8, -9\}, \{8, -7\}, \{7, -4\}, \{6, -3\}, \{7, -4\}, \{6, -3\}, \{7, -4\}, \{6, -3\}, \{7, -4\}, \{6, -3\}, \{7, -4\}, \{6, -3\}, \{7, -4\}, \{6, -3\}, \{7, -4\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, \{8, -2\}, 
5230: };
5231: static XPoint seg5_3068[] = {
5232:
                                                \{5, -12\}, \{6, -11\}, \{7, -9\}, \{7, -7\}, \{6, -4\}, \{4, -2\},\
5233: };
5234: static XPoint seg6 3068[] = {
5235:
                                                 \{0, -2\}, \{2, -1\}, \{3, 0\}, \{5, 6\}, \{6, 7\}, \{7, 7\}, \{8, 6\},\
5236: };
5237: static XPoint seg7 3068[] = {
5238:
                                                \{5,7\},\{6,8\},\{7,8\},
5239: };
5240: static XPoint seg8 3068[] = {
5241:
                                                \{3,0\},\{4,8\},\{5,9\},\{7,9\},\{8,6\},\{8,5\},
5242: };
5243: static XPoint seg9_3068[] = {
5244:
                                                \{-12,9\},\{-4,9\},
5245: };
5246: static XPoint seg10 3068[] = {
5247:
                                                 \{-5, -12\}, \{-2, -11\},\
5248: };
5249: static XPoint seg11 3068[] = {
5250:
                                                 \{-4, -12\}, \{-3, -10\},\
5251: };
5252: static XPoint seg12 3068[] = {
5253:
                                                \{0, -12\}, \{-2, -10\},\
5254: };
5255: static XPoint seg13 3068[] = {
5256:
                                                 \{1, -12\}, \{-2, -11\}, 
5257: };
```

```
5258: static XPoint seg14 3068[] = {
5259:
                                      \{-8,8\},\{-11,9\},
5260: };
5261: static XPoint seg15 3068[] = {
5262:
                                      \{-8,7\},\{-10,9\},
5263: };
5264: static XPoint seg16 3068[] = {
5265:
                                      \{-7,7\},\{-6,9\},
5266: };
5267: static XPoint seg17_3068[] = {
5268:
                                      \{-8,8\},\{-5,9\},
5269: };
5270: static XPoint *char3068[] = {
5271:
                                      seg0 3068, seg1 3068, seg2 3068, seg3 3068, seg4 3068,
5272:
                                      seg5 3068, seg6 3068, seg7 3068, seg8 3068, seg9 3068,
5273:
                                      seg10 3068, seg11 3068, seg12 3068, seg13 3068, seg14 3068,
5274:
                                      seg15_3068, seg16_3068, seg17_3068,
5275:
                                      NULL,
5276: };
5277: static int char p3068[] = {
                                      XtNumber(seg0_3068),XtNumber(seg1_3068),XtNumber(seg2_3068),
5278:
5279:
                                      XtNumber(seg3 3068), XtNumber(seg4 3068), XtNumber(seg5 3068),
5280:
                                      XtNumber(seg6_3068),XtNumber(seg7_3068),XtNumber(seg8_3068),
5281:
                                      XtNumber(seg9_3068), XtNumber(seg10_3068), XtNumber(seg11_3068),
5282:
                                      XtNumber(seg12 3068), XtNumber(seg13 3068), XtNumber(seg14 3068),
5283:
                                      XtNumber(seg15 3068), XtNumber(seg16 3068), XtNumber(seg17 3068),
5284: };
5285: static XPoint seg0 3069[] = {
                                      \{8, -10\}, \{9, -10\}, \{10, -12\}, \{9, -6\}, \{9, -8\}, \{8, -10\}, \{7, -11\}, \{4, -12\}, \{9, -10\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12\}, \{10, -12
5286:
                                      \{0, -12\}, \{-3, -11\}, \{-5, -9\}, \{-5, -6\}, \{-4, -4\}, \{-2, -2\}, \{4, 1\}, \{-2, -2\}, \{4, 1\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2
5287:
5288:
                                      \{5,3\},\{5,6\},\{4,8\},
5289: };
5290: static XPoint seg1 3069[] = {
5291:
                                      \{-4, -6\}, \{-3, -4\}, \{4, 0\}, \{5, 2\},
5292: };
5293: static XPoint seg2 3069[] = {
5294:
                                      \{-3, -11\}, \{-4, -9\}, \{-4, -7\}, \{-3, -5\}, \{3, -2\}, \{5, 0\}, \{6, 2\}, \{6, 5\},
5295:
                                      \{5,7\},\{4,8\},\{1,9\},\{-3,9\},\{-6,8\},\{-7,7\},\{-8,5\},\{-8,3\},\{-9,9\},
5296:
                                      \{-8,7\},\{-7,7\},\
5297: };
5298: static XPoint *char3069[] = {
5299:
                                      seg0 3069, seg1 3069, seg2 3069,
5300:
                                      NULL,
5301: };
5302: static int char p3069[] = {
5303:
                                      XtNumber(seg0 3069), XtNumber(seg1 3069), XtNumber(seg2 3069),
5304: };
5305: static XPoint seg0 3070[] = {
5306:
                                      \{2, -12\}, \{-4, 9\},\
5307: };
5308: static XPoint seg1 3070[] = {
5309:
                                      \{3, -12\}, \{-3, 9\},\
5310: };
```

```
5311: static XPoint seg2 3070[] = {
5312:
          \{4, -12\}, \{-2, 9\},\
5313: };
5314: static XPoint seg3 3070[] = {
5315:
           \{-5, -12\}, \{-7, -6\},\
5316: };
5317: static XPoint seg4 3070[] = {
5318:
           \{11, -12\}, \{10, -6\},\
5319: };
5320: static XPoint seg5 3070[] = {
5321:
          \{-5, -12\}, \{11, -12\}, 
5322: };
5323: static XPoint seg6 3070[] = {
5324:
          {-7,9},{1,9},
5325: };
5326: static XPoint seg7_3070[] = {
5327:
           \{-4, -12\}, \{-7, -6\},\
5328: };
5329: static XPoint seg8 3070[] = {
5330:
           \{-2, -12\}, \{-6, -9\},
5331: };
5332: static XPoint seg9 3070[] = {
5333:
          \{0, -12\}, \{-5, -11\},\
5334: };
5335: static XPoint seg10_3070[] = {
5336:
           \{7, -12\}, \{10, -11\},\
5337: };
5338: static XPoint seg11 3070[] = {
5339:
           \{8, -12\}, \{10, -10\},\
5340: };
5341: static XPoint seg12 3070[] = {
5342:
          \{9, -12\}, \{10, -9\},\
5343: };
5344: static XPoint seg13 3070[] = {
5345:
          \{10, -12\}, \{10, -6\},\
5346: };
5347: static XPoint seg14 3070[] = {
5348:
           \{-3,8\},\{-6,9\},
5349: };
5350: static XPoint seg15 3070[] = {
5351:
           \{-3,7\},\{-5,9\},
5352: };
5353: static XPoint seg16_3070[] = {
5354:
           \{-2,7\},\{-1,9\},
5355: };
5356: static XPoint seg17_3070[] = {
5357:
           \{-3,8\},\{0,9\},
5358: };
5359: static XPoint *char3070[] = {
5360:
           seg0 3070, seg1 3070, seg2 3070, seg3 3070, seg4 3070,
5361:
           seg5 3070, seg6 3070, seg7 3070, seg8 3070, seg9 3070,
```

```
seg10 3070, seg11 3070, seg12_3070, seg13_3070, seg14_3070,
5362:
                         seg15_3070, seg16_3070, seg17_3070,
5363:
5364:
                         NULL,
5365: };
5366: static int char_p3070[] = {
                         XtNumber(seg0_3070), XtNumber(seg1_3070), XtNumber(seg2_3070),
5367:
5368:
                         XtNumber(seg3_3070), XtNumber(seg4_3070), XtNumber(seg5_3070),
                         XtNumber(seg6 3070), XtNumber(seg7_3070), XtNumber(seg8_3070),
5369:
5370:
                         XtNumber(seg9 3070), XtNumber(seg10 3070), XtNumber(seg11 3070),
5371:
                         XtNumber(seg12_3070), XtNumber(seg13_3070), XtNumber(seg14_3070),
5372:
                         XtNumber(seg15 3070), XtNumber(seg16 3070), XtNumber(seg17 3070),
5373: };
5374: static XPoint seg0 3071[] = {
5375:
                         \{-4, -12\}, \{-7, -1\}, \{-8, 3\}, \{-8, 6\}, \{-7, 8\}, \{-4, 9\}, \{-4, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-2, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}
5376:
                         \{0,9\},\{3,8\},\{5,6\},\{6,3\},\{10,-11\},
5377: };
5378: static XPoint seg1_3071[] = {
5379:
                         \{-3, -12\}, \{-6, -1\}, \{-7, 3\}, \{-7, 7\}, \{-6, 8\},
5380: };
5381: static XPoint seg2 3071[] = {
5382:
                         \{-2, -12\}, \{-5, -1\}, \{-6, 3\}, \{-6, 7\}, \{-4, 9\},
5383: };
5384: static XPoint seg3 3071[] = {
5385:
                         \{-7, -12\}, \{1, -12\}, 
5386: };
5387: static XPoint seg4 3071[] = {
5388:
                         \{7, -12\}, \{13, -12\}, 
5389: };
5390: static XPoint seg5 3071[] = {
5391:
                         \{-6, -12\}, \{-3, -11\},
5392: };
5393: static XPoint seg6 3071[] = {
5394:
                         \{-5, -12\}, \{-4, -10\},
5395: };
5396: static XPoint seg7_3071[] = {
5397:
                         \{-1, -12\}, \{-3, -10\},
5398: };
5399: static XPoint seg8 3071[] = {
5400:
                         \{0, -12\}, \{-3, -11\},\
5401: };
5402: static XPoint seg9 3071[] = {
5403:
                         \{8, -12\}, \{10, -11\}, 
5404: };
5405: static XPoint seg10 3071[] = {
5406:
                         \{12, -12\}, \{10, -11\},\
5407: };
5408: static XPoint *char3071[] = {
5409:
                         seg0 3071, seg1 3071, seg2 3071, seg3 3071, seg4 3071, seg5 3071,
5410:
                         seg6_3071, seg7_3071, seg8_3071, seg9_3071, seg10_3071,
5411:
                         NULL,
5412: };
5413: static int char p3071[] = {
5414:
                         XtNumber(seg0_3071), XtNumber(seg1_3071), XtNumber(seg2_3071),
```

```
Listing C.4 Continued
```

```
5415:
           XtNumber(seg3 3071), XtNumber(seg4 3071), XtNumber(seg5 3071),
5416:
           XtNumber(seg6_3071), XtNumber(seg7_3071), XtNumber(seg8_3071),
5417:
           XtNumber(seg9 3071), XtNumber(seg10 3071),
5418: };
5419: static XPoint seg0 3072[] = {
5420:
           \{-4, -12\}, \{-4, -10\}, \{-3, 7\}, \{-3, 9\},
5421: };
5422: static XPoint seg1 3072[] = {
5423:
          \{-3, -11\}, \{-2, 6\},
5424: };
5425: static XPoint seg2_3072[] = {
5426:
           \{-2, -12\}, \{-1, 5\}, 
5427: };
5428: static XPoint seg3 3072[] = {
5429:
           \{9, -11\}, \{-3, 9\},\
5430: };
5431: static XPoint seg4 3072[] = {
5432:
          \{-6, -12\}, \{1, -12\}, 
5433: };
5434: static XPoint seg5 3072[] = {
5435:
           \{6, -12\}, \{12, -12\}, 
5436: };
5437: static XPoint seg6 3072[] = {
5438:
           \{-5, -12\}, \{-4, -10\},
5439: };
5440: static XPoint seg7 3072[] = {
5441:
           \{-1, -12\}, \{-2, -10\}, 
5442: };
5443: static XPoint seg8 3072[] = {
5444:
          \{0, -12\}, \{-3, -11\},\
5445: };
5446: static XPoint seg9 3072[] = {
5447:
           \{7, -12\}, \{9, -11\}, 
5448: };
5449: static XPoint seg10 3072[] = {
5450:
           \{11, -12\}, \{9, -11\},\
5451: };
5452: static XPoint *char3072[] = {
           seg0_3072,seg1_3072,seg2_3072,seg3_3072,seg4_3072,seg5_3072,
5453:
5454:
           seg6 3072, seg7 3072, seg8 3072, seg9 3072, seg10 3072,
5455:
           NULL,
5456: };
5457: static int char p3072[] = {
5458:
           XtNumber(seg0 3072), XtNumber(seg1 3072), XtNumber(seg2 3072),
5459:
           XtNumber(seg3_3072),XtNumber(seg4_3072),XtNumber(seg5_3072),
5460:
           XtNumber(seg6 3072), XtNumber(seg7 3072), XtNumber(seg8 3072),
           XtNumber(seg9 3072), XtNumber(seg10 3072),
5461:
5462: };
5463: static XPoint seg0 3073[] = {
5464:
           \{-5, -12\}, \{-5, -10\}, \{-7, 7\}, \{-7, 9\},
5465: };
```

```
5466: static XPoint seg1 3073[] = {
5467:
           \{-4, -11\}, \{-6, 6\},\
5468: };
5469: static XPoint seg2 3073[] = {
5470:
           \{-3, -12\}, \{-5, 5\},
5471: };
5472: static XPoint seg3 3073[] = {
5473:
           \{3, -12\}, \{-5, 5\}, \{-7, 9\},\
5474: };
5475: static XPoint seg4_3073[] = {
5476:
           \{3, -12\}, \{3, -10\}, \{1, 7\}, \{1, 9\},\
5477: };
5478: static XPoint seg5 3073[] = {
5479:
           \{4, -11\}, \{2, 6\},\
5480: };
5481: static XPoint seg6_3073[] = {
5482:
           \{5, -12\}, \{3, 5\},\
5483: };
5484: static XPoint seg7 3073[] = {
5485:
           \{11, -11\}, \{3, 5\}, \{1, 9\},\
5486: };
5487: static XPoint seg8 3073[] = {
5488:
           \{-8, -12\}, \{0, -12\}, 
5489: };
5490: static XPoint seg9 3073[] = {
5491:
           \{3, -12\}, \{5, -12\},\
5492: };
5493: static XPoint seg10 3073[] = {
5494:
           \{8, -12\}, \{14, -12\}, 
5495: };
5496: static XPoint seg11 3073[] = {
5497:
           \{-7, -12\}, \{-4, -11\},
5498: };
5499: static XPoint seg12 3073[] = {
5500:
           \{-6, -12\}, \{-5, -10\},
5501: };
5502: static XPoint seg13 3073[] = {
5503:
           \{-2, -12\}, \{-4, -9\},
5504: };
5505: static XPoint seg14 3073[] = {
5506:
           \{-1, -12\}, \{-4, -11\},
5507: };
5508: static XPoint seg15 3073[] = {
5509:
           \{9, -12\}, \{11, -11\},\
5510: };
5511: static XPoint seg16 3073[] = {
5512:
           \{13, -12\}, \{11, -11\},\
5513: };
5514: static XPoint *char3073[] = {
5515:
           seg0 3073, seg1 3073, seg2 3073, seg3 3073, seg4 3073,
5516:
           seg5 3073, seg6 3073, seg7 3073, seg8 3073, seg9 3073,
5517:
           seg10 3073, seg11 3073, seg12 3073, seg13 3073, seg14 3073,
```

```
5518:
                             seg15 3073, seg16 3073,
5519:
                             NULL,
5520: };
5521: static int char p3073[] = {
5522:
                             XtNumber(seg0_3073),XtNumber(seg1_3073),XtNumber(seg2_3073),
5523:
                             XtNumber(seg3 3073),XtNumber(seg4 3073),XtNumber(seg5 3073),
5524:
                             XtNumber(seg6 3073), XtNumber(seg7 3073), XtNumber(seg8 3073),
5525:
                             XtNumber(seg9 3073), XtNumber(seg10 3073), XtNumber(seg11 3073),
5526:
                             XtNumber(seg12 3073), XtNumber(seg13 3073), XtNumber(seg14 3073),
                             XtNumber(seg15 3073), XtNumber(seg16 3073),
5527:
5528: };
5529: static XPoint seg0 3074[] = {
5530:
                             \{-4, -12\}, \{2, 9\},\
5531: };
5532: static XPoint seg1 3074[] = {
5533:
                             \{-3, -12\}, \{3, 9\},\
5534: };
5535: static XPoint seg2 3074[] = {
5536:
                             \{-2, -12\}, \{4, 9\},\
5537: };
5538: static XPoint seg3 3074[] = {
5539:
                             \{9, -11\}, \{-9, 8\},\
5540: };
5541: static XPoint seg4 3074[] = {
                             \{-6, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1,
5542:
5543: };
5544: static XPoint seg5 3074[] = {
5545:
                             \{6, -12\}, \{12, -12\}, 
5546: };
5547: static XPoint seg6 3074[] = {
                             \{-12,9\},\{-6,9\},
5548:
5549: };
5550: static XPoint seg7_3074[] = {
5551:
                             \{-1,9\},\{6,9\},
5552: };
5553: static XPoint seg8 3074[] = {
5554:
                             \{-5, -12\}, \{-3, -10\},
5555: };
5556: static XPoint seg9 3074[] = {
5557:
                             \{-1, -12\}, \{-2, -10\}, 
5558: };
5559: static XPoint seg10 3074[] = {
5560:
                             \{0, -12\}, \{-2, -11\},\
5561: };
5562: static XPoint seg11 3074[] = {
5563:
                             \{7, -12\}, \{9, -11\},\
5564: };
5565: static XPoint seg12 3074[] = {
5566:
                             \{11, -12\}, \{9, -11\},\
5567: };
```

```
5568: static XPoint seg13 3074[] = {
5569:
                                                          \{-9,8\},\{-11,9\},
5570: };
5571: static XPoint seg14 3074[] = {
5572:
                                                          \{-9,8\},\{-7,9\},\
5573: };
5574: static XPoint seg15 3074[] = {
5575:
                                                         \{2,8\},\{0,9\},\
5576: };
5577: static XPoint seg16 3074[] = {
5578:
                                                         \{2,7\},\{1,9\},\
5579: };
5580: static XPoint seg17 3074[] = {
                                                         \{3,7\},\{5,9\},\
5581:
5582: };
5583: static XPoint *char3074[] = {
5584:
                                                          seg0_3074, seg1_3074, seg2_3074, seg3_3074, seg4_3074,
5585:
                                                          seg5 3074, seg6 3074, seg7 3074, seg8 3074, seg9 3074,
5586:
                                                          seg10_3074, seg11_3074, seg12_3074, seg13_3074, seg14_3074,
5587:
                                                          seg15 3074, seg16 3074, seg17 3074,
5588:
                                                         NULL,
5589: };
5590: static int char_p3074[] = {
5591:
                                                         XtNumber(seg0_3074), XtNumber(seg1_3074), XtNumber(seg2_3074),
5592:
                                                          XtNumber(seg3 3074),XtNumber(seg4 3074),XtNumber(seg5 3074),
5593:
                                                         XtNumber(seg6_3074), XtNumber(seg7_3074), XtNumber(seg8_3074),
5594:
                                                          XtNumber(seq9 3074), XtNumber(seq10 3074), XtNumber(seq11 3074),
5595:
                                                         XtNumber(seg12 3074), XtNumber(seg13 3074), XtNumber(seg14 3074),
5596:
                                                          XtNumber(seg15 3074), XtNumber(seg16 3074), XtNumber(seg17 3074),
5597: };
5598: static XPoint seg0 3075[] = {
5599:
                                                          \{-5, -12\}, \{-1, -2\}, \{-4, 9\}, \{-5, -12\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2, -2\}, \{-2
5600: };
5601: static XPoint seg1 3075[] = {
5602:
                                                         \{-4, -12\}, \{0, -2\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\},
5603: };
5604: static XPoint seg2 3075[] = {
5605:
                                                         \{-3, -12\}, \{1, -2\}, \{-2, 9\},
5606: };
5607: static XPoint seg3 3075[] = {
5608:
                                                         \{10, -11\}, \{1, -2\},\
5609: };
5610: static XPoint seg4 3075[] = {
5611:
                                                          \{-7, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0,
5612: };
5613: static XPoint seg5 3075[] = {
5614:
                                                         \{7, -12\}, \{13, -12\}, 
5615: };
5616: static XPoint seg6_3075[] = {
5617:
                                                          \{-7,9\},\{1,9\},
5618: };
5619: static XPoint seg7 3075[] = {
5620:
                                                          \{-6, -12\}, \{-4, -11\},
5621: };
```

```
Listing C.4 Continued
```

```
5622: static XPoint seg8 3075[] = {
5623:
           \{-2, -12\}, \{-3, -10\},\
5624: };
5625: static XPoint seg9_3075[] = {
5626:
           \{-1, -12\}, \{-4, -11\}, 
5627: };
5628: static XPoint seg10 3075[] = {
5629:
           \{8, -12\}, \{10, -11\},\
5630: };
5631: static XPoint seg11 3075[] = {
5632:
           \{12, -12\}, \{10, -11\},\
5633: };
5634: static XPoint seg12 3075[] = {
5635:
           \{-3,8\},\{-6,9\},
5636: };
5637: static XPoint seg13 3075[] = {
5638:
           \{-3,7\},\{-5,9\},
5639: };
5640: static XPoint seg14_3075[] = {
5641:
           \{-2,7\},\{-1,9\},
5642: };
5643: static XPoint seg15_3075[] = {
5644:
           \{-3,8\},\{0,9\},\
5645: };
5646: static XPoint *char3075[] = {
5647:
           seg0 3075, seg1 3075, seg2 3075, seg3 3075, seg4 3075,
5648:
           seg5 3075, seg6 3075, seg7 3075, seg8 3075, seg9 3075,
           seg10 3075, seg11 3075, seg12 3075, seg13 3075, seg14 3075,
5649:
5650:
           seg15 3075,
5651:
           NULL,
5652: };
5653: static int char p3075[] = {
5654:
           XtNumber(seg0 3075), XtNumber(seg1 3075), XtNumber(seg2 3075),
5655:
           XtNumber(seg3_3075),XtNumber(seg4_3075),XtNumber(seg5_3075),
5656:
           XtNumber(seg6 3075), XtNumber(seg7 3075), XtNumber(seg8 3075),
5657:
           XtNumber(seg9_3075), XtNumber(seg10_3075), XtNumber(seg11_3075),
5658:
           XtNumber(seg12 3075), XtNumber(seg13 3075), XtNumber(seg14 3075),
           XtNumber(seg15_3075),
5659:
5660: };
5661: static XPoint seg0 3076[] = {
5662:
           \{8, -12\}, \{-10, 9\},\
5663: };
5664: static XPoint seg1 3076[] = {
5665:
           \{9, -12\}, \{-9, 9\},\
5666: };
5667: static XPoint seg2 3076[] = {
5668:
           \{10, -12\}, \{-8, 9\},\
5669: };
```

```
5670: static XPoint seg3 3076[] = {
5671:
                                                          \{10, -12\}, \{-4, -12\}, \{-6, -6\},\
5672: };
5673: static XPoint seg4 3076[] = {
5674:
                                                          \{-10,9\},\{4,9\},\{6,3\},
5675: };
5676: static XPoint seg5 3076[] = {
5677:
                                                         \{-3, -12\}, \{-6, -6\},
5678: };
5679: static XPoint seq6 3076[] = {
5680:
                                                         \{-2, -12\}, \{-5, -9\},\
5681: };
5682: static XPoint seg7 3076[] = {
5683:
                                                         \{0, -12\}, \{-4, -11\},\
5684: };
5685: static XPoint seg8 3076[] = {
5686:
                                                         \{0,9\},\{4,8\},\
5687: };
5688: static XPoint seg9_3076[] = {
5689:
                                                         \{2,9\},\{5,6\},\
5690: };
5691: static XPoint seg10 3076[] = {
5692:
                                                          \{3,9\},\{6,3\},\
5693: };
5694: static XPoint *char3076[] = {
5695:
                                                          seg0_3076, seg1_3076, seg2_3076, seg3_3076, seg4_3076, seg5_3076,
5696:
                                                          seg6 3076, seg7 3076, seg8 3076, seg9 3076, seg10 3076,
5697:
                                                         NULL,
5698: };
5699: static int char p3076[] = {
5700:
                                                         XtNumber(seg0 3076), XtNumber(seg1 3076), XtNumber(seg2 3076),
5701:
                                                         XtNumber(seg3_3076),XtNumber(seg4_3076),XtNumber(seg5_3076),
5702:
                                                         XtNumber(seg6 3076), XtNumber(seg7 3076), XtNumber(seg8 3076),
5703:
                                                         XtNumber(seg9 3076), XtNumber(seg10 3076),
5704: };
5705: static XPoint seq0 3101[] = {
5706:
                                                          \{-4, -2\}, \{-4, -3\}, \{-3, -3\}, \{-3, -1\}, \{-5, -1\}, \{-5, -3\},
5707:
                                                          \{-4, -4\}, \{-2, -5\}, \{2, -5\}, \{4, -4\}, \{5, -3\}, \{6, -1\}, \{6, 6\}, \{7, 8\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 9\}, \{8, 
5708: };
5709: static XPoint seg1 3101[] = {
5710:
                                                         \{4, -3\}, \{5, -1\}, \{5, 6\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6, 8\}, \{6
5711: };
5712: static XPoint seg2 3101[] = {
5713:
                                                          \{2, -5\}, \{3, -4\}, \{4, -2\}, \{4, 6\}, \{5, 8\}, \{8, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{
5714: };
5715: static XPoint seg3 3101[] = {
5716:
                                                          \{4,0\},\{3,1\},\{-2,2\},\{-5,3\},\{-6,5\},\{-6,6\},\{-5,8\},\{-2,9\},
5717:
                                                          \{1,9\},\{3,8\},\{4,6\},
5718: };
5719: static XPoint seg4 3101[] = {
5720:
                                                          \{-4,3\},\{-5,5\},\{-5,6\},\{-4,8\},
5721: };
5722: static XPoint seg5 3101[] = {
5723:
                                                          \{3,1\},\{-1,2\},\{-3,3\},\{-4,5\},\{-4,6\},\{-3,8\},\{-2,9\},
5724: };
```

```
5725: static XPoint *char3101[] = {
5726:
                                             seg0_3101, seg1_3101, seg2_3101, seg3_3101, seg4_3101, seg5_3101,
5727:
                                            NULL,
5728: };
5729: static int char p3101[] = {
5730:
                                            XtNumber(seg0 3101), XtNumber(seg1 3101), XtNumber(seg2 3101),
5731:
                                            XtNumber(seg3 3101), XtNumber(seg4 3101), XtNumber(seg5 3101),
5732: };
5733: static XPoint seg0 3102[] = {
5734:
                                             \{-6, -12\},\
5735:
                                             \{-6,9\},\{-5,8\},\{-3,8\},
5736: };
5737: static XPoint seg1 3102[] = {
5738:
                                            \{-5, -11\}, \{-5, 7\},
5739: };
5740: static XPoint seg2 3102[] = {
5741:
                                            \{-9, -12\}, \{-4, -12\}, \{-4, 8\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12\}, \{-12
5742: };
5743: static XPoint seg3 3102[] = {
5744:
                                             \{-4, -2\}, \{-3, -4\}, \{-1, -5\}, \{1, -5\}, \{4, -4\}, \{6, -2\}, \{7, 1\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7, 3\}, \{7,
5745:
                                             \{6,6\},\{4,8\},\{1,9\},\{-1,9\},\{-3,8\},\{-4,6\},
5746: };
5747: static XPoint seg4_3102[] = {
5748:
                                             \{5, -2\}, \{6, 0\}, \{6, 4\}, \{5, 6\},\
5749: };
5750: static XPoint seg5 3102[] = {
5751:
                                             \{1, -5\}, \{3, -4\}, \{4, -3\}, \{5, 0\}, \{5, 4\}, \{4, 7\}, \{3, 8\}, \{1, 9\},
5752: };
5753: static XPoint seg6 3102[] = {
5754:
                                            \{-8, -12\}, \{-6, -11\},
5755: };
5756: static XPoint seg7 3102[] = {
5757:
                                             \{-7, -12\}, \{-6, -10\},
5758: };
5759: static XPoint *char3102[] = {
5760:
                                            seg0 3102,seg1 3102,seg2 3102,seg3 3102,seg4 3102,seg5 3102,
5761:
                                             seg6 3102, seg7 3102,
5762:
                                            NULL,
5763: };
5764: static int char p3102[] = {
5765:
                                            XtNumber(seg0_3102),XtNumber(seg1_3102),XtNumber(seg2_3102),
5766:
                                             XtNumber(seg3 3102), XtNumber(seg4 3102), XtNumber(seg5 3102),
5767:
                                            XtNumber(seg6 3102), XtNumber(seg7 3102),
5768: };
5769: static XPoint seg0 3103[] = {
5770:
                                            \{5, -1\}, \{5, -2\}, \{4, -2\}, \{4, 0\}, \{6, 0\}, \{6, -2\},\
5771:
                                             \{4, -4\}, \{2, -5\}, \{-1, -5\}, \{-4, -4\}, \{-6, -2\}, \{-7, 1\}, \{-7, 3\}, \{-6, 6\},
5772:
                                             \{-4,8\},\{-1,9\},\{1,9\},\{4,8\},\{6,6\},
5773: };
```

```
5774: static XPoint seg1 3103[] = {
5775:
                            \{-5, -2\}, \{-6, 0\}, \{-6, 4\}, \{-5, 6\},
5776: };
5777: static XPoint seg2 3103[] = {
5778:
                           \{-1, -5\}, \{-3, -4\}, \{-4, -3\}, \{-5, 0\}, \{-5, 4\}, \{-4, 7\}, \{-3, 8\}, \{-1, 9\},
5779: };
5780: static XPoint *char3103[] = {
5781:
                           seg0_3103, seg1_3103, seg2_3103,
5782:
                           NULL,
5783: };
5784: static int char p3103[] = {
5785:
                           XtNumber(seg0 3103), XtNumber(seg1 3103), XtNumber(seg2 3103),
5786: };
5787: static XPoint seg0 3104[] = {
5788:
                           \{4, -12\}, \{4, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9, 9\}, \{9
5789: };
5790: static XPoint seg1_3104[] = {
5791:
                           \{5, -11\}, \{5, 8\},\
5792: };
5793: static XPoint seg2 3104[] = {
5794:
                           \{1, -12\}, \{6, -12\}, \{6, 9\},\
5795: };
5796: static XPoint seg3 3104[] = {
5797:
                            \{4, -2\}, \{3, -4\}, \{1, -5\}, \{-1, -5\}, \{-4, -4\}, \{-6, -2\}, \{-7, 1\}, \{-7, 3\},
5798:
                            \{-6,6\},\{-4,8\},\{-1,9\},\{1,9\},\{3,8\},\{4,6\},
5799: };
5800: static XPoint seq4 3104[] = {
5801:
                           \{-5, -2\}, \{-6, 0\}, \{-6, 4\}, \{-5, 6\},
5802: };
5803: static XPoint seg5 3104[] = {
5804:
                           \{-1, -5\}, \{-3, -4\}, \{-4, -3\}, \{-5, 0\}, \{-5, 4\}, \{-4, 7\}, \{-3, 8\}, \{-1, 9\},
5805: };
5806: static XPoint seg6_3104[] = {
5807:
                           \{2, -12\}, \{4, -11\}, 
5808: };
5809: static XPoint seg7 3104[] = {
5810:
                            \{3, -12\}, \{4, -10\},\
5811: };
5812: static XPoint seg8 3104[] = {
5813:
                            \{6,7\},\{7,9\},\
5814: };
5815: static XPoint seg9 3104[] = {
5816:
                           \{6,8\},\{8,9\},\
5817: };
5818: static XPoint *char3104[] = {
5819:
                            seg0 3104, seg1 3104, seg2 3104, seg3 3104, seg4 3104, seg5 3104,
5820:
                            seg6_3104, seg7_3104, seg8_3104, seg9_3104,
5821:
                            NULL,
5822: };
5823: static int char p3104[] = {
                           XtNumber(seg0 3104), XtNumber(seg1 3104), XtNumber(seg2 3104),
5824:
5825:
                           XtNumber(seg3 3104), XtNumber(seg4 3104), XtNumber(seg5 3104),
5826:
                           XtNumber(seg6_3104),XtNumber(seg7_3104),XtNumber(seg8_3104),
```

```
Listing C.4 Continued
```

```
5827:
                                                               XtNumber(seg9 3104),
5828: };
5829: static XPoint seq0 3105[] = {
5830:
                                                               \{-5,1\},\{6,1\},\{6,-1\},\{5,-3\},\{4,-4\},\{1,-5\},
5831:
                                                                \{-1, -5\}, \{-4, -4\}, \{-6, -2\}, \{-7, 1\}, \{-7, 3\}, \{-6, 6\}, \{-4, 8\}, \{-1, 9\},
5832:
                                                                \{1,9\},\{4,8\},\{6,6\},
5833: };
5834: static XPoint seg1 3105[] = {
5835:
                                                               \{5,0\},\{5,-1\},\{4,-3\},\
5836: };
5837: static XPoint seg2_3105[] = {
5838:
                                                                \{-5, -2\}, \{-6, 0\}, \{-6, 4\}, \{-5, 6\},
5839: };
5840: static XPoint seg3 3105[] = {
5841:
                                                               \{4,1\},\{4,-2\},\{3,-4\},\{1,-5\},
5842: };
5843: static XPoint seg4 3105[] = {
5844:
                                                               \{-1, -5\}, \{-3, -4\}, \{-4, -3\}, \{-5, 0\}, \{-5, 4\}, \{-4, 7\}, \{-3, 8\}, \{-1, 9\},
5845: };
5846: static XPoint *char3105[] = {
5847:
                                                                seg0 3105, seg1 3105, seg2 3105, seg3 3105, seg4 3105,
                                                               NULL,
5848:
5849: };
5850: static int char p3105[] = {
5851:
                                                               XtNumber(seg0_3105),XtNumber(seg1_3105),XtNumber(seg2_3105),
5852:
                                                               XtNumber(seg3 3105), XtNumber(seg4 3105),
5853: };
5854: static XPoint seg0 3106[] = {
5855:
                                                                \{5, -10\}, \{5, -11\}, \{4, -11\}, \{4, -9\}, \{6, -9\}, \{6, -11\}, \{5, -12\}, \{6, -11\}, \{5, -12\}, \{6, -11\}, \{5, -12\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -11\}, \{6, -1
5856:
                                                                \{2, -12\}, \{0, -11\}, \{-1, -10\}, \{-2, -7\}, \{-2, 9\},
5857: };
5858: static XPoint seg1 3106[] = {
5859:
                                                                \{0, -10\}, \{-1, -7\}, \{-1, 8\},\
5860: };
5861: static XPoint seg2 3106[] = {
5862:
                                                               \{2, -12\}, \{1, -11\}, \{0, -9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\},
5863: };
5864: static XPoint seg3_3106[] = {
5865:
                                                              \{-5, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, \{4, -5\}, 
5866: };
5867: static XPoint seg4_3106[] = {
5868:
                                                               \{-5,9\},\{3,9\},
5869: };
5870: static XPoint seg5_3106[] = {
5871:
                                                                \{-2,8\},\{-4,9\},
5872: };
5873: static XPoint seg6 3106[] = {
5874:
                                                                \{-2,7\},\{-3,9\},
5875: };
5876: static XPoint seg7 3106[] = {
5877:
                                                               \{0,7\},\{1,9\},\
5878: };
```

```
5879: static XPoint seg8 3106[] = {
5880:
           \{0,8\},\{2,9\},\
5881: };
5882: static XPoint *char3106[] = {
5883:
           seg0_3106, seg1_3106, seg2_3106, seg3_3106, seg4_3106, seg5_3106,
5884:
           seg6 3106, seg7 3106, seg8 3106,
5885:
           NULL,
5886: };
5887: static int char p3106[] = {
5888:
           XtNumber(seg0_3106),XtNumber(seg1_3106),XtNumber(seg2_3106),
5889:
           XtNumber(seg3 3106), XtNumber(seg4 3106), XtNumber(seg5 3106),
5890:
           XtNumber(seg6 3106), XtNumber(seg7 3106), XtNumber(seg8 3106),
5891: };
5892: static XPoint seg0 3107[] = {
5893:
           \{6, -4\}, \{7, -3\}, \{8, -4\}, \{7, -5\}, \{6, -5\}, \{4, -4\},
5894:
           \{3, -3\},\
5895: };
5896: static XPoint seg1 3107[] = {
5897:
           \{-1, -5\}, \{-3, -4\}, \{-4, -3\}, \{-5, -1\}, \{-5, 1\}, \{-4, 3\}, \{-3, 4\}, \{-1, 5\},
5898:
           \{1,5\},\{3,4\},\{4,3\},\{5,1\},\{5,-1\},\{4,-3\},\{3,-4\},\{1,-5\},\{-1,-5\},
5899: };
5900: static XPoint seg2 3107[] = {
5901:
           \{-3, -3\}, \{-4, -1\}, \{-4, 1\}, \{-3, 3\},
5902: };
5903: static XPoint seg3 3107[] = {
5904:
           \{3,3\},\{4,1\},\{4,-1\},\{3,-3\},
5905: };
5906: static XPoint seg4 3107[] = {
5907:
           \{-1, -5\}, \{-2, -4\}, \{-3, -2\}, \{-3, 2\}, \{-2, 4\}, \{-1, 5\},
5908: };
5909: static XPoint seg5 3107[] = {
5910:
           \{1,5\},\{2,4\},\{3,2\},\{3,-2\},\{2,-4\},\{1,-5\},
5911: };
5912: static XPoint seg6 3107[] = {
5913:
           \{-4,3\},\{-5,4\},\{-6,6\},\{-6,7\},\{-5,9\},\{-4,10\},\{-1,11\},\{3,11\},
5914:
           \{6, 12\}, \{7, 13\},\
5915: };
5916: static XPoint seg7 3107[] = {
5917:
           \{-4,9\},\{-1,10\},\{3,10\},\{6,11\},
5918: };
5919: static XPoint seg8 3107[] = {
5920:
           \{-6,7\},\{-5,8\},\{-2,9\},\{3,9\},\{6,10\},\{7,12\},\{7,13\},\{6,15\},
5921:
           \{3,16\},\{-3,16\},\{-6,15\},\{-7,13\},\{-7,12\},\{-6,10\},\{-3,9\},
5922: };
5923: static XPoint seg9 3107[] = {
5924:
           \{-3, 16\}, \{-5, 15\}, \{-6, 13\}, \{-6, 12\}, \{-5, 10\}, \{-3, 9\},
5925: };
5926: static XPoint *char3107[] = {
5927:
           seg0_3107, seg1_3107, seg2_3107, seg3_3107, seg4_3107, seg5_3107,
5928:
           seg6 3107, seg7 3107, seg8 3107, seg9 3107,
5929:
           NULL,
5930: };
```

```
5931: static int char_p3107[] = {
5932:
                              XtNumber(seg0 3107), XtNumber(seg1 3107), XtNumber(seg2 3107),
5933:
                              XtNumber(seg3_3107),XtNumber(seg4_3107),XtNumber(seg5_3107),
5934:
                               XtNumber(seg6_3107), XtNumber(seg7_3107), XtNumber(seg8_3107),
5935:
                              XtNumber(seg9 3107),
5936: };
5937: static XPoint seg0 3108[] = {
5938:
                              \{-6, -12\}, \{-6, 9\},
5939: };
5940: static XPoint seg1 3108[] = {
5941:
                              \{-5, -11\}, \{-5, 8\},\
5942: };
5943: static XPoint seg2 3108[] = {
5944:
                               \{-9, -12\}, \{-4, -12\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9
5945: };
5946: static XPoint seg3 3108[] = {
5947:
                               \{-4, -1\}, \{-3, -3\}, \{-2, -4\}, \{0, -5\}, \{3, -5\}, \{5, -4\}, \{6, -3\}, \{7, 0\},
5948:
                               {7,9},
5949: };
5950: static XPoint seq4 3108[] = {
5951:
                               \{5, -3\}, \{6, 0\}, \{6, 8\},\
5952: };
5953: static XPoint seg5 3108[] = {
5954:
                              \{3, -5\}, \{4, -4\}, \{5, -1\}, \{5, 9\},\
5955: };
5956: static XPoint seg6 3108[] = {
5957:
                              \{-9,9\},\{-1,9\},
5958: };
5959: static XPoint seg7 3108[] = {
5960:
                              \{2,9\},\{10,9\},\
5961: };
5962: static XPoint seg8 3108[] = {
5963:
                              \{-8, -12\}, \{-6, -11\},
5964: };
5965: static XPoint seg9 3108[] = {
5966:
                              \{-7, -12\}, \{-6, -10\},\
5967: };
5968: static XPoint seg10 3108[] = {
5969:
                              \{-6,8\},\{-8,9\},
5970: };
5971: static XPoint seg11 3108[] = {
5972:
                               \{-6,7\},\{-7,9\},
5973: };
5974: static XPoint seg12 3108[] = {
5975:
                              \{-4,7\},\{-3,9\},
5976: };
5977: static XPoint seg13_3108[] = {
5978:
                               \{-4,8\},\{-2,9\},\
5979: };
```

```
5980: static XPoint seg14 3108[] = {
5981:
                                           \{5,8\},\{3,9\},\
5982: };
5983: static XPoint seg15 3108[] = {
5984:
                                           \{5,7\},\{4,9\},\
5985: };
5986: static XPoint seg16 3108[] = {
5987:
                                           \{7,7\},\{8,9\},
5988: };
5989: static XPoint seg17_3108[] = {
5990:
                                           \{7,8\},\{9,9\},\
5991: };
5992: static XPoint *char3108[] = {
5993:
                                           seg0 3108, seg1 3108, seg2 3108, seg3 3108, seg4 3108,
5994:
                                           seg5 3108, seg6 3108, seg7 3108, seg8 3108, seg9 3108,
5995:
                                           seg10 3108, seg11 3108, seg12 3108, seg13 3108, seg14 3108,
5996:
                                           seg15_3108, seg16_3108, seg17_3108,
5997:
                                          NULL,
5998: };
5999: static int char p3108[] = {
6000:
                                          XtNumber(seg0_3108),XtNumber(seg1_3108),XtNumber(seg2_3108),
6001:
                                           XtNumber(seg3 3108), XtNumber(seg4 3108), XtNumber(seg5 3108),
6002:
                                          XtNumber(seg6_3108), XtNumber(seg7_3108), XtNumber(seg8_3108),
6003:
                                          XtNumber(seg9_3108), XtNumber(seg10_3108), XtNumber(seg11_3108),
6004:
                                          XtNumber(seg12 3108), XtNumber(seg13 3108), XtNumber(seg14 3108),
6005:
                                          XtNumber(seg15_3108), XtNumber(seg16_3108), XtNumber(seg17_3108),
6006: };
6007: static XPoint seg0 3109[] = {
6008:
                                           \{-1, -12\}, \{-1, -10\}, \{1, -10\}, \{1, -12\}, \{-1, -12\}, 
6009: };
6010: static XPoint seg1 3109[] = {
6011:
                                           \{0, -12\}, \{0, -10\},\
6012: };
6013: static XPoint seg2 3109[] = {
6014:
                                          \{-1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1, -11\}, \{1,
6015: };
6016: static XPoint seg3 3109[] = {
6017:
                                          \{-1, -5\}, \{-1, 9\},\
6018: };
6019: static XPoint seg4 3109[] = {
                                          \{0, -4\}, \{0, 8\},\
6020:
6021: };
6022: static XPoint seg5 3109[] = {
6023:
                                           \{-4, -5\}, \{1, -5\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{
6024: };
6025: static XPoint seg6 3109[] = {
6026:
                                           \{-4,9\},\{4,9\},
6027: };
6028: static XPoint seg7_3109[] = {
6029:
                                           \{-3, -5\}, \{-1, -4\},
6030: };
6031: static XPoint seg8 3109[] = {
6032:
                                           \{-2, -5\}, \{-1, -3\},
6033: };
```

```
Listing C.4 Continued
```

```
6034: static XPoint seg9 3109[] = {
6035:
                                                              \{-1,8\},\{-3,9\},
6036: };
6037: static XPoint seg10 3109[] = {
6038:
                                                             \{-1,7\},\{-2,9\},
6039: };
6040: static XPoint seg11 3109[] = {
6041:
                                                             \{1,7\},\{2,9\},\
6042: };
6043: static XPoint seg12 3109[] = {
6044:
                                                             \{1,8\},\{3,9\},\
6045: };
6046: static XPoint *char3109[] = {
                                                              seg0 3109, seg1 3109, seg2 3109, seg3 3109, seg4 3109,
6047:
6048:
                                                              seg5_3109, seg6_3109, seg7_3109, seg8_3109, seg9_3109,
6049:
                                                              seg10 3109, seg11 3109, seg12 3109,
6050:
                                                             NULL,
6051: };
6052: static int char p3109[] = {
6053:
                                                             XtNumber(seg0_3109),XtNumber(seg1_3109),XtNumber(seg2_3109),
6054:
                                                              XtNumber(seg3_3109),XtNumber(seg4_3109),XtNumber(seg5_3109),
6055:
                                                             XtNumber(seg6_3109), XtNumber(seg7_3109), XtNumber(seg8_3109),
6056:
                                                             XtNumber(seg9 3109), XtNumber(seg10 3109), XtNumber(seg11 3109),
6057:
                                                             XtNumber(seg12_3109),
6058: };
6059: static XPoint seg0 3110[] = {
6060:
                                                              \{0, -12\}, \{0, -10\}, \{2, -10\}, \{2, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, -12\}, \{0, 
6061: };
6062: static XPoint seg1 3110[] = {
6063:
                                                             \{1, -12\}, \{1, -10\}, 
6064: };
6065: static XPoint seg2 3110[] = {
6066:
                                                             \{0, -11\}, \{2, -11\}, 
6067: };
6068: static XPoint seg3 3110[] = {
6069:
                                                             \{0, -5\}, \{0, 12\}, \{-1, 15\}, \{-2, 16\},
6070: };
6071: static XPoint seg4_3110[] = {
6072:
                                                             \{1, -4\}, \{1, 11\}, \{0, 14\}, \{0, 14\}, \{0, 14\}, \{0, 14\}, \{0, 14\}, \{0, 14\}, \{0, 14\}, \{0, 14\}, \{0, 14\}, \{0, 14\}, \{0, 14\}, \{0, 14\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{1, 11\}, \{
6073: };
6074: static XPoint seg5 3110[] = {
6075:
                                                              \{-3, -5\}, \{2, -5\}, \{2, 11\}, \{1, 14\}, \{0, 15\}, \{-2, 16\}, \{-5, 16\}, \{-6, 15\}, \{-3, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}
6076:
                                                              \{-6, 13\}, \{-4, 13\}, \{-4, 15\}, \{-5, 15\}, \{-5, 14\},
6077: };
6078: static XPoint seg6_3110[] = {
6079:
                                                             \{-2, -5\}, \{0, -4\}, 
6080: };
6081: static XPoint seg7 3110[] = {
6082:
                                                              \{-1, -5\}, \{0, -3\},\
6083: };
```

```
6084: static XPoint *char3110[] = {
                              seg0_3110, seg1_3110, seg2_3110, seg3_3110, seg4_3110, seg5_3110,
6085:
6086:
                              seg6 3110, seg7 3110,
6087:
                              NULL,
6088: };
6089: static int char p3110[] = {
6090:
                              XtNumber(seg0 3110), XtNumber(seg1 3110), XtNumber(seg2 3110),
6091:
                              XtNumber(seg3 3110), XtNumber(seg4 3110), XtNumber(seg5 3110),
6092:
                              XtNumber(seg6 3110),XtNumber(seg7 3110),
6093: };
6094: static XPoint seg0 3111[] = {
6095:
                              \{-6, -12\}, \{-6, 9\},
6096: };
6097: static XPoint seg1 3111[] = {
6098:
                              \{-5, -11\}, \{-5, 8\},\
6099: };
6100: static XPoint seg2 3111[] = {
6101:
                              \{-9, -12\}, \{-4, -12\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9
6102: };
6103: static XPoint seg3 3111[] = {
6104:
                              \{5, -4\}, \{-4, 5\},\
6105: };
6106: static XPoint seg4 3111[] = {
6107:
                              \{0,1\},\{7,9\},\
6108: };
6109: static XPoint seg5 3111[] = {
6110:
                             \{0,2\},\{6,9\},\
6111: };
6112: static XPoint seg6 3111[] = {
6113:
                              \{-1,2\},\{5,9\},\
6114: };
6115: static XPoint seg7_3111[] = {
6116:
                              \{2, -5\}, \{9, -5\},\
6117: };
6118: static XPoint seg8_3111[] = {
6119:
                              \{-9,9\},\{-1,9\},
6120: };
6121: static XPoint seg9 3111[] = {
6122:
                              \{2,9\},\{9,9\},\
6123: };
6124: static XPoint seg10 3111[] = {
6125:
                              \{-8, -12\}, \{-6, -11\},
6126: };
6127: static XPoint seg11 3111[] = {
6128:
                              \{-7, -12\}, \{-6, -10\},\
6129: };
6130: static XPoint seg12_3111[] = {
6131:
                             \{3, -5\}, \{5, -4\},\
6132: };
6133: static XPoint seg13 3111[] = {
6134:
                              \{8, -5\}, \{5, -4\},\
6135: };
```



```
Listing C.4 Continued
```

```
6136: static XPoint seg14_3111[] = {
6137:
                           \{-6,8\},\{-8,9\},
6138: };
6139: static XPoint seg15 3111[] = {
6140:
                           \{-6,7\},\{-7,9\},\
6141: };
6142: static XPoint seg16_3111[] = {
6143:
                           \{-4,7\},\{-3,9\},
6144: };
6145: static XPoint seg17 3111[] = {
6146:
                          \{-4,8\},\{-2,9\},
6147: };
6148: static XPoint seg18 3111[] = {
                           \{5,7\},\{3,9\},
6149:
6150: };
6151: static XPoint seg19 3111[] = {
6152:
                           \{4,7\},\{8,9\},\
6153: };
6154: static XPoint *char3111[] = {
6155:
                           seg0 3111, seg1 3111, seg2 3111, seg3 3111, seg4 3111,
6156:
                           seg5_3111, seg6_3111, seg7_3111, seg8_3111, seg9_3111,
                           seg10 3111, seg11 3111, seg12 3111, seg13 3111, seg14 3111,
6157:
                           seg15_3111, seg16_3111, seg17_3111, seg18_3111, seg19_3111,
6158:
6159:
                           NULL,
6160: };
6161: static int char p3111[] = {
6162:
                           XtNumber(seg0_3111),XtNumber(seg1_3111),XtNumber(seg2_3111),
6163:
                           XtNumber(seg3 3111),XtNumber(seg4 3111),XtNumber(seg5 3111),
6164:
                           XtNumber(seg6 3111), XtNumber(seg7 3111), XtNumber(seg8 3111),
6165:
                           XtNumber(seg9 3111), XtNumber(seg10 3111), XtNumber(seg11 3111),
                           XtNumber(seg12 3111), XtNumber(seg13 3111), XtNumber(seg14 3111),
6166:
6167:
                           XtNumber(seg15 3111), XtNumber(seg16 3111), XtNumber(seg17 3111),
6168:
                           XtNumber(seg18 3111), XtNumber(seg19 3111),
6169: };
6170: static XPoint seg0 3112[] = {
6171:
                           \{-1, -12\}, \{-1, 9\},
6172: };
6173: static XPoint seg1 3112[] = {
6174:
                          \{0, -11\}, \{0, 8\},\
6175: };
6176: static XPoint seg2 3112[] = {
6177:
                           \{-4, -12\}, \{1, -12\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\}, \{1, 9\},
6178: };
6179: static XPoint seg3 3112[] = {
6180:
                           \{-4,9\},\{4,9\},
6181: };
6182: static XPoint seg4 3112[] = {
6183:
                           \{-3, -12\}, \{-1, -11\},
6184: };
6185: static XPoint seg5 3112[] = {
6186:
                           \{-2, -12\}, \{-1, -10\},
6187: };
```

```
6188: static XPoint seg6 3112[] = {
6189:
                            \{-1,8\},\{-3,9\},
6190: };
6191: static XPoint seg7 3112[] = {
6192:
                            \{-1,7\},\{-2,9\},
6193: };
6194: static XPoint seg8 3112[] = {
6195:
                            \{1,7\},\{2,9\},\
6196: };
6197: static XPoint seg9 3112[] = {
6198:
                            \{1,8\},\{3,9\},\
6199: };
6200: static XPoint *char3112[] = {
6201:
                            seg0 3112, seg1 3112, seg2 3112, seg3 3112, seg4 3112, seg5 3112,
6202:
                            seg6 3112, seg7 3112, seg8 3112, seg9 3112,
6203:
                            NULL,
6204: };
6205: static int char p3112[] = {
6206:
                            XtNumber(seg0_3112),XtNumber(seg1_3112),XtNumber(seg2_3112),
6207:
                            XtNumber(seg3_3112),XtNumber(seg4_3112),XtNumber(seg5_3112),
6208:
                            XtNumber(seg6 3112), XtNumber(seg7 3112), XtNumber(seg8 3112),
6209:
                            XtNumber(seq9 3112),
6210: };
6211: static XPoint seg0 3113[] = {
6212:
                            \{-12, -5\}, \{-12, 9\}, 
6213: };
6214: static XPoint seq1 3113[] = {
6215:
                            \{-11, -4\}, \{-11, 8\},
6216: };
6217: static XPoint seg2 3113[] = {
6218:
                            \{-15, -5\}, \{-10, -5\}, \{-10, 9\}, 
6219: };
6220: static XPoint seg3 3113[] = {
6221:
                            \{-10, -1\}, \{-9, -3\}, \{-8, -4\}, \{-6, -5\}, \{-3, -5\}, \{-1, -4\}, \{0, -3\}, \{-1, -4\}, \{0, -3\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, 
6222:
                            \{1,0\},\{1,9\},\
6223: };
6224: static XPoint seg4 3113[] = {
6225:
                            \{-1, -3\}, \{0, 0\}, \{0, 8\},
6226: };
6227: static XPoint seg5_3113[] = {
6228:
                            \{-3, -5\}, \{-2, -4\}, \{-1, -1\}, \{-1, 9\},
6229: };
6230: static XPoint seg6 3113[] = {
6231:
                            \{1, -1\}, \{2, -3\}, \{3, -4\}, \{5, -5\}, \{8, -5\}, \{10, -4\}, \{11, -3\}, \{12, 0\},
6232:
                            {12,9},
6233: };
6234: static XPoint seg7_3113[] = {
6235:
                            \{10, -3\}, \{11, 0\}, \{11, 8\}, \}
6236: };
6237: static XPoint seg8 3113[] = {
6238:
                            \{8, -5\}, \{9, -4\}, \{10, -1\}, \{10, 9\},\
6239: };
```

```
Listing C.4 Continued
```

```
6240: static XPoint seg9 3113[] = {
6241:
                               \{-15,9\},\{-7,9\},
6242: };
6243: static XPoint seg10 3113[] = {
6244:
                               \{-4,9\},\{4,9\},
6245: };
6246: static XPoint seg11 3113[] = {
6247:
                               {7,9},{15,9},
6248: };
6249: static XPoint seg12 3113[] = {
6250:
                               \{-14, -5\}, \{-12, -4\},
6251: };
6252: static XPoint seg13 3113[] = {
6253:
                                \{-13, -5\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}, \{-12, -3\}
6254: };
6255: static XPoint seg14 3113[] = {
6256:
                                \{-12,8\},\{-14,9\},
6257: };
6258: static XPoint seg15_3113[] = {
6259:
                                \{-12,7\},\{-13,9\},
6260: };
6261: static XPoint seg16 3113[] = {
6262:
                               \{-10,7\},\{-9,9\},
6263: };
6264: static XPoint seg17 3113[] = {
6265:
                               \{-10,8\},\{-8,9\},
6266: };
6267: static XPoint seg18 3113[] = {
6268:
                               \{-1,8\},\{-3,9\},
6269: };
6270: static XPoint seg19 3113[] = {
6271:
                               \{-1,7\},\{-2,9\},
6272: };
6273: static XPoint seg20 3113[] = {
6274:
                                \{1,7\},\{2,9\},\
6275: };
6276: static XPoint seg21 3113[] = {
6277:
                                \{1,8\},\{3,9\},\
6278: };
6279: static XPoint seg22 3113[] = {
6280:
                                \{10,8\},\{8,9\},\
6281: };
6282: static XPoint seg23 3113[] = {
6283:
                               \{10,7\},\{9,9\},\
6284: };
6285: static XPoint seg24 3113[] = {
6286:
                               {12,7},{13,9},
6287: };
6288: static XPoint seg25_3113[] = {
6289:
                                \{12,8\},\{14,9\},\
6290: };
```

```
6291: static XPoint *char3113[] = {
6292:
                         seg0_3113, seg1_3113, seg2_3113, seg3_3113, seg4_3113,
6293:
                         seg5 3113, seg6 3113, seg7 3113, seg8 3113, seg9 3113,
6294:
                         seg10 3113, seg11 3113, seg12 3113, seg13 3113, seg14 3113,
                         seg15_3113, seg16_3113, seg17_3113, seg18_3113, seg19_3113,
6295:
6296:
                         seg20 3113, seg21 3113, seg22 3113, seg23 3113, seg24 3113,
6297:
                         seg25 3113,
6298:
                        NULL,
6299: };
6300: static int char p3113[] = {
6301:
                        XtNumber(seg0 3113), XtNumber(seg1 3113), XtNumber(seg2 3113),
6302:
                         XtNumber(seg3 3113),XtNumber(seg4 3113),XtNumber(seg5 3113),
6303:
                         XtNumber(seg6 3113),XtNumber(seg7 3113),XtNumber(seg8 3113),
6304:
                        XtNumber(seg9 3113), XtNumber(seg10 3113), XtNumber(seg11 3113),
                         XtNumber(seg12 3113), XtNumber(seg13 3113), XtNumber(seg14 3113),
6305:
6306:
                        XtNumber(seg15 3113), XtNumber(seg16 3113), XtNumber(seg17 3113),
6307:
                        XtNumber(seg18_3113), XtNumber(seg19_3113), XtNumber(seg20_3113),
6308:
                         XtNumber(seg21 3113), XtNumber(seg22 3113), XtNumber(seg23 3113),
6309:
                        XtNumber(seg24_3113),XtNumber(seg25_3113),
6310: };
6311: static XPoint seg0 3114[] = {
6312:
                        \{-6, -5\}, \{-6, 9\},\
6313: };
6314: static XPoint seg1 3114[] = {
6315:
                        \{-5, -4\}, \{-5, 8\},
6316: };
6317: static XPoint seg2 3114[] = {
6318:
                        \{-9, -5\}, \{-4, -5\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\}, \{-4, 9\},
6319: };
6320: static XPoint seg3_3114[] = {
6321:
                         \{-4, -1\}, \{-3, -3\}, \{-2, -4\}, \{0, -5\}, \{3, -5\}, \{5, -4\}, \{6, -3\}, \{7, 0\},
6322:
                         {7,9},
6323: };
6324: static XPoint seg4 3114[] = {
6325:
                        \{5, -3\}, \{6, 0\}, \{6, 8\},\
6326: };
6327: static XPoint seg5 3114[] = {
6328:
                        \{3, -5\}, \{4, -4\}, \{5, -1\}, \{5, 9\},\
6329: };
6330: static XPoint seg6 3114[] = {
6331:
                        \{-9,9\},\{-1,9\},
6332: };
6333: static XPoint seg7 3114[] = {
6334:
                         \{2,9\},\{10,9\},\
6335: };
6336: static XPoint seg8 3114[] = {
6337:
                         \{-8, -5\}, \{-6, -4\},\
6338: };
6339: static XPoint seg9_3114[] = {
6340:
                        \{-7, -5\}, \{-6, -3\}, 
6341: };
6342: static XPoint seg10 3114[] = {
6343:
                         \{-6,8\},\{-8,9\},
6344: };
```

```
Listing C.4 Continued
```

```
6345: static XPoint seg11_3114[] = {
6346:
           \{-6,7\},\{-7,9\},\
6347: };
6348: static XPoint seg12 3114[] = {
6349:
           \{-4,7\},\{-3,9\},
6350: };
6351: static XPoint seg13 3114[] = {
6352:
           \{-4,8\},\{-2,9\},\
6353: };
6354: static XPoint seg14 3114[] = {
6355:
           \{5,8\},\{3,9\},\
6356: };
6357: static XPoint seg15 3114[] = {
6358:
           \{5,7\},\{4,9\},\
6359: };
6360: static XPoint seg16 3114[] = {
6361:
           \{7,7\},\{8,9\},\
6362: };
6363: static XPoint seg17 3114[] = {
6364:
           \{7,8\},\{9,9\},\
6365: };
6366: static XPoint *char3114[] = {
6367:
           seg0 3114, seg1 3114, seg2 3114, seg3 3114, seg4 3114,
6368:
           seg5_3114, seg6_3114, seg7_3114, seg8_3114, seg9_3114,
6369:
           seg10 3114, seg11 3114, seg12 3114, seg13 3114,
6370:
           seg14 3114, seg15 3114, seg16 3114, seg17 3114,
6371:
           NULL,
6372: };
6373: static int char p3114[] = {
6374:
           XtNumber(seg0_3114),XtNumber(seg1_3114),XtNumber(seg2_3114),
6375:
           XtNumber(seg3 3114), XtNumber(seg4 3114), XtNumber(seg5 3114),
6376:
           XtNumber(seg6_3114), XtNumber(seg7_3114), XtNumber(seg8_3114),
6377:
           XtNumber(seg9_3114), XtNumber(seg10_3114), XtNumber(seg11_3114),
6378:
           XtNumber(seg12 3114), XtNumber(seg13 3114), XtNumber(seg14 3114),
6379:
           XtNumber(seg15_3114), XtNumber(seg16_3114), XtNumber(seg17_3114),
6380: };
6381: static XPoint seg0 3115[] = {
6382:
           \{-1, -5\}, \{-4, -4\}, \{-6, -2\}, \{-7, 1\}, \{-7, 3\}, \{-6, 6\},
6383:
           \{-4,8\},\{-1,9\},\{1,9\},\{4,8\},\{6,6\},\{7,3\},\{7,1\},\{6,-2\},\{4,-4\},
6384:
           \{1, -5\}, \{-1, -5\}, 
6385: };
6386: static XPoint seg1 3115[] = {
6387:
           \{-5, -2\}, \{-6, 0\}, \{-6, 4\}, \{-5, 6\},
6388: };
6389: static XPoint seg2_3115[] = {
6390:
           \{5,6\},\{6,4\},\{6,0\},\{5,-2\},\
6391: };
6392: static XPoint seg3 3115[] = {
6393:
           \{-1, -5\}, \{-3, -4\}, \{-4, -3\}, \{-5, 0\}, \{-5, 4\}, \{-4, 7\}, \{-3, 8\}, \{-1, 9\},
6394: };
```

```
6395: static XPoint seg4 3115[] = {
6396:
                             \{1,9\},\{3,8\},\{4,7\},\{5,4\},\{5,0\},\{4,-3\},\{3,-4\},\{1,-5\},
6397: };
6398: static XPoint *char3115[] = {
                             seg0_3115, seg1_3115, seg2_3115, seg3_3115, seg4_3115,
6399:
6400:
                             NULL,
6401: };
6402: static int char p3115[] = {
6403:
                             XtNumber(seg0 3115), XtNumber(seg1 3115), XtNumber(seg2 3115),
6404:
                             XtNumber(seg3 3115), XtNumber(seg4 3115),
6405: };
6406: static XPoint seg0 3116[] = {
6407:
                            \{-6, -5\}, \{-6, 16\},
6408: };
6409: static XPoint seg1 3116[] = {
6410:
                             \{-5, -4\}, \{-5, 15\},
6411: };
6412: static XPoint seg2 3116[] = {
6413:
                             \{-9, -5\}, \{-4, -5\}, \{-4, 16\},
6414: };
6415: static XPoint seg3 3116[] = {
6416:
                             \{-4, -2\}, \{-3, -4\}, \{-1, -5\}, \{1, -5\}, \{4, -4\}, \{6, -2\}, \{7, 1\}, \{7, 3\},
6417:
                             \{6,6\},\{4,8\},\{1,9\},\{-1,9\},\{-3,8\},\{-4,6\},
6418: };
6419: static XPoint seg4 3116[] = {
6420:
                             \{5, -2\}, \{6, 0\}, \{6, 4\}, \{5, 6\},\
6421: };
6422: static XPoint seg5 3116[] = {
                             \{1, -5\}, \{3, -4\}, \{4, -3\}, \{5, 0\}, \{5, 4\}, \{4, 7\}, \{3, 8\}, \{1, 9\},
6423:
6424: };
6425: static XPoint seg6 3116[] = {
6426:
                             \{-9, 16\}, \{-1, 16\},
6427: };
6428: static XPoint seg7 3116[] = {
6429:
                            \{-8, -5\}, \{-6, -4\},
6430: };
6431: static XPoint seg8 3116[] = {
6432:
                            \{-7, -5\}, \{-6, -3\},\
6433: };
6434: static XPoint seg9 3116[] = {
6435:
                            \{-6, 15\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-8,
6436: };
6437: static XPoint seg10 3116[] = {
6438:
                             {-6,14}<i>,{-7,16},
6439: };
6440: static XPoint seg11 3116[] = {
6441:
                             \{-4, 14\}, \{-3, 16\},\
6442: };
6443: static XPoint seg12_3116[] = {
6444:
                             \{-4, 15\}, \{-2, 16\},
6445: };
```

```
6446: static XPoint *char3116[] = {
6447:
           seg0 3116, seg1 3116, seg2 3116, seg3 3116, seg4 3116,
6448:
           seg5 3116, seg6 3116, seg7 3116, seg8 3116, seg9 3116,
6449:
           seg10_3116, seg11_3116,
6450:
           seg12 3116,
6451:
           NULL,
6452: };
6453: static int char p3116[] = {
           XtNumber(seg0 3116), XtNumber(seg1 3116), XtNumber(seg2 3116),
6454:
6455:
           XtNumber(seg3_3116),XtNumber(seg4_3116),XtNumber(seg5_3116),
6456:
           XtNumber(seg6_3116), XtNumber(seg7_3116), XtNumber(seg8_3116),
6457:
           XtNumber(seg9 3116), XtNumber(seg10 3116), XtNumber(seg11 3116),
6458:
           XtNumber(seg12 3116),
6459: };
6460: static XPoint seg0 3117[] = {
6461:
           \{4, -4\}, \{4, 16\},\
6462: };
6463: static XPoint seg1 3117[] = {
6464:
           \{5, -3\}, \{5, 15\},\
6465: };
6466: static XPoint seg2 3117[] = {
6467:
           \{3, -4\}, \{5, -4\}, \{6, -5\}, \{6, 16\},\
6468: };
6469: static XPoint seg3 3117[] = {
6470:
           \{4, -2\}, \{3, -4\}, \{1, -5\}, \{-1, -5\}, \{-4, -4\}, \{-6, -2\}, \{-7, 1\}, \{-7, 3\},
6471:
           \{-6,6\},\{-4,8\},\{-1,9\},\{1,9\},\{3,8\},\{4,6\},
6472: };
6473: static XPoint seg4 3117[] = {
6474:
           \{-5, -2\}, \{-6, 0\}, \{-6, 4\}, \{-5, 6\},
6475: };
6476: static XPoint seg5 3117[] = {
6477:
           \{-1, -5\}, \{-3, -4\}, \{-4, -3\}, \{-5, 0\}, \{-5, 4\}, \{-4, 7\}, \{-3, 8\}, \{-1, 9\},
6478: };
6479: static XPoint seg6_3117[] = {
6480:
           \{1, 16\}, \{9, 16\}, 
6481: };
6482: static XPoint seg7 3117[] = {
6483:
           {4,15},{2,16},
6484: };
6485: static XPoint seg8 3117[] = {
6486:
           \{4, 14\}, \{3, 16\},\
6487: };
6488: static XPoint seg9 3117[] = {
6489:
           \{6,14\},\{7,16\},\
6490: };
6491: static XPoint seg10 3117[] = {
6492:
           \{6, 15\}, \{8, 16\},\
6493: };
6494: static XPoint *char3117[] = {
6495:
           seg0 3117, seg1 3117, seg2 3117, seg3 3117, seg4 3117, seg5 3117,
6496:
           seg6 3117, seg7 3117, seg8 3117, seg9 3117, seg10 3117,
```

```
6497:
                          NULL,
6498: };
6499: static int char p3117[] = {
6500:
                          XtNumber(seg0 3117), XtNumber(seg1 3117), XtNumber(seg2 3117),
6501:
                          XtNumber(seg3_3117),XtNumber(seg4_3117),XtNumber(seg5_3117),
6502:
                           XtNumber(seg6_3117),XtNumber(seg7_3117),XtNumber(seg8_3117),
6503:
                          XtNumber(seg9 3117), XtNumber(seg10 3117),
6504: };
6505: static XPoint seg0 3118[] = {
6506:
                          \{-4, -5\}, \{-4, 9\},\
6507: };
6508: static XPoint seg1 3118[] = {
6509:
                          \{-3, -4\}, \{-3, 8\},
6510: };
6511: static XPoint seg2 3118[] = {
6512:
                           \{-7, -5\}, \{-2, -5\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\},
6513: };
6514: static XPoint seg3 3118[] = {
6515:
                           \{5, -3\}, \{5, -4\}, \{4, -4\}, \{4, -2\}, \{6, -2\}, \{6, -4\}, \{5, -5\}, \{3, -5\},
6516:
                           \{1, -4\}, \{-1, -2\}, \{-2, 1\},\
6517: };
6518: static XPoint seq4 3118[] = {
6519:
                           \{-7,9\},\{1,9\},
6520: };
6521: static XPoint seg5 3118[] = {
6522:
                          \{-6, -5\}, \{-4, -4\},
6523: };
6524: static XPoint seg6 3118[] = {
6525:
                          \{-5, -5\}, \{-4, -3\},
6526: };
6527: static XPoint seg7 3118[] = {
6528:
                          \{-4,8\},\{-6,9\},\
6529: };
6530: static XPoint seg8 3118[] = {
6531:
                          \{-4,7\},\{-5,9\},
6532: };
6533: static XPoint seg9 3118[] = {
6534:
                          \{-2,7\},\{-1,9\},
6535: };
6536: static XPoint seg10 3118[] = {
6537:
                          \{-2,8\},\{0,9\},
6538: };
6539: static XPoint *char3118[] = {
                           seg0 3118, seg1 3118, seg2 3118, seg3 3118, seg4 3118, seg5 3118,
6540:
                           seg6 3118, seg7 3118, seg8 3118, seg9 3118, seg10 3118,
6541:
6542:
                          NULL,
6543: };
6544: static int char p3118[] = {
6545:
                          XtNumber(seg0_3118),XtNumber(seg1_3118),XtNumber(seg2_3118),
6546:
                          XtNumber(seg3 3118), XtNumber(seg4 3118), XtNumber(seg5 3118),
6547:
                          XtNumber(seg6 3118), XtNumber(seg7 3118), XtNumber(seg8 3118),
6548:
                          XtNumber(seg9 3118), XtNumber(seg10 3118),
6549: };
```

```
6550: static XPoint seg0 3119[] = {
6551:
                                                              \{5, -3\}, \{6, -5\}, \{6, -1\}, \{5, -3\}, \{4, -4\}, \{2, -5\}, \{-2, -5\},
6552:
                                                              \{-4, -4\}, \{-5, -3\}, \{-5, -1\}, \{-4, 1\}, \{-2, 2\}, \{3, 3\}, \{-4, -4\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}
6553:
                                                             \{5,4\},\{6,7\},
6554: };
6555: static XPoint seg1_3119[] = {
6556:
                                                              \{-4, -4\}, \{-5, -1\},\
6557: };
6558: static XPoint seg2 3119[] = {
6559:
                                                              \{-4,0\},\{-2,1\},\{3,2\},\{5,3\},
6560: };
6561: static XPoint seg3 3119[] = {
6562:
                                                             \{6,4\},\{5,8\},\
6563: };
6564: static XPoint seg4 3119[] = {
6565:
                                                              \{-5, -3\}, \{-4, -1\}, \{-2, 0\}, \{3, 1\}, \{5, 2\}, \{6, 4\}, \{6, 7\}, \{5, 8\},
6566:
                                                              \{3,9\},\{-1,9\},\{-3,8\},\{-4,7\},\{-5,5\},\{-5,9\},\{-4,7\},
6567: };
6568: static XPoint *char3119[] = {
                                                             seg0 3119, seg1 3119, seg2 3119, seg3 3119, seg4 3119,
6569:
6570:
                                                             NULL,
6571: };
6572: static int char p3119[] = {
                                                             XtNumber(seg0 3119), XtNumber(seg1 3119), XtNumber(seg2 3119),
6573:
                                                             XtNumber(seg3_3119),XtNumber(seg4_3119),
6574:
6575: };
6576: static XPoint seg0 3120[] = {
6577:
                                                              \{-2, -10\}, \{-2, 4\}, 
6578:
                                                              \{-1,7\},\{0,8\},\{2,9\},\{4,9\},\{6,8\},\{7,6\},
6579: };
6580: static XPoint seg1 3120[] = {
6581:
                                                            \{-1, -10\}, \{-1, 5\}, \{0, 7\},
6582: };
6583: static XPoint seg2_3120[] = {
6584:
                                                              \{-2, -10\}, \{0, -12\}, \{0, 5\}, \{1, 8\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{2, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\}, \{3, 9\},
6585: };
6586: static XPoint seg3 3120[] = {
6587:
                                                              \{-5, -5\}, \{4, -5\}, 
6588: };
6589: static XPoint *char3120[] = {
6590:
                                                              seg0 3120, seg1 3120, seg2 3120, seg3 3120,
6591:
                                                             NULL,
6592: };
6593: static int char p3120[] = {
                                                             XtNumber(seg0 3120), XtNumber(seg1 3120), XtNumber(seg2 3120),
6594:
6595:
                                                             XtNumber(seg3 3120),
6596: };
6597: static XPoint seg0 3121[] = {
6598:
                                                              \{-6, -5\}, \{-6, 4\}, \{-5, 7\}, \{-4, 8\}, \{-2, 9\}, \{1, 9\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, \{-6, -5\}, 
6599:
                                                              \{3,8\},\{4,7\},\{5,5\},
6600: };
```

```
6601: static XPoint seg1 3121[] = {
6602:
                                                 \{-5, -4\}, \{-5, 5\}, \{-4, 7\},
6603: };
6604: static XPoint seg2 3121[] = {
6605:
                                                \{-9, -5\}, \{-4, -5\}, \{-4, 5\}, \{-3, 8\}, \{-2, 9\},
6606: };
6607: static XPoint seg3 3121[] = {
6608:
                                                \{5, -5\}, \{5, 9\}, \{10, 9\},\
6609: };
6610: static XPoint seg4_3121[] = {
6611:
                                                \{6, -4\}, \{6, 8\},\
6612: };
6613: static XPoint seg5 3121[] = {
6614:
                                                \{2, -5\}, \{7, -5\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7, 9\}, \{7
6615: };
6616: static XPoint seg6 3121[] = {
6617:
                                               \{-8, -5\}, \{-6, -4\},
6618: };
6619: static XPoint seg7 3121[] = {
6620:
                                              \{-7, -5\}, \{-6, -3\},
6621: };
6622: static XPoint seg8 3121[] = {
6623:
                                                \{7,7\},\{8,9\},\
6624: };
6625: static XPoint seg9 3121[] = {
6626:
                                                \{7,8\},\{9,9\},\
6627: };
6628: static XPoint *char3121[] = {
                                                 seg0 3121, seg1 3121, seg2 3121, seg3 3121, seg4 3121, seg5 3121,
6629:
                                                 seg6 3121, seg7 3121, seg8 3121, seg9 3121,
6630:
6631:
                                                NULL,
6632: };
6633: static int char p3121[] = {
6634:
                                                XtNumber(seg0 3121), XtNumber(seg1 3121), XtNumber(seg2 3121),
6635:
                                                XtNumber(seg3_3121),XtNumber(seg4_3121),XtNumber(seg5_3121),
6636:
                                                XtNumber(seg6 3121), XtNumber(seg7 3121), XtNumber(seg8 3121),
6637:
                                                XtNumber(seg9 3121),
6638: };
6639: static XPoint seg0 3122[] = {
6640:
                                                 \{-6, -5\}, \{0, 9\},\
6641: };
6642: static XPoint seg1 3122[] = {
6643:
                                                \{-5, -5\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0, 7\}, \{0
6644: };
6645: static XPoint seg2_3122[] = {
6646:
                                                \{-4, -5\}, \{1, 7\}, 
6647: };
6648: static XPoint seg3 3122[] = {
6649:
                                                \{6, -4\}, \{1, 7\}, \{0, 9\},\
6650: };
6651: static XPoint seg4 3122[] = {
6652:
                                                 \{-8, -5\}, \{-1, -5\},
6653: };
```

```
Listing C.4 Continued
```

```
6654: static XPoint seg5_3122[] = {
6655:
                              \{2, -5\}, \{8, -5\},\
6656: };
6657: static XPoint seg6 3122[] = {
6658:
                              \{-7, -5\}, \{-4, -3\},
6659: };
6660: static XPoint seg7 3122[] = {
6661:
                              \{-2, -5\}, \{-4, -4\},\
6662: };
6663: static XPoint seg8 3122[] = {
6664:
                              \{4, -5\}, \{6, -4\},\
6665: };
6666: static XPoint seg9 3122[] = {
6667:
                              \{7, -5\}, \{6, -4\},\
6668: };
6669: static XPoint *char3122[] = {
6670:
                               seg0_3122, seg1_3122, seg2_3122, seg3_3122, seg4_3122, seg5_3122,
6671:
                               seg6_3122, seg7_3122, seg8_3122, seg9_3122,
6672:
                              NULL,
6673: };
6674: static int char_p3122[] = {
6675:
                              XtNumber(seg0_3122),XtNumber(seg1_3122),XtNumber(seg2_3122),
6676:
                              XtNumber(seg3 3122), XtNumber(seg4 3122), XtNumber(seg5 3122),
6677:
                              XtNumber(seg6_3122),XtNumber(seg7_3122),XtNumber(seg8_3122),
6678:
                              XtNumber(seq9 3122),
6679: };
6680: static XPoint seg0 3123[] = {
6681:
                               \{-8, -5\}, \{-4, 9\},
6682: };
6683: static XPoint seg1_3123[] = {
6684:
                              \{-7, -5\}, \{-4, 6\},
6685: };
6686: static XPoint seg2_3123[] = {
6687:
                              \{-6, -5\}, \{-3, 6\},\
6688: };
6689: static XPoint seg3 3123[] = {
6690:
                              \{0, -5\}, \{-3, 6\}, \{-4, 9\}, \{-4, 9\}, \{-3, 6\}, \{-4, 9\}, \{-3, 6\}, \{-4, 9\}, \{-3, 6\}, \{-4, 9\}, \{-3, 6\}, \{-4, 9\}, \{-3, 6\}, \{-4, 9\}, \{-3, 6\}, \{-3, 6\}, \{-4, 9\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{-3, 6\}, \{
6691: };
6692: static XPoint seg4 3123[] = {
6693:
                              \{0, -5\}, \{4, 9\},\
6694: };
6695: static XPoint seg5 3123[] = {
6696:
                              \{1, -5\}, \{4, 6\},\
6697: };
6698: static XPoint seg6 3123[] = {
6699:
                              \{0, -5\}, \{2, -5\}, \{5, 6\},\
6700: };
6701: static XPoint seg7 3123[] = {
6702:
                               \{8, -4\}, \{5, 6\}, \{4, 9\},\
6703: };
```

```
6704: static XPoint seg8 3123[] = {
                             \{-11, -5\}, \{-3, -5\},
6705:
6706: };
6707: static XPoint seg9 3123[] = {
6708:
                            \{5, -5\}, \{11, -5\},\
6709: };
6710: static XPoint seg10 3123[] = {
6711:
                            \{-10, -5\}, \{-7, -4\},\
6712: };
6713: static XPoint seg11 3123[] = {
6714:
                            \{-4, -5\}, \{-6, -4\},\
6715: };
6716: static XPoint seg12 3123[] = {
6717:
                            \{6, -5\}, \{8, -4\},\
6718: };
6719: static XPoint seg13 3123[] = {
6720:
                            \{10, -5\}, \{8, -4\},\
6721: };
6722: static XPoint *char3123[] = {
6723:
                             seg0_3123, seg1_3123, seg2_3123, seg3_3123, seg4_3123,
6724:
                             seg5_3123, seg6_3123, seg7_3123, seg8_3123, seg9_3123,
6725:
                             seg10 3123, seg11 3123, seg12 3123, seg13 3123,
6726:
                            NULL,
6727: };
6728: static int char p3123[] = {
6729:
                            XtNumber(seg0_3123),XtNumber(seg1_3123),XtNumber(seg2_3123),
6730:
                            XtNumber(seg3 3123), XtNumber(seg4 3123), XtNumber(seg5 3123),
6731:
                            XtNumber(seg6 3123), XtNumber(seg7 3123), XtNumber(seg8 3123),
6732:
                            XtNumber(seg9 3123), XtNumber(seg10 3123), XtNumber(seg11 3123),
6733:
                            XtNumber(seg12 3123), XtNumber(seg13 3123),
6734: };
6735: static XPoint seg0_3124[] = {
6736:
                            \{-6, -5\}, \{4, 9\},\
6737: };
6738: static XPoint seg1_3124[] = {
6739:
                            \{-5, -5\}, \{5, 9\},\
6740: };
6741: static XPoint seg2 3124[] = {
6742:
                            \{-4, -5\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6, 9\}, \{6
6743: };
6744: static XPoint seg3 3124[] = {
6745:
                            \{5, -4\}, \{-5, 8\},\
6746: };
6747: static XPoint seg4 3124[] = {
6748:
                            \{-8, -5\}, \{-1, -5\},\
6749: };
6750: static XPoint seg5_3124[] = {
6751:
                            \{2, -5\}, \{8, -5\},\
6752: };
6753: static XPoint seg6 3124[] = {
6754:
                             \{-8,9\},\{-2,9\},
6755: };
```

```
Listing C.4 Continued
```

```
6756: static XPoint seg7 3124[] = {
6757:
                            \{1,9\},\{8,9\},\
6758: };
6759: static XPoint seg8_3124[] = {
6760:
                            \{-7, -5\}, \{-5, -4\},
6761: };
6762: static XPoint seg9 3124[] = {
6763:
                            \{-2, -5\}, \{-4, -4\},
6764: };
6765: static XPoint seg10 3124[] = {
6766:
                            \{3, -5\}, \{5, -4\},\
6767: };
6768: static XPoint seg11 3124[] = {
6769:
                            \{7, -5\}, \{5, -4\},\
6770: };
6771: static XPoint seg12 3124[] = {
6772:
                             \{-5,8\},\{-7,9\},
6773: };
6774: static XPoint seg13_3124[] = {
6775:
                            \{-5,8\},\{-3,9\},
6776: };
6777: static XPoint seg14_3124[] = {
6778:
                            \{4,8\},\{2,9\},\
6779: };
6780: static XPoint seg15 3124[] = {
6781:
                            \{5,8\},\{7,9\},\
6782: };
6783: static XPoint *char3124[] = {
6784:
                            seg0 3124, seg1 3124, seg2 3124, seg3 3124, seg4 3124,
6785:
                             seg5 3124, seg6 3124, seg7 3124, seg8 3124, seg9 3124,
6786:
                             seg10 3124, seg11 3124, seg12 3124, seg13 3124, seg14 3124,
6787:
                             seg15 3124,
6788:
                            NULL,
6789: };
6790: static int char p3124[] = {
6791:
                            XtNumber(seg0_3124),XtNumber(seg1_3124),XtNumber(seg2_3124),
6792:
                             XtNumber(seg3_3124), XtNumber(seg4_3124), XtNumber(seg5_3124),
6793:
                            XtNumber(seg6 3124), XtNumber(seg7 3124), XtNumber(seg8 3124),
6794:
                             XtNumber(seg9 3124), XtNumber(seg10 3124), XtNumber(seg11 3124),
6795:
                            XtNumber(seg12 3124), XtNumber(seg13 3124), XtNumber(seg14 3124),
6796:
                            XtNumber(seg15 3124),
6797: };
6798: static XPoint seg0 3125[] = {
6799:
                            \{-6, -5\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0
6800: };
6801: static XPoint seg1 3125[] = {
6802:
                             \{-5, -5\}, \{0, 7\},\
6803: };
6804: static XPoint seg2_3125[] = {
6805:
                             \{-4, -5\}, \{1, 7\},
6806: };
```

```
6807: static XPoint seg3 3125[] = {
                             \{6, -4\}, \{1, 7\}, \{-2, 13\}, \{-4, 15\}, \{-6, 16\}, \{-8, 16\}, \{-9, 15\},
6808:
6809:
                             \{-9,13\},\{-7,13\},\{-7,15\},\{-8,15\},\{-8,14\},
6810: };
6811: static XPoint seg4_3125[] = {
6812:
                            \{-8, -5\}, \{-1, -5\},
6813: };
6814: static XPoint seg5 3125[] = {
6815:
                             \{2, -5\}, \{8, -5\}, \{2, -5\}, \{2, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{3, -5\}, \{
6816: };
6817: static XPoint seg6 3125[] = {
6818:
                             \{-7, -5\}, \{-4, -3\},
6819: };
6820: static XPoint seg7 3125[] = {
6821:
                             \{-2, -5\}, \{-4, -4\},
6822: };
6823: static XPoint seg8 3125[] = {
6824:
                             \{4, -5\}, \{6, -4\},\
6825: };
6826: static XPoint seg9 3125[] = {
6827:
                             \{7, -5\}, \{6, -4\},\
6828: };
6829: static XPoint *char3125[] = {
6830:
                             seg0_3125, seg1_3125, seg2_3125, seg3_3125, seg4_3125, seg5_3125,
6831:
                             seg6 3125, seg7 3125, seg8 3125, seg9 3125,
6832:
                             NULL,
6833: };
6834: static int char p3125[] = {
6835:
                             XtNumber(seg0 3125), XtNumber(seg1 3125), XtNumber(seg2 3125),
                             XtNumber(seg3 3125), XtNumber(seg4 3125), XtNumber(seg5 3125),
6836:
6837:
                             XtNumber(seg6 3125), XtNumber(seg7 3125), XtNumber(seg8 3125),
6838:
                             XtNumber(seg9_3125),
6839: };
6840: static XPoint seg0 3126[] = {
6841:
                             \{4, -5\}, \{-6, 9\},\
6842: };
6843: static XPoint seg1 3126[] = {
6844:
                             \{5, -5\}, \{-5, 9\},\
6845: };
6846: static XPoint seg2 3126[] = {
6847:
                             \{6, -5\}, \{-4, 9\},\
6848: };
6849: static XPoint seg3 3126[] = {
6850:
                             \{6, -5\}, \{-6, -5\}, \{-6, -1\}, 
6851: };
6852: static XPoint seg4 3126[] = {
6853:
                             \{-6,9\},\{6,9\},\{6,5\},
6854: };
6855: static XPoint seg5_3126[] = {
6856:
                             \{-5, -5\}, \{-6, -1\},
6857: };
```

```
Listing C.4 Continued
```

```
6858: static XPoint seg6 3126[] = {
6859:
           \{-4, -5\}, \{-6, -2\},\
6860: };
6861: static XPoint seg7 3126[] = {
6862:
           \{-3, -5\}, \{-6, -3\}, 
6863: };
6864: static XPoint seg8 3126[] = {
6865:
           \{-1, -5\}, \{-6, -4\},\
6866: };
6867: static XPoint seg9 3126[] = {
6868:
           \{1,9\},\{6,8\},\
6869: };
6870: static XPoint seg10 3126[] = {
           \{3,9\},\{6,7\},
6871:
6872: };
6873: static XPoint seg11 3126[] = {
6874:
           \{4,9\},\{6,6\},\
6875: };
6876: static XPoint seg12 3126[] = {
6877:
           \{5,9\},\{6,5\},\
6878: };
6879: static XPoint *char3126[] = {
           seg0_3126, seg1_3126, seg2_3126, seg3_3126, seg4_3126,
6880:
6881:
           seg5_3126, seg6_3126, seg7_3126, seg8_3126, seg9_3126,
6882:
           seg10 3126, seg11 3126, seg12 3126,
6883:
           NULL,
6884: };
6885: static int char p3126[] = {
6886:
           XtNumber(seg0_3126),XtNumber(seg1_3126),XtNumber(seg2_3126),
           XtNumber(seg3_3126),XtNumber(seg4_3126),XtNumber(seg5_3126),
6887:
6888:
           XtNumber(seg6 3126),XtNumber(seg7 3126),XtNumber(seg8 3126),
6889:
           XtNumber(seg9 3126), XtNumber(seg10 3126), XtNumber(seg11 3126),
6890:
           XtNumber(seg12 3126),
6891: };
6892: static XPoint seg0 3151[] = {
6893:
           \{5, -5\}, \{3, 2\}, \{3, 6\}, \{4, 8\}, \{5, 9\}, \{7, 9\},
6894:
           \{9,7\},\{10,5\},\
6895: };
6896: static XPoint seg1_3151[] = {
6897:
           \{6, -5\}, \{4, 2\}, \{4, 8\}, 
6898: };
6899: static XPoint seg2 3151[] = {
6900:
           \{5, -5\}, \{7, -5\}, \{5, 2\}, \{4, 6\},\
6901: };
6902: static XPoint seg3 3151[] = {
6903:
           \{3,2\},\{3,-1\},\{2,-4\},\{0,-5\},\{-2,-5\},\{-5,-4\},\{-7,-1\},\{-8,2\},
6904:
           \{-8,4\},\{-7,7\},\{-6,8\},\{-4,9\},\{-2,9\},\{0,8\},\{1,7\},\{2,5\},\{3,2\},
6905: };
6906: static XPoint seg4 3151[] = {
6907:
           \{-4, -4\}, \{-6, -1\}, \{-7, 2\}, \{-7, 5\}, \{-6, 7\},
6908: };
```

```
6909: static XPoint seg5 3151[] = {
            \{-2, -5\}, \{-4, -3\}, \{-5, -1\}, \{-6, 2\}, \{-6, 5\}, \{-5, 8\}, \{-4, 9\},
6910:
6911: };
6912: static XPoint *char3151[] = {
6913:
            seg0_3151, seg1_3151, seg2_3151, seg3_3151, seg4_3151, seg5_3151,
6914:
           NULL,
6915: };
6916: static int char_p3151[] = {
6917:
            XtNumber(seg0 3151), XtNumber(seg1 3151), XtNumber(seg2 3151),
6918:
           XtNumber(seg3 3151), XtNumber(seg4 3151), XtNumber(seg5 3151),
6919: };
6920: static XPoint seg0 3152[] = {
6921:
           \{-2, -12\}, \{-4, -5\}, \{-5, 1\}, \{-5, 5\}, \{-4, 7\}, \{-3, 8\}, \{-1, 9\},
6922:
            \{1,9\},\{4,8\},\{6,5\},\{7,2\},\{7,0\},\{6,-3\},\{5,-4\},\{3,-5\},
6923:
            \{1, -5\}, \{-1, -4\}, \{-2, -3\}, \{-3, -1\}, \{-4, 2\},
6924: };
6925: static XPoint seg1 3152[] = {
6926:
            \{-1, -12\}, \{-3, -5\}, \{-4, -1\}, \{-4, 5\}, \{-3, 8\},
6927: };
6928: static XPoint seg2 3152[] = {
6929:
           \{4,7\},\{5,5\},\{6,2\},\{6,-1\},\{5,-3\},
6930: };
6931: static XPoint seg3 3152[] = {
6932:
           \{-5, -12\}, \{0, -12\}, \{-2, -5\}, \{-4, 2\}, 
6933: };
6934: static XPoint seg4 3152[] = {
6935:
           \{1,9\},\{3,7\},\{4,5\},\{5,2\},\{5,-1\},\{4,-4\},\{3,-5\},
6936: };
6937: static XPoint seg5 3152[] = {
6938:
            \{-4, -12\}, \{-1, -11\},
6939: };
6940: static XPoint seg6 3152[] = {
6941:
           \{-3, -12\}, \{-2, -10\},
6942: };
6943: static XPoint *char3152[] = {
            seg0 3152, seg1 3152, seg2 3152, seg3 3152, seg4 3152, seg5 3152,
6944:
6945:
           seg6 3152,
6946:
           NULL,
6947: };
6948: static int char p3152[] = {
           XtNumber(seg0 3152), XtNumber(seg1 3152), XtNumber(seg2 3152),
6949:
6950:
           XtNumber(seg3 3152), XtNumber(seg4 3152), XtNumber(seg5 3152),
6951:
           XtNumber(seg6 3152),
6952: };
6953: static XPoint seg0 3153[] = {
6954:
            \{5, -1\}, \{5, -2\}, \{4, -2\}, \{4, 0\}, \{6, 0\}, \{6, -2\},
6955:
            \{5, -4\}, \{3, -5\}, \{0, -5\}, \{-3, -4\}, \{-5, -1\}, \{-6, 2\}, \{-6, 4\}, \{-5, 7\},
6956:
            \{-4,8\},\{-2,9\},\{0,9\},\{3,8\},\{5,5\},
6957: };
6958: static XPoint seg1 3153[] = {
6959:
            \{-3, -3\}, \{-4, -1\}, \{-5, 2\}, \{-5, 5\}, \{-4, 7\},
6960: };
```

```
C
```

```
Listing C.4 Continued
```

```
6961: static XPoint seg2_3153[] = {
6962:
                                           \{0, -5\}, \{-2, -3\}, \{-3, -1\}, \{-4, 2\}, \{-4, 5\}, \{-3, 8\}, \{-2, 9\}, \{-2, 9\}, \{-3, 8\}, \{-2, 9\}, \{-3, 8\}, \{-2, 9\}, \{-3, 8\}, \{-3, 8\}, \{-2, 9\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\},
6963: };
6964: static XPoint *char3153[] = {
6965:
                                           seg0 3153, seg1 3153, seg2 3153,
6966:
                                          NULL,
6967: };
6968: static int char p3153[] = {
                                          XtNumber(seg0 3153), XtNumber(seg1 3153), XtNumber(seg2 3153),
6969:
6970: };
6971: static XPoint seg0 3154[] = {
6972:
                                           \{7, -12\},\
6973:
                                           \{4, -1\}, \{3, 3\}, \{3, 6\}, \{4, 8\}, \{5, 9\}, \{7, 9\}, \{9, 7\}, \{10, 5\},
6974: };
6975: static XPoint seg1 3154[] = {
6976:
                                          \{8, -12\}, \{5, -1\}, \{4, 3\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{
6977: };
6978: static XPoint seg2 3154[] = {
6979:
                                           \{4, -12\}, \{9, -12\}, \{5, 2\}, \{4, 6\},
6980: };
6981: static XPoint seg3 3154[] = {
6982:
                                           \{3,2\},\{3,-1\},\{2,-4\},\{0,-5\},\{-2,-5\},\{-5,-4\},\{-7,-1\},\{-8,2\},
6983:
                                           \{-8,4\},\{-7,7\},\{-6,8\},\{-4,9\},\{-2,9\},\{0,8\},\{1,7\},\{2,5\},\{3,2\},
6984: };
6985: static XPoint seg4 3154[] = {
6986:
                                           \{-5, -3\}, \{-6, -1\}, \{-7, 2\}, \{-7, 5\}, \{-6, 7\},
6987: };
6988: static XPoint seg5 3154[] = {
                                          \{-2, -5\}, \{-4, -3\}, \{-5, -1\}, \{-6, 2\}, \{-6, 5\}, \{-5, 8\}, \{-4, 9\},
6989:
6990: };
6991: static XPoint seg6 3154[] = {
6992:
                                          \{5, -12\}, \{8, -11\},\
6993: };
6994: static XPoint seg7 3154[] = {
6995:
                                           \{6, -12\}, \{7, -10\},\
6996: };
6997: static XPoint *char3154[] = {
6998:
                                           seg0_3154, seg1_3154, seg2_3154, seg3_3154, seg4_3154, seg5_3154,
6999:
                                           seg6 3154, seg7 3154,
7000:
                                          NULL,
7001: };
7002: static int char p3154[] = {
7003:
                                          XtNumber(seg0_3154), XtNumber(seg1_3154), XtNumber(seg2_3154),
7004:
                                          XtNumber(seq3 3154), XtNumber(seq4 3154), XtNumber(seq5 3154),
7005:
                                          XtNumber(seg6 3154), XtNumber(seg7 3154),
7006: };
7007: static XPoint seg0 3155[] = {
7008:
                                           \{-5,4\},\{-1,3\},\{2,2\},\{5,0\},\{6,-2\},\{5,-4\},
7009:
                                           \{3, -5\}, \{0, -5\}, \{-3, -4\}, \{-5, -1\}, \{-6, 2\}, \{-6, 4\}, \{-5, 7\}, \{-4, 8\},
7010:
                                           \{-2,9\},\{0,9\},\{3,8\},\{5,6\},
7011: };
```

```
7012: static XPoint seg1 3155[] = {
7013:
                                                                     \{-3, -3\}, \{-4, -1\}, \{-5, 2\}, \{-5, 5\}, \{-4, 7\},
7014: };
7015: static XPoint seg2 3155[] = {
7016:
                                                                    \{0, -5\}, \{-2, -3\}, \{-3, -1\}, \{-4, 2\}, \{-4, 5\}, \{-3, 8\}, \{-2, 9\},
7017: };
7018: static XPoint *char3155[] = {
7019:
                                                                    seg0_3155, seg1_3155, seg2_3155,
7020:
                                                                    NULL,
7021: };
7022: static int char p3155[] = {
7023:
                                                                    XtNumber(seg0 3155), XtNumber(seg1 3155), XtNumber(seg2 3155),
7024: };
7025: static XPoint seg0 3156[] = {
7026:
                                                                     \{8, -10\}, \{8, -11\}, \{7, -11\}, \{7, -9\}, \{9, -9\}, \{9, -11\}, \{8, -12\},
7027:
                                                                     \{6, -12\}, \{4, -11\}, \{2, -9\}, \{1, -7\}, \{0, -4\}, \{-1, 0\}, \{-3, 9\}, \{-4, 12\}, \{-1, 0\}, \{-3, 9\}, \{-4, 12\}, \{-1, 0\}, \{-3, 9\}, \{-4, 12\}, \{-1, 0\}, \{-3, 9\}, \{-4, 12\}, \{-1, 0\}, \{-3, 9\}, \{-4, 12\}, \{-3, 9\}, \{-4, 12\}, \{-3, 9\}, \{-4, 12\}, \{-3, 9\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, \{-4, 12\}, 
7028:
                                                                    {-5,14}, {-7,16},
7029: };
7030: static XPoint seg1 3156[] = {
7031:
                                                                    \{2, -8\}, \{1, -5\}, \{0, 0\}, \{-2, 9\}, \{-3, 12\}, 
7032: };
7033: static XPoint seg2 3156[] = {
7034:
                                                                     \{6, -12\}, \{4, -10\}, \{3, -8\}, \{2, -5\}, \{1, 0\}, \{-1, 8\}, \{-2, 11\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\},
7035:
                                                                     \{-5, 15\}, \{-7, 16\}, \{-9, 16\}, \{-10, 15\}, \{-10, 13\}, \{-8, 13\}, \{-8, 15\},
7036:
                                                                     \{-9, 15\}, \{-9, 14\},
7037: };
7038: static XPoint seq3 3156[] = {
7039:
                                                                    \{-4, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, 
7040: };
7041: static XPoint *char3156[] = {
7042:
                                                                     seg0 3156, seg1 3156, seg2 3156, seg3 3156,
7043:
                                                                    NULL,
7044: };
7045: static int char p3156[] = {
7046:
                                                                    XtNumber(seg0_3156),XtNumber(seg1_3156),XtNumber(seg2_3156),
7047:
                                                                    XtNumber(seq3 3156),
7048: };
7049: static XPoint seg0 3157[] = {
7050:
                                                                     \{6, -5\}, \{2, 9\}, \{1, 12\}, \{-1, 15\}, \{-3, 16\}, 
7051: };
7052: static XPoint seg1_3157[] = {
7053:
                                                                    \{7, -5\}, \{3, 9\}, \{1, 13\}, \{7, -5\}, \{3, 9\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1, 13\}, \{1,
7054: };
7055: static XPoint seg2 3157[] = {
7056:
                                                                     \{6, -5\}, \{8, -5\}, \{4, 9\}, \{2, 13\}, \{0, 15\}, \{-3, 16\}, \{-6, 16\}, \{-8, 15\},
7057:
                                                                     \{-9,14\},\{-9,12\},\{-7,12\},\{-7,14\},\{-8,14\},\{-8,13\},
7058: };
7059: static XPoint seg3 3157[] = {
7060:
                                                                     \{4,2\},\{4,-1\},\{3,-4\},\{1,-5\},\{-1,-5\},\{-4,-4\},\{-6,-1\},\{-7,2\},
7061:
                                                                     \{-7,4\},\{-6,7\},\{-5,8\},\{-3,9\},\{-1,9\},\{1,8\},\{2,7\},\{3,5\},\{4,2\},
7062: };
7063: static XPoint seq4 3157[] = {
7064:
                                                                     \{-4, -3\}, \{-5, -1\}, \{-6, 2\}, \{-6, 5\}, \{-5, 7\},
7065: };
```
```
7066: static XPoint seg5_3157[] = {
                           \{-1, -5\}, \{-3, -3\}, \{-4, -1\}, \{-5, 2\}, \{-5, 5\}, \{-4, 8\}, \{-3, 9\},
7067:
7068: };
7069: static XPoint *char3157[] = {
7070:
                           seg0 3157, seg1 3157, seg2 3157, seg3 3157, seg4 3157, seg5 3157,
7071:
                           NULL,
7072: };
7073: static int char p3157[] = {
7074:
                           XtNumber(seg0 3157), XtNumber(seg1 3157), XtNumber(seg2 3157),
7075:
                           XtNumber(seg3_3157),XtNumber(seg4_3157),XtNumber(seg5_3157),
7076: };
7077: static XPoint seg0 3158[] = {
                           {-3,-12},
7078:
7079:
                           \{-9,9\},\{-7,9\},
7080: };
7081: static XPoint seg1_3158[] = {
7082:
                           \{-2, -12\}, \{-8, 9\},
7083: };
7084: static XPoint seg2_3158[] = {
                           \{-6, -12\}, \{-1, -12\}, \{-7, 9\},
7085:
7086: };
7087: static XPoint seg3 3158[] = {
7088:
                           \{-5,2\},\{-3,-2\},\{-1,-4\},\{1,-5\},\{3,-5\},\{5,-4\},\{6,-2\},\{6,1\},
7089:
                           {4,6},
7090: };
7091: static XPoint seg4 3158[] = {
7092:
                           \{5, -4\}, \{5, 0\}, \{4, 4\}, \{4, 8\},\
7093: };
7094: static XPoint seg5 3158[] = {
7095:
                           \{5, -2\}, \{3, 3\}, \{3, 6\}, \{4, 8\}, \{5, 9\}, \{7, 9\}, \{9, 7\}, \{10, 5\},
7096: };
7097: static XPoint seg6_3158[] = {
7098:
                           \{-5, -12\}, \{-2, -11\}, 
7099: };
7100: static XPoint seg7 3158[] = {
7101:
                           \{-4, -12\}, \{-3, -10\},\
7102: };
7103: static XPoint *char3158[] = {
7104:
                           seg0 3158, seg1 3158, seg2 3158, seg3 3158, seg4 3158, seg5 3158,
7105:
                           seg6 3158, seg7 3158,
7106:
                           NULL,
7107: };
7108: static int char_p3158[] = {
7109:
                           XtNumber(seq0 3158), XtNumber(seq1 3158), XtNumber(seq2 3158),
7110:
                           XtNumber(seg3 3158), XtNumber(seg4 3158), XtNumber(seg5 3158),
7111:
                           XtNumber(seg6 3158), XtNumber(seg7 3158),
7112: };
7113: static XPoint seg0 3159[] = {
7114:
                           \{1, -12\}, \{1, -10\}, \{3, -10\}, \{3, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, 
7115: };
```

```
7116: static XPoint seg1 3159[] = {
7117:
                                                       \{2, -12\}, \{2, -10\}, 
7118: };
7119: static XPoint seg2 3159[] = {
7120:
                                                      \{1, -11\}, \{3, -11\}, 
7121: };
7122: static XPoint seg3 3159[] = {
7123:
                                                      \{-6, -1\}, \{-5, -3\}, \{-3, -5\}, \{-1, -5\}, \{0, -4\}, \{1, -2\}, \{1, 1\}, \{-1, 6\}, \{-6, -1\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\}, \{-1, -2\},
7124: };
7125: static XPoint seg4 3159[] = {
7126:
                                                      \{0, -4\}, \{0, 0\}, \{-1, 4\}, \{-1, 8\},\
7127: };
7128: static XPoint seg5 3159[] = {
7129:
                                                      \{0, -2\}, \{-2, 3\}, \{-2, 6\}, \{-1, 8\}, \{0, 9\}, \{2, 9\}, \{4, 7\}, \{5, 5\}, \}
7130: };
7131: static XPoint *char3159[] = {
7132:
                                                       seg0_3159, seg1_3159, seg2_3159, seg3_3159, seg4_3159, seg5_3159,
7133:
                                                      NULL,
7134: };
7135: static int char p3159[] = {
7136:
                                                      XtNumber(seg0 3159), XtNumber(seg1 3159), XtNumber(seg2 3159),
7137:
                                                      XtNumber(seg3 3159), XtNumber(seg4 3159), XtNumber(seg5 3159),
7138: };
7139: static XPoint seg0 3160[] = {
7140:
                                                       \{3, -12\}, \{3, -10\}, \{5, -10\}, \{5, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, 
7141: };
7142: static XPoint seq1 3160[] = {
7143:
                                                      \{4, -12\}, \{4, -10\}, 
7144: };
7145: static XPoint seg2 3160[] = {
7146:
                                                    \{3, -11\}, \{5, -11\}, 
7147: };
7148: static XPoint seg3 3160[] = {
7149:
                                                       \{-5, -1\}, \{-4, -3\}, \{-2, -5\}, \{0, -5\}, \{1, -4\}, \{2, -2\}, \{2, 1\}, \{0, 8\}, 
7150:
                                                       \{-1,11\},\{-2,13\},\{-4,15\},\{-6,16\},\{-8,16\},\{-9,15\},\{-9,13\},
7151:
                                                       \{-7, 13\}, \{-7, 15\}, \{-8, 15\}, \{-8, 14\},
7152: };
7153: static XPoint seg4 3160[] = {
7154:
                                                      \{1, -4\}, \{1, 1\}, \{-1, 8\}, \{-2, 11\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}, \{-3, 13\}
7155: };
7156: static XPoint seg5 3160[] = {
7157:
                                                      \{1, -2\}, \{0, 2\}, \{-2, 9\}, \{-3, 12\}, \{-4, 14\}, \{-6, 16\},\
7158: };
7159: static XPoint *char3160[] = {
                                                       seg0 3160, seg1 3160, seg2 3160, seg3 3160, seg4 3160, seg5 3160,
7160:
7161:
                                                      NULL,
7162: };
7163: static int char p3160[] = {
7164:
                                                      XtNumber(seg0_3160),XtNumber(seg1_3160),XtNumber(seg2_3160),
7165:
                                                      XtNumber(seg3 3160), XtNumber(seg4 3160), XtNumber(seg5 3160),
7166: };
7167: static XPoint seq0 3161[] = {
7168:
                                                      \{-3, -12\}, \{-9, 9\},
```

```
7169:
                                             \{-7,9\},\
7170: };
7171: static XPoint seg1 3161[] = {
7172:
                                             \{-2, -12\}, \{-8, 9\},
7173: };
7174: static XPoint seg2 3161[] = {
7175:
                                              \{-6, -12\}, \{-1, -12\}, \{-7, 9\},
7176: };
7177: static XPoint seg3 3161[] = {
7178:
                                              \{7, -3\}, \{7, -4\}, \{6, -4\}, \{6, -2\}, \{8, -2\}, \{8, -4\}, \{7, -5\}, \{5, -5\},
7179:
                                              \{3, -4\}, \{-1, 0\}, \{-3, 1\},\
7180: };
7181: static XPoint seg4 3161[] = {
7182:
                                              \{-5,1\},\{-3,1\},\{-1,2\},\{0,3\},\{2,7\},\{3,8\},\{5,8\},
7183: };
7184: static XPoint seg5 3161[] = {
7185:
                                             \{-1,3\},\{1,7\},\{2,8\},
7186: };
7187: static XPoint seg6 3161[] = {
7188:
                                             \{-3,1\},\{-2,2\},\{0,8\},\{1,9\},\{3,9\},\{5,8\},\{7,5\},
7189: };
7190: static XPoint seg7 3161[] = {
7191:
                                             \{-5, -12\}, \{-2, -11\},
7192: };
7193: static XPoint seg8 3161[] = {
7194:
                                             \{-4, -12\}, \{-3, -10\},
7195: };
7196: static XPoint *char3161[] = {
7197:
                                              seg0 3161, seg1 3161, seg2 3161, seg3 3161, seg4 3161, seg5 3161,
7198:
                                              seg6_3161, seg7_3161, seg8_3161,
7199:
                                             NULL,
7200: };
7201: static int char p3161[] = {
7202:
                                             XtNumber(seg0 3161), XtNumber(seg1 3161), XtNumber(seg2 3161),
7203:
                                             XtNumber(seg3 3161), XtNumber(seg4 3161), XtNumber(seg5 3161),
7204:
                                             XtNumber(seq6 3161), XtNumber(seq7 3161), XtNumber(seq8 3161),
7205: };
7206: static XPoint seg0 3162[] = {
7207:
                                              \{2, -12\}, \{-1, -1\}, \{-2, 3\}, \{-2, 6\}, \{-1, 8\}, \{0, 9\}, \{0, 9\}, \{-1, 8\}, \{0, 9\}, \{-1, 8\}, \{0, 9\}, \{0, 9\}, \{-1, 8\}, \{0, 9\}, \{-1, 8\}, \{0, 9\}, \{-1, 8\}, \{0, 9\}, \{-1, 8\}, \{0, 9\}, \{-1, 8\}, \{0, 9\}, \{-1, 8\}, \{0, 9\}, \{-1, 8\}, \{-1, 8\}, \{0, 9\}, \{-1, 8\}, \{0, 9\}, \{-1, 8\}, \{0, 9\}, \{-1, 8\}, \{0, 9\}, \{-1, 8\}, \{-1, 8\}, \{0, 9\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8
7208:
                                              \{2,9\},\{4,7\},\{5,5\},
7209: };
7210: static XPoint seg1 3162[] = {
7211:
                                             \{3, -12\}, \{0, -1\}, \{-1, 3\}, \{-1, 8\},\
7212: };
7213: static XPoint seg2 3162[] = {
7214:
                                             \{-1, -12\}, \{4, -12\}, \{0, 2\}, \{-1, 6\}, \{0, 2\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, 
7215: };
7216: static XPoint seg3 3162[] = {
7217:
                                              \{0, -12\}, \{3, -11\},\
7218: };
```

```
7219: static XPoint seg4 3162[] = {
7220:
                                                       \{1, -12\}, \{2, -10\},\
7221: };
7222: static XPoint *char3162[] = {
                                                       seg0_3162, seg1_3162, seg2_3162, seg3_3162, seg4_3162,
7223:
7224:
                                                      NULL,
7225: };
7226: static int char_p3162[] = {
7227:
                                                       XtNumber(seg0 3162), XtNumber(seg1 3162), XtNumber(seg2 3162),
7228:
                                                      XtNumber(seg3 3162), XtNumber(seg4 3162),
7229: };
7230: static XPoint seg0 3163[] = {
7231:
                                                      \{-17, -1\}, \{-16, -3\}, \{-14, -5\}, \{-12, -5\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-11, -4\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}, \{-10, -2\}
7232:
                                                       \{-10,1\},\{-12,9\},
7233: };
7234: static XPoint seg1_3163[] = {
7235:
                                                     \{-11, -4\}, \{-11, 1\}, \{-13, 9\},
7236: };
7237: static XPoint seg2 3163[] = {
7238:
                                                      \{-11, -2\}, \{-12, 2\}, \{-14, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-12, 9\}, \{-1
7239: };
7240: static XPoint seg3 3163[] = {
7241:
                                                       \{-10,1\},\{-8,-2\},\{-6,-4\},\{-4,-5\},\{-2,-5\},\{0,-4\},\{1,-2\},\{1,1\},
7242:
                                                       \{-1,9\},\
7243: };
7244: static XPoint seg4 3163[] = {
7245:
                                                      \{0, -4\}, \{0, 1\}, \{-2, 9\},\
7246: };
7247: static XPoint seg5 3163[] = {
7248:
                                                       \{0, -2\}, \{-1, 2\}, \{-3, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{
7249: };
7250: static XPoint seg6 3163[] = {
7251:
                                                       \{1,1\},\{3,-2\},\{5,-4\},\{7,-5\},\{9,-5\},\{11,-4\},\{12,-2\},\{12,1\},
7252:
                                                       \{10, 6\},\
7253: };
7254: static XPoint seg7 3163[] = {
7255:
                                                       \{11, -4\}, \{11, 0\}, \{10, 4\}, \{10, 8\},\
7256: };
7257: static XPoint seg8 3163[] = {
7258:
                                                       \{11, -2\}, \{9, 3\}, \{9, 6\}, \{10, 8\}, \{11, 9\}, \{13, 9\}, \{15, 7\}, \{16, 5\},
7259: };
7260: static XPoint *char3163[] = {
7261:
                                                       seg0 3163, seg1 3163, seg2 3163, seg3 3163, seg4 3163, seg5 3163,
7262:
                                                       seg6 3163, seg7 3163, seg8 3163,
7263:
                                                      NULL,
7264: };
7265: static int char_p3163[] = {
7266:
                                                       XtNumber(seg0 3163),XtNumber(seg1 3163),XtNumber(seg2 3163),
7267:
                                                       XtNumber(seg3_3163),XtNumber(seg4_3163),XtNumber(seg5_3163),
7268:
                                                      XtNumber(seg6 3163), XtNumber(seg7 3163), XtNumber(seg8 3163),
7269: };
7270: static XPoint seg0 3164[] = {
7271:
                                                       \{-11, -1\}, \{-10, -3\}, \{-8, -5\}, \{-6, -5\}, \{-5, -4\}, \{-4, -2\}, \{-4, 1\},
```

```
7272:
                                         \{-6,9\},\
7273: };
7274: static XPoint seg1 3164[] = {
7275:
                                         \{-5, -4\}, \{-5, 1\}, \{-7, 9\},
7276: };
7277: static XPoint seg2 3164[] = {
7278:
                                         \{-5, -2\}, \{-6, 2\}, \{-8, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, \{-6, 9\}, 
7279: };
7280: static XPoint seg3 3164[] = {
7281:
                                          \{-4,1\},\{-2,-2\},\{0,-4\},\{2,-5\},\{4,-5\},\{6,-4\},\{7,-2\},\{7,1\},
7282:
                                          {5,6},
7283: };
7284: static XPoint seg4 3164[] = {
7285:
                                          \{6, -4\}, \{6, 0\}, \{5, 4\}, \{5, 8\},\
7286: };
7287: static XPoint seg5 3164[] = {
7288:
                                          \{6, -2\}, \{4, 3\}, \{4, 6\}, \{5, 8\}, \{6, 9\}, \{8, 9\}, \{10, 7\}, \{11, 5\},
7289: };
7290: static XPoint *char3164[] = {
                                          seg0 3164, seg1 3164, seg2 3164, seg3 3164, seg4 3164, seg5 3164,
7291:
7292:
                                         NULL,
7293: };
7294: static int char p3164[] = {
7295:
                                         XtNumber(seg0 3164), XtNumber(seg1 3164), XtNumber(seg2 3164),
                                         XtNumber(seg3_3164),XtNumber(seg4_3164),XtNumber(seg5_3164),
7296:
7297: };
7298: static XPoint seg0 3165[] = {
7299:
                                          \{-1, -5\}, \{-4, -4\}, \{-6, -1\}, \{-7, 2\}, \{-7, 4\}, \{-6, 7\}, \{-5, 8\}, \{-2, 9\}, \{-1, -5\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 9\}, \{-2, -2, 
7300:
                                          \{1,9\},\{4,8\},\{6,5\},\{7,2\},\{7,0\},\{6,-3\},\{5,-4\},\{2,-5\},\{-1,-5\},
7301: };
7302: static XPoint seg1 3165[] = {
7303:
                                         \{-4, -3\}, \{-5, -1\}, \{-6, 2\}, \{-6, 5\}, \{-5, 7\},
7304: };
7305: static XPoint seg2 3165[] = {
7306:
                                         \{4,7\},\{5,5\},\{6,2\},\{6,-1\},\{5,-3\},
7307: };
7308: static XPoint seg3 3165[] = {
7309:
                                          \{-1, -5\}, \{-3, -3\}, \{-4, -1\}, \{-5, 2\}, \{-5, 5\}, \{-4, 8\}, \{-2, 9\},
7310: };
7311: static XPoint seg4 3165[] = {
7312:
                                          \{1,9\},\{3,7\},\{4,5\},\{5,2\},\{5,-1\},\{4,-4\},\{2,-5\},
7313: };
7314: static XPoint *char3165[] = {
7315:
                                          seg0 3165, seg1 3165, seg2 3165, seg3 3165, seg4 3165,
7316:
                                         NULL,
7317: };
7318: static int char p3165[] = {
7319:
                                         XtNumber(seg0 3165), XtNumber(seg1 3165), XtNumber(seg2 3165),
7320:
                                          XtNumber(seg3_3165), XtNumber(seg4_3165),
7321: };
```

```
7322: static XPoint seg0 3166[] = {
7323:
                           \{-10, -1\}, \{-9, -3\}, \{-7, -5\}, \{-5, -5\}, \{-4, -4\}, \{-3, -2\}, \{-3, 1\},
7324:
                           \{-4,5\},\{-7,16\},
7325: };
7326: static XPoint seg1 3166[] = {
                           \{-4, -4\}, \{-4, 1\}, \{-5, 5\}, \{-8, 16\},
7327:
7328: };
7329: static XPoint seg2 3166[] = {
7330:
                           \{-4, -2\}, \{-5, 2\}, \{-9, 16\},
7331: };
7332: static XPoint seg3 3166[] = {
7333:
                           \{-3,2\},\{-2,-1\},\{-1,-3\},\{0,-4\},\{2,-5\},\{4,-5\},\{6,-4\},\{7,-3\},
7334:
                           \{8,0\},\{8,2\},\{7,5\},\{5,8\},\{2,9\},\{0,9\},\{-2,8\},\{-3,5\},\{-3,2\},
7335: };
7336: static XPoint seq4 3166[] = {
7337:
                           \{6, -3\}, \{7, -1\}, \{7, 2\}, \{6, 5\}, \{5, 7\},
7338: };
7339: static XPoint seg5 3166[] = {
7340:
                           \{4, -5\}, \{5, -4\}, \{6, -1\}, \{6, 2\}, \{5, 5\}, \{4, 7\}, \{2, 9\},
7341: };
7342: static XPoint seg6 3166[] = {
7343:
                           \{-12, 16\}, \{-4, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\},
7344: };
7345: static XPoint seg7 3166[] = {
7346:
                           \{-8, 15\}, \{-11, 16\},
7347: };
7348: static XPoint seg8 3166[] = {
7349:
                           \{-8, 14\}, \{-10, 16\},
7350: };
7351: static XPoint seg9_3166[] = {
7352:
                           \{-7, 14\}, \{-6, 16\},
7353: };
7354: static XPoint seg10 3166[] = {
7355:
                           \{-8, 15\}, \{-5, 16\},
7356: };
7357: static XPoint *char3166[] = {
7358:
                           seg0 3166, seg1 3166, seg2 3166, seg3 3166, seg4 3166, seg5 3166,
7359:
                           seg6 3166, seg7 3166, seg8 3166, seg9 3166, seg10 3166,
                           NULL,
7360:
7361: };
7362: static int char p3166[] = {
7363:
                           XtNumber(seg0 3166), XtNumber(seg1 3166), XtNumber(seg2 3166),
7364:
                           XtNumber(seg3 3166), XtNumber(seg4 3166), XtNumber(seg5 3166),
                           XtNumber(seg6 3166), XtNumber(seg7 3166), XtNumber(seg8 3166),
7365:
                           XtNumber(seg9 3166), XtNumber(seg10 3166),
7366:
7367: };
7368: static XPoint seg0_3167[] = {
7369:
                           \{5, -5\}, \{-1, 16\},\
7370: };
7371: static XPoint seg1 3167[] = {
7372:
                           \{6, -5\}, \{0, 16\},\
7373: };
```

```
7374: static XPoint seg2 3167[] = {
7375:
                                                         \{5, -5\}, \{7, -5\}, \{1, 16\},\
7376: };
7377: static XPoint seg3 3167[] = {
7378:
                                                           \{3,2\},\{3,-1\},\{2,-4\},\{0,-5\},\{-2,-5\},\{-5,-4\},\{-7,-1\},\{-8,2\},
7379:
                                                           \{-8,4\},\{-7,7\},\{-6,8\},\{-4,9\},\{-2,9\},\{0,8\},\{1,7\},\{2,5\},\{3,2\},
7380: };
7381: static XPoint seg4 3167[] = {
7382:
                                                          \{-5, -3\}, \{-6, -1\}, \{-7, 2\}, \{-7, 5\}, \{-6, 7\},
7383: };
7384: static XPoint seg5_3167[] = {
7385:
                                                           \{-2, -5\}, \{-4, -3\}, \{-5, -1\}, \{-6, 2\}, \{-6, 5\}, \{-5, 8\}, \{-4, 9\},
7386: };
7387: static XPoint seg6 3167[] = {
7388:
                                                          \{-4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, \{4, 16\}, 
7389: };
7390: static XPoint seg7 3167[] = {
7391:
                                                         \{0, 15\}, \{-3, 16\}, 
7392: };
7393: static XPoint seg8 3167[] = {
7394:
                                                          \{0, 14\}, \{-2, 16\},\
7395: };
7396: static XPoint seg9_3167[] = {
7397:
                                                          \{1, 14\}, \{2, 16\},\
7398: };
7399: static XPoint seg10 3167[] = {
7400:
                                                           \{0, 15\}, \{3, 16\},\
7401: };
7402: static XPoint *char3167[] = {
7403:
                                                           seg0_3167, seg1_3167, seg2_3167, seg3_3167, seg4_3167, seg5_3167,
7404:
                                                           seg6 3167, seg7 3167, seg8 3167, seg9 3167, seg10 3167,
7405:
                                                          NULL,
7406: };
7407: static int char p3167[] = {
7408:
                                                          XtNumber(seg0 3167), XtNumber(seg1 3167), XtNumber(seg2 3167),
7409:
                                                          XtNumber(seg3 3167), XtNumber(seg4 3167), XtNumber(seg5 3167),
7410:
                                                          XtNumber(seg6 3167), XtNumber(seg7 3167), XtNumber(seg8 3167),
7411:
                                                          XtNumber(seg9_3167), XtNumber(seg10_3167),
7412: };
7413: static XPoint seg0 3168[] = {
7414:
                                                           \{-8, -1\}, \{-7, -3\}, \{-5, -5\}, \{-3, -5\}, \{-2, -4\}, \{-1, -2\},
7415:
                                                           \{-1,2\},\{-3,9\},
7416: };
7417: static XPoint seg1 3168[] = {
7418:
                                                           \{-2, -4\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-4, 9\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, \{-2, 2\}, 
7419: };
7420: static XPoint seg2 3168[] = {
7421:
                                                           \{-2, -2\}, \{-3, 2\}, \{-5, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, \{-3, 9\}, 
7422: };
7423: static XPoint seg3 3168[] = {
7424:
                                                          \{7, -3\}, \{7, -4\}, \{6, -4\}, \{6, -2\}, \{8, -2\}, \{8, -4\}, \{7, -5\}, \{5, -5\},
```

```
7425:
                                                                                           \{3, -4\}, \{1, -2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{
7426: };
7427: static XPoint *char3168[] = {
7428:
                                                                                             seg0 3168, seg1 3168, seg2 3168, seg3 3168,
7429:
                                                                                           NULL,
7430: };
7431: static int char p3168[] = {
7432:
                                                                                           XtNumber(seg0 3168), XtNumber(seg1 3168), XtNumber(seg2 3168),
7433:
                                                                                           XtNumber(seg3 3168),
7434: };
7435: static XPoint seg0 3169[] = {
7436
                                                                                             \{6, -2\}, \{6, -3\}, \{5, -3\}, \{5, -1\}, \{7, -1\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{
7437:
                                                                                             \{6, -4\}, \{3, -5\}, \{0, -5\}, \{-3, -4\}, \{-4, -3\}, \{-4, -1\}, \{-3, 1\}, \{-1, 2\},
7438:
                                                                                           \{2,3\},\{4,4\},\{5,6\},\
7439: };
7440: static XPoint seg1_3169[] = {
7441:
                                                                                         \{-3, -4\}, \{-4, -1\},
7442: };
7443: static XPoint seg2 3169[] = {
7444:
                                                                                         \{-3,0\},\{-1,1\},\{2,2\},\{4,3\},
7445: };
7446: static XPoint seg3 3169[] = {
7447:
                                                                                           \{5,4\},\{4,8\},
7448: };
7449: static XPoint seg4 3169[] = {
7450:
                                                                                             \{-4, -3\}, \{-3, -1\}, \{-1, 0\}, \{2, 1\}, \{4, 2\}, \{5, 4\}, \{5, 6\}, \{4, 8\},
7451:
                                                                                             \{1,9\},\{-2,9\},\{-5,8\},\{-6,7\},\{-6,5\},\{-4,5\},\{-4,7\},\{-5,7\},
7452:
                                                                                           \{-5,6\},\
7453: };
7454: static XPoint *char3169[] = {
7455:
                                                                                             seg0_3169, seg1_3169, seg2_3169, seg3_3169, seg4_3169,
                                                                                           NULL,
7456:
7457: };
7458: static int char p3169[] = {
7459:
                                                                                           XtNumber(seg0_3169),XtNumber(seg1_3169),XtNumber(seg2_3169),
7460:
                                                                                           XtNumber(seg3 3169),XtNumber(seg4 3169),
7461: };
7462: static XPoint seg0 3170[] = {
7463:
                                                                                             \{2, -12\}, \{-1, -1\}, \{-2, 3\}, \{-2, 6\}, \{-1, 8\}, \{0, 9\}, \{2, 9\}, \{4, 7\}, \{-2, 3\}, \{-2, 6\}, \{-1, 8\}, \{0, 9\}, \{2, 9\}, \{4, 7\}, \{-2, 9\}, \{-1, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 
7464:
                                                                                             {5,5},
7465: };
7466: static XPoint seg1 3170[] = {
7467:
                                                                                           \{3, -12\}, \{0, -1\}, \{-1, 3\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, \{-1, 8\}, 
7468: };
7469: static XPoint seg2 3170[] = {
7470:
                                                                                           \{2, -12\}, \{4, -12\}, \{0, 2\}, \{-1, 6\}, \{0, 2\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{-1, 6\}, \{
7471: };
7472: static XPoint seg3 3170[] = {
7473:
                                                                                           \{-4, -5\}, \{6, -5\},\
7474: };
7475: static XPoint *char3170[] = {
7476:
                                                                                             seg0 3170, seg1 3170, seg2 3170, seg3 3170,
7477:
                                                                                           NULL,
7478: };
```

```
7479: static int char p3170[] = {
7480:
                                                            XtNumber(seg0_3170),XtNumber(seg1_3170),XtNumber(seg2_3170),
7481:
                                                             XtNumber(seg3 3170),
7482: };
7483: static XPoint seg0 3171[] = {
7484:
                                                             \{-11, -1\}, \{-10, -3\}, \{-8, -5\}, \{-6, -5\}, \{-5, -4\}, \{-4, -2\}, \{-10, -3\}, \{-10, -3\}, \{-2, -2\}, \{-2, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, 
7485:
                                                               \{-4,1\},\{-6,6\},
7486: };
7487: static XPoint seg1 3171[] = {
7488:
                                                             \{-5, -4\}, \{-5, 0\}, \{-6, 4\}, \{-6, 8\},
7489: };
7490: static XPoint seg2 3171[] = {
7491:
                                                            \{-5, -2\}, \{-7, 3\}, \{-7, 6\}, \{-6, 8\}, \{-4, 9\}, \{-2, 9\}, \{0, 8\}, \{2, 6\},
7492:
                                                             {4,3},
7493: };
7494: static XPoint seg3 3171[] = {
7495:
                                                             \{6, -5\}, \{4, 3\}, \{4, 6\}, \{5, 8\}, \{6, 9\}, \{8, 9\}, \{10, 7\}, \{11, 5\},
7496: };
7497: static XPoint seg4 3171[] = {
7498:
                                                           \{7, -5\}, \{5, 3\}, \{5, 8\},
7499: };
7500: static XPoint seg5 3171[] = {
7501:
                                                             \{6, -5\}, \{8, -5\}, \{6, 2\}, \{5, 6\},\
7502: };
7503: static XPoint *char3171[] = {
7504:
                                                               seg0_3171, seg1_3171, seg2_3171, seg3_3171, seg4_3171, seg5_3171,
7505:
                                                             NULL,
7506: };
7507: static int char p3171[] = {
7508:
                                                             XtNumber(seg0 3171), XtNumber(seg1 3171), XtNumber(seg2 3171),
7509:
                                                             XtNumber(seg3 3171), XtNumber(seg4 3171), XtNumber(seg5 3171),
7510: };
7511: static XPoint seg0_3172[] = {
7512:
                                                           \{-9, -1\}, \{-8, -3\}, \{-6, -5\}, \{-4, -5\}, \{-4, -5\}, \{-2, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3,
7513:
                                                               \{-3, -4\}, \{-2, -2\}, \{-2, 1\}, \{-4, 6\},
7514: };
7515: static XPoint seg1 3172[] = {
                                                             \{-3, -4\}, \{-3, 0\}, \{-4, 4\}, \{-4, 8\},
7516:
7517: };
7518: static XPoint seg2_3172[] = {
7519:
                                                             \{-3, -2\}, \{-5, 3\}, \{-5, 6\}, \{-4, 8\}, \{-2, 9\}, \{0, 9\}, \{2, 8\}, \{4, 6\}, \{-3, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5, -2\}, \{-5,
7520:
                                                               \{6,3\},\{7,-1\},\{7,-5\},\{6,-5\},\{6,-4\},\{7,-2\},
7521: };
7522: static XPoint *char3172[] = {
7523:
                                                             seg0 3172, seg1 3172, seg2 3172,
7524:
                                                             NULL,
7525: };
```

```
7526: static int char p3172[] = {
7527:
                                                      XtNumber(seg0_3172),XtNumber(seg1_3172),XtNumber(seg2_3172),
7528: };
7529: static XPoint seg0 3173[] = {
7530:
                                                       \{-14, -1\}, \{-13, -3\}, \{-11, -5\},
7531:
                                                       \{-9, -5\}, \{-8, -4\}, \{-7, -2\}, \{-7, 1\}, \{-9, 6\},\
7532: };
7533: static XPoint seg1 3173[] = {
7534:
                                                       \{-8, -4\}, \{-8, 0\}, \{-9, 4\}, \{-9, 8\},
7535: };
7536: static XPoint seg2 3173[] = {
                                                       \{-8, -2\}, \{-10, 3\}, \{-10, 6\}, \{-9, 8\}, \{-7, 9\}, \{-5, 9\}, \{-3, 8\}, \{-1, 6\},
7537:
7538:
                                                       \{0,3\},\
7539: };
7540: static XPoint seg3 3173[] = {
7541:
                                                       \{2, -5\}, \{0, 3\}, \{0, 6\}, \{1, 8\}, \{3, 9\}, \{5, 9\}, \{7, 8\}, \{9, 6\},
7542:
                                                       \{11,3\},\{12,-1\},\{12,-5\},\{11,-5\},\{11,-4\},\{12,-2\},
7543: };
7544: static XPoint seg4 3173[] = {
7545:
                                                     \{3, -5\}, \{1, 3\}, \{1, 8\},\
7546: };
7547: static XPoint seq5 3173[] = {
7548:
                                                       \{2, -5\}, \{4, -5\}, \{2, 2\}, \{1, 6\},\
7549: };
7550: static XPoint *char3173[] = {
7551:
                                                       seg0_3173, seg1_3173, seg2_3173, seg3_3173, seg4_3173, seg5_3173,
7552:
                                                       NULL,
7553: };
7554: static int char p3173[] = {
                                                       XtNumber(seg0 3173), XtNumber(seg1 3173), XtNumber(seg2 3173),
7555:
7556:
                                                      XtNumber(seg3 3173), XtNumber(seg4 3173), XtNumber(seg5 3173),
7557: };
7558: static XPoint seg0 3174[] = {
7559:
                                                      \{-8, -1\}, \{-6, -4\}, \{-4, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2, -5\}, \{-2,
7560:
                                                      \{0, -4\}, \{1, -2\}, \{1, 0\},\
7561: };
7562: static XPoint seg1 3174[] = {
7563:
                                                       \{-2, -5\}, \{-1, -4\}, \{-1, 0\}, \{-2, 4\}, \{-3, 6\}, \{-5, 8\}, \{-7, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\},
7564:
                                                       \{-10,8\},\{-10,6\},\{-8,6\},\{-8,8\},\{-9,8\},\{-9,7\},
7565: };
7566: static XPoint seg2_3174[] = {
7567:
                                                       \{0, -3\}, \{0, 0\}, \{-1, 4\}, \{-1, 7\}, \{0, 0\}, \{-1, 4\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1, 7\}, \{-1
7568: };
7569: static XPoint seg3 3174[] = {
                                                       \{8, -3\}, \{8, -4\}, \{7, -4\}, \{7, -2\}, \{9, -2\}, \{9, -4\}, \{8, -5\}, \{6, -5\},
7570:
7571:
                                                       \{4, -4\}, \{2, -2\}, \{1, 0\}, \{0, 4\}, \{0, 8\}, \{1, 9\},
7572: };
7573: static XPoint seg4 3174[] = {
7574:
                                                       \{-2,4\},\{-2,6\},\{-1,8\},\{1,9\},\{3,9\},\{5,8\},\{7,5\},
7575: };
7576: static XPoint *char3174[] = {
7577:
                                                       seg0 3174, seg1 3174, seg2 3174, seg3 3174, seg4 3174,
7578:
                                                      NULL,
7579: };
```

```
7580: static int char p3174[] = {
                                            XtNumber(seg0 3174),XtNumber(seg1_3174),XtNumber(seg2_3174),
7581:
                                            XtNumber(seg3_3174),XtNumber(seg4_3174),
7582:
7583: };
7584: static XPoint seg0 3175[] = {
7585:
                                             \{-10, -1\},\
7586:
                                             \{-9, -3\}, \{-7, -5\}, \{-5, -5\}, \{-4, -4\}, \{-3, -2\}, \{-3, 1\}, \{-5, 6\},
7587: };
7588: static XPoint seq1 3175[] = {
7589:
                                            \{-4, -4\}, \{-4, 0\}, \{-5, 4\}, \{-5, 8\},
7590: };
7591: static XPoint seg2 3175[] = {
7592:
                                             \{-4, -2\}, \{-6, 3\}, \{-6, 6\}, \{-5, 8\}, \{-3, 9\}, \{-1, 9\}, \{1, 8\}, \{3, 6\},
7593:
                                             {5,2},
7594: };
7595: static XPoint seg3 3175[] = {
7596:
                                             \{7, -5\}, \{3, 9\}, \{2, 12\}, \{0, 15\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\}, \{-2, 16\},
7597: };
7598: static XPoint seg4 3175[] = {
7599:
                                            \{8, -5\}, \{4, 9\}, \{2, 13\},
7600: };
7601: static XPoint seg5 3175[] = {
7602:
                                             \{7, -5\}, \{9, -5\}, \{5, 9\}, \{3, 13\}, \{1, 15\}, \{-2, 16\}, \{-5, 16\}, \{-7, 15\},
                                             \{-8,14\},\{-8,12\},\{-6,12\},\{-6,14\},\{-7,14\},\{-7,13\},
7603:
7604: };
7605: static XPoint *char3175[] = {
7606:
                                            seg0 3175, seg1 3175, seg2 3175, seg3 3175, seg4 3175, seg5 3175,
                                            NULL,
7607:
7608: };
7609: static int char p3175[] = {
7610:
                                            XtNumber(seg0 3175), XtNumber(seg1 3175), XtNumber(seg2 3175),
7611:
                                            XtNumber(seg3 3175), XtNumber(seg4 3175), XtNumber(seg5 3175),
7612: };
7613: static XPoint seg0 3176[] = {
7614:
                                             \{7, -5\}, \{6, -3\}, \{4, -1\}, \{6, -3\}, \{4, -1\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{7, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{6, -3\}, \{
7615:
                                             \{-4,5\},\{-6,7\},\{-7,9\},
7616: };
7617: static XPoint seg1 3176[] = {
                                            \{6, -3\}, \{-3, -3\}, \{-5, -2\}, \{-6, 0\},\
7618:
7619: };
7620: static XPoint seg2 3176[] = {
7621:
                                            \{4, -3\}, \{0, -4\}, \{-3, -4\}, \{-4, -3\},
7622: };
7623: static XPoint seg3 3176[] = {
                                             \{4, -3\}, \{0, -5\}, \{-3, -5\}, \{-5, -3\}, \{-6, 0\},\
7624:
7625: };
7626: static XPoint seg4 3176[] = {
7627:
                                             \{-6,7\},\{3,7\},\{5,6\},\{6,4\},
7628: };
```

```
7629: static XPoint seg5 3176[] = {
7630:
                                          \{-4,7\},\{0,8\},\{3,8\},\{4,7\},
7631: };
7632: static XPoint seg6 3176[] = {
7633:
                                          \{-4,7\},\{0,9\},\{3,9\},\{5,7\},\{6,4\},
7634: };
7635: static XPoint *char3176[] = {
                                          seg0 3176, seg1 3176, seg2_3176, seg3_3176, seg4_3176, seg5_3176,
7636:
7637:
                                          seg6 3176,
7638:
                                          NULL,
7639: };
7640: static int char p3176[] = {
7641:
                                          XtNumber(seg0 3176), XtNumber(seg1 3176), XtNumber(seg2 3176),
7642:
                                          XtNumber(seg3 3176), XtNumber(seg4 3176), XtNumber(seg5 3176),
7643:
                                          XtNumber(seg6 3176),
7644: };
7645: static XPoint seg0_3199[] = {
7646:
                                          {-8,8},
7647: };
7648: static XPoint *char3199[] = {
7649:
                                          seg0 3199,
7650:
                                          NULL,
7651: };
7652: static int char p3199[] = {
7653:
                                          XtNumber(seg0 3199),
7654: };
7655: static XPoint seq0 3200[] = {
7656:
                                          \{-1, -12\}, \{-4, -11\}, \{-6, -8\}, \{-7, -3\}, \{-7, 0\}, \{-6, 5\}, \{-4, 8\}, \{-7, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-6, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -12\}, \{-7, -
7657:
                                          \{-1,9\},\{1,9\},\{4,8\},\{6,5\},\{7,0\},\{7,-3\},\{6,-8\},\{4,-11\},
7658:
                                          \{1, -12\}, \{-1, -12\},\
7659: };
7660: static XPoint seg1 3200[] = {
7661:
                                          \{-4, -10\}, \{-5, -8\}, \{-6, -4\}, \{-6, 1\}, \{-5, 5\}, \{-4, 7\},
7662: };
7663: static XPoint seg2_3200[] = {
7664:
                                          \{4,7\},\{5,5\},\{6,1\},\{6,-4\},\{5,-8\},\{4,-10\},
7665: };
7666: static XPoint seg3 3200[] = {
7667:
                                          \{-1, -12\}, \{-3, -11\}, \{-4, -9\}, \{-5, -4\}, \{-5, 1\}, \{-4, 6\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3, 8\}, \{-3,
7668:
                                          \{-1,9\},\
7669: };
7670: static XPoint seg4 3200[] = {
7671:
                                          \{1,9\},\{3,8\},\{4,6\},\{5,1\},\{5,-4\},\{4,-9\},\{3,-11\},\{1,-12\},
7672: };
7673: static XPoint *char3200[] = {
7674:
                                          seg0 3200, seg1 3200, seg2 3200, seg3 3200, seg4 3200,
7675:
                                          NULL.
7676: };
7677: static int char p3200[] = {
7678:
                                          XtNumber(seg0 3200), XtNumber(seg1 3200), XtNumber(seg2 3200),
7679:
                                          XtNumber(seg3 3200), XtNumber(seg4 3200),
7680: };
```

```
Listing C.4 Continued
```

```
7681: static XPoint seg0_3201[] = {
7682:
                                                           \{-1, -10\}, \{-1, 9\}, 
7683: };
7684: static XPoint seg1 3201[] = {
7685:
                                                           \{0, -10\}, \{0, 8\},\
7686: };
7687: static XPoint seg2 3201[] = {
7688:
                                                          \{1, -12\}, \{1, 9\}, 
7689: };
7690: static XPoint seg3 3201[] = {
7691:
                                                          \{1, -12\}, \{-2, -9\}, \{-4, -8\},
7692: };
7693: static XPoint seg4_3201[] = {
7694:
                                                           \{-5,9\},\{5,9\},
7695: };
7696: static XPoint seg5 3201[] = {
7697:
                                                           \{-1,8\},\{-3,9\},
7698: };
7699: static XPoint seg6 3201[] = {
7700:
                                                           \{-1,7\},\{-2,9\},
7701: };
7702: static XPoint seg7 3201[] = {
7703:
                                                          \{1,7\},\{2,9\},\
7704: };
7705: static XPoint seg8 3201[] = {
7706:
                                                           \{1,8\},\{3,9\},\
7707: };
7708: static XPoint *char3201[] = {
7709:
                                                           seg0_3201, seg1_3201, seg2_3201, seg3_3201, seg4_3201, seg5_3201,
7710:
                                                            seg6_3201, seg7_3201, seg8_3201,
7711:
                                                           NULL,
7712: };
7713: static int char_p3201[] = {
7714:
                                                           XtNumber(seg0 3201),XtNumber(seg1_3201),XtNumber(seg2_3201),
7715:
                                                            XtNumber(seg3_3201), XtNumber(seg4_3201), XtNumber(seg5_3201),
                                                           XtNumber(seg6_3201),XtNumber(seg7_3201),XtNumber(seg8_3201),
7716:
7717: };
7718: static XPoint seg0_3202[] = {
7719:
                                                           \{-6, -8\}, \{-6, -7\}, \{-5, -7\}, \{-5, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6,
7720: };
7721: static XPoint seg1_3202[] = {
7722:
                                                            \{-6, -9\}, \{-5, -9\}, \{-4, -8\}, \{-4, -7\}, \{-5, -6\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7,
7723:
                                                            \{-7, -8\}, \{-6, -10\}, \{-5, -11\}, \{-2, -12\}, \{2, -12\}, \{5, -11\}, \{6, -10\}, 
7724:
                                                            \{7, -8\}, \{7, -6\}, \{6, -4\}, \{3, -2\}, \{-2, 0\}, \{-4, 1\}, \{-6, 3\}, \{-7, 6\},
7725:
                                                            {-7,9},
7726: };
7727: static XPoint seg2_3202[] = {
7728:
                                                           \{5, -10\}, \{6, -8\}, \{6, -6\}, \{5, -4\},\
7729: };
7730: static XPoint seg3 3202[] = {
7731:
                                                            \{2, -12\}, \{4, -11\}, \{5, -8\}, \{5, -6\}, \{4, -4\}, \{2, -2\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\}, \{-2, 0\},
7732: };
```

```
7733: static XPoint seg4 3202[] = {
7734:
                                                                       \{-7,7\},\{-6,6\},\{-4,6\},\{1,7\},\{5,7\},\{7,6\},
7735: };
7736: static XPoint seg5 3202[] = {
7737:
                                                                      \{-4,6\},\{1,8\},\{5,8\},\{6,7\},
7738: };
7739: static XPoint seg6 3202[] = {
7740:
                                                                      \{-4,6\},\{1,9\},\{5,9\},\{6,8\},\{7,6\},\{7,4\},
7741: };
7742: static XPoint *char3202[] = {
7743:
                                                                      seg0 3202, seg1 3202, seg2 3202, seg3 3202, seg4 3202, seg5 3202,
7744:
                                                                      seg6 3202,
7745:
                                                                      NULL,
7746: };
7747: static int char p3202[] = {
7748:
                                                                      XtNumber(seg0_3202),XtNumber(seg1_3202),XtNumber(seg2_3202),
7749:
                                                                      XtNumber(seg3_3202),XtNumber(seg4_3202),XtNumber(seg5_3202),
7750:
                                                                      XtNumber(seg6 3202),
7751: };
7752: static XPoint seg0 3203[] = {
7753:
                                                                      \{-6, -8\}, \{-6, -7\},\
7754:
                                                                       \{-5, -7\}, \{-5, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-6, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8, -8\}, \{-8,
7755: };
7756: static XPoint seg1 3203[] = {
7757:
                                                                      \{-6, -9\}, \{-5, -9\}, \{-4, -8\}, \{-4, -7\}, \{-5, -6\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-6, -6\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7, -7\}, \{-7,
7758:
                                                                       \{-7, -8\}, \{-6, -10\}, \{-5, -11\}, \{-2, -12\}, \{2, -12\}, \{5, -11\}, \{6, -9\},
7759:
                                                                       \{6, -6\}, \{5, -4\}, \{2, -3\}, \{2, -3\}, \{2, -3\}, \{2, -3\}, \{2, -3\}, \{2, -3\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{
7760: };
7761: static XPoint seg2 3203[] = {
7762:
                                                                       \{4, -11\}, \{5, -9\}, \{5, -6\}, \{4, -4\},
7763: };
7764: static XPoint seg3 3203[] = {
7765:
                                                                      \{1, -12\}, \{3, -11\}, \{4, -9\}, \{4, -6\}, \{3, -4\}, \{1, -3\},
7766: };
7767: static XPoint seg4_3203[] = {
7768:
                                                                      \{-1, -3\}, \{2, -3\}, \{4, -2\}, \{6, 0\}, \{7, 2\}, \{7, 5\}, \{6, 7\}, \{5, 8\},
7769:
                                                                       \{2,9\},\{-2,9\},\{-5,8\},\{-6,7\},\{-7,5\},\{-7,4\},\{-6,3\},\{-5,3\},
7770:
                                                                      \{-4,4\},\{-4,5\},\{-5,6\},\{-6,6\},
7771: };
7772: static XPoint seq5 3203[] = {
7773:
                                                                     \{5,0\},\{6,2\},\{6,5\},\{5,7\},
7774: };
7775: static XPoint seg6 3203[] = {
7776:
                                                                      \{1, -3\}, \{3, -2\}, \{4, -1\}, \{5, 2\}, \{5, 5\}, \{4, 8\}, \{2, 9\}, \{5, 5\}, \{4, 8\}, \{2, 9\}, \{4, 8\}, \{2, 9\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{4, 8\}, \{
7777: };
7778: static XPoint seg7 3203[] = {
7779:
                                                                      \{-6,4\},\{-6,5\},\{-5,5\},\{-5,4\},\{-6,4\},
7780: };
7781: static XPoint *char3203[] = {
7782:
                                                                       seg0 3203, seg1 3203, seg2 3203, seg3 3203, seg4 3203, seg5 3203,
7783:
                                                                       seg6 3203, seg7 3203,
7784:
                                                                      NULL,
7785: };
```

```
7786: static int char p3203[] = {
7787:
           XtNumber(seg0 3203), XtNumber(seg1 3203), XtNumber(seg2 3203),
7788:
           XtNumber(seg3_3203),XtNumber(seg4_3203),XtNumber(seg5_3203),
7789:
           XtNumber(seg6 3203), XtNumber(seg7 3203),
7790: };
7791: static XPoint seg0 3204[] = {
7792:
           \{1, -9\}, \{1, 9\}, 
7793: };
7794: static XPoint seg1 3204[] = {
7795:
           \{2, -10\}, \{2, 8\},\
7796: };
7797: static XPoint seg2_3204[] = {
7798:
           \{3, -12\}, \{3, 9\},\
7799: };
7800: static XPoint seg3 3204[] = {
7801:
           \{3, -12\}, \{-8, 3\}, \{8, 3\}, 
7802: };
7803: static XPoint seg4 3204[] = {
7804:
           \{-2,9\},\{6,9\},\
7805: };
7806: static XPoint seg5 3204[] = {
           \{1,8\},\{-1,9\},\
7807:
7808: };
7809: static XPoint seg6 3204[] = {
7810:
           \{1,7\},\{0,9\},\
7811: };
7812: static XPoint seg7_3204[] = {
7813:
           \{3,7\},\{4,9\},\
7814: };
7815: static XPoint seg8_3204[] = {
7816:
           \{3,8\},\{5,9\},\
7817: };
7818: static XPoint *char3204[] = {
7819:
           seg0 3204, seg1 3204, seg2 3204, seg3 3204, seg4 3204, seg5 3204,
7820:
           seg6_3204, seg7_3204, seg8_3204,
           NULL,
7821:
7822: };
7823: static int char p3204[] = {
7824:
           XtNumber(seg0 3204), XtNumber(seg1 3204), XtNumber(seg2 3204),
7825:
           XtNumber(seg3 3204), XtNumber(seg4 3204), XtNumber(seg5 3204),
7826:
           XtNumber(seg6 3204), XtNumber(seg7 3204), XtNumber(seg8 3204),
7827: };
7828: static XPoint seg0 3205[] = {
7829:
           \{-5, -12\}, \{-7, -2\}, \{-5, -4\}, \{-2, -5\}, \{1, -5\}, \{4, -4\},
7830:
           \{6, -2\}, \{7, 1\}, \{7, 3\}, \{6, 6\}, \{4, 8\}, \{1, 9\}, \{-2, 9\}, \{-5, 8\}, \{-6, 7\},
7831:
           \{-7,5\},\{-7,4\},\{-6,3\},\{-5,3\},\{-4,4\},\{-4,5\},\{-5,6\},\{-6,6\},
7832: };
7833: static XPoint seg1 3205[] = {
7834:
           \{5, -2\}, \{6, 0\}, \{6, 4\}, \{5, 6\},\
7835: };
```

```
7836: static XPoint seg2 3205[] = {
7837:
                                           \{1, -5\}, \{3, -4\}, \{4, -3\}, \{5, 0\}, \{5, 4\}, \{4, 7\}, \{3, 8\}, \{1, 9\},
7838: };
7839: static XPoint seg3 3205[] = {
7840:
                                           \{-6,4\},\{-6,5\},\{-5,5\},\{-5,4\},\{-6,4\},
7841: };
7842: static XPoint seg4 3205[] = {
7843:
                                          \{-5, -12\}, \{5, -12\}, 
7844: };
7845: static XPoint seg5 3205[] = {
7846:
                                         \{-5, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3,
7847: };
7848: static XPoint seg6 3205[] = {
7849:
                                           \{-5, -10\}, \{-1, -10\}, \{3, -11\}, \{5, -12\},
7850: };
7851: static XPoint *char3205[] = {
7852:
                                           seg0 3205, seg1 3205, seg2 3205, seg3 3205, seg4 3205, seg5 3205,
7853:
                                           seg6 3205,
                                          NULL,
7854:
7855: };
7856: static int char p3205[] = {
7857:
                                          XtNumber(seg0 3205), XtNumber(seg1 3205), XtNumber(seg2 3205),
7858:
                                           XtNumber(seg3_3205),XtNumber(seg4_3205),XtNumber(seg5_3205),
7859:
                                          XtNumber(seg6 3205),
7860: };
7861: static XPoint seg0 3206[] = {
                                           \{4, -9\}, \{4, -8\}, \{5, -8\}, \{5, -9\},\
7862:
7863:
                                           {4,-9},
7864: };
7865: static XPoint seg1 3206[] = {
7866:
                                           \{5, -10\}, \{4, -10\}, \{3, -9\}, \{3, -8\}, \{4, -7\}, \{5, -7\}, \{6, -8\}, \{6, -9\},
7867:
                                           \{5, -11\}, \{3, -12\}, \{0, -12\}, \{-3, -11\}, \{-5, -9\}, \{-6, -7\}, \{-7, -3\},
7868:
                                           \{-7,3\}, \{-6,6\}, \{-4,8\}, \{-1,9\}, \{1,9\}, \{4,8\}, \{6,6\}, \{7,3\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,2\}, \{7,
7869:
                                           \{6, -1\}, \{4, -3\}, \{1, -4\}, \{-1, -4\}, \{-3, -3\}, \{-4, -2\}, \{-5, 0\},
7870: };
7871: static XPoint seg2 3206[] = {
7872:
                                          \{-4, -9\}, \{-5, -7\}, \{-6, -3\}, \{-6, 3\}, \{-5, 6\}, \{-4, 7\},
7873: };
7874: static XPoint seg3 3206[] = {
7875:
                                          \{5,6\},\{6,4\},\{6,1\},\{5,-1\},
7876: };
7877: static XPoint seg4 3206[] = {
                                           \{0, -12\}, \{-2, -11\}, \{-3, -10\}, \{-4, -8\}, \{-5, -4\}, \{-5, 3\}, \{-4, 6\},
7878:
7879:
                                           \{-3,8\},\{-1,9\},
7880: };
7881: static XPoint seg5 3206[] = {
7882:
                                           \{1,9\},\{3,8\},\{4,7\},\{5,4\},\{5,1\},\{4,-2\},\{3,-3\},\{1,-4\},
7883: };
7884: static XPoint *char3206[] = {
7885:
                                           seg0 3206, seg1 3206, seg2 3206, seg3 3206, seg4 3206, seg5 3206,
7886:
                                           NULL,
7887: };
```

```
Listing C.4 Continued
```

```
7888: static int char p3206[] = {
                                                                                        XtNumber(seg0 3206), XtNumber(seg1 3206), XtNumber(seg2 3206),
7889:
7890:
                                                                                        XtNumber(seg3 3206), XtNumber(seg4 3206), XtNumber(seg5 3206),
7891: };
7892: static XPoint seg0 3207[] = {
7893:
                                                                                        \{-7, -12\}, \{-7, -6\},\
7894: };
7895: static XPoint seg1 3207[] = {
                                                                                        \{7, -12\}, \{7, -9\}, \{6, -6\}, \{2, -1\}, \{1, 1\}, \{0, 5\}, \{0, 9\},
7896:
7897: };
7898: static XPoint seg2 3207[] = {
7899:
                                                                                          \{1,0\},\{0,2\},\{-1,5\},\{-1,9\},
7900: };
7901: static XPoint seg3 3207[] = {
7902:
                                                                                        \{6, -6\}, \{1, -1\}, \{-1, 2\}, \{-2, 5\}, \{-2, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\},
7903: };
7904: static XPoint seg4 3207[] = {
                                                                                        \{-7, -8\}, \{-6, -10\}, \{-4, -12\}, \{-2, -12\}, \{3, -9\}, \{5, -9\}, \{6, -10\},
7905:
7906:
                                                                                          \{7, -12\},\
7907: };
7908: static XPoint seg5 3207[] = {
7909:
                                                                                        \{-5, -10\}, \{-4, -11\}, \{-2, -11\}, \{0, -10\}, \{-2, -11\}, \{0, -10\}, \{-2, -11\}, \{-2, -11\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, 
7910: };
7911: static XPoint seg6 3207[] = {
7912:
                                                                                        \{-7, -8\}, \{-6, -9\}, \{-4, -10\}, \{-2, -10\}, \{3, -9\}, \{-7, -8\}, \{-6, -9\}, \{-4, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2, -10\}, \{-2,
7913: };
7914: static XPoint *char3207[] = {
7915:
                                                                                          seg0 3207, seg1 3207, seg2 3207, seg3 3207, seg4 3207, seg5 3207,
7916:
                                                                                          seg6 3207,
7917:
                                                                                        NULL,
7918: };
7919: static int char p3207[] = {
7920:
                                                                                        XtNumber(seg0 3207), XtNumber(seg1 3207), XtNumber(seg2 3207),
7921:
                                                                                        XtNumber(seg3_3207),XtNumber(seg4_3207),XtNumber(seg5_3207),
7922:
                                                                                        XtNumber(seg6 3207),
7923: };
7924: static XPoint seg0 3208[] = {
7925:
                                                                                          \{-2, -12\}, \{-5, -11\}, \{-6, -9\}, \{-6, -6\}, \{-5, -4\}, \{-2, -3\}, \{2, -3\}, \{2, -3\}, \{-2, -3\}, \{-2, -3\}, \{-2, -3\}, \{-2, -3\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3,
7926:
                                                                                          \{5, -4\}, \{6, -6\}, \{6, -9\}, \{5, -11\}, \{2, -12\}, \{-2, -12\}, 
7927: };
7928: static XPoint seg1 3208[] = {
7929:
                                                                                          \{-4, -11\}, \{-5, -9\}, \{-5, -6\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4, -4\}, \{-4
7930: };
7931: static XPoint seg2 3208[] = {
7932:
                                                                                          \{4, -4\}, \{5, -6\}, \{5, -9\}, \{4, -11\},
7933: };
7934: static XPoint seg3 3208[] = {
                                                                                          \{-2, -12\}, \{-3, -11\}, \{-4, -9\}, \{-4, -6\}, \{-3, -4\}, \{-2, -3\},
7935:
7936: };
```

```
7937: static XPoint seg4 3208[] = {
7938:
                                                                  \{2, -3\}, \{3, -4\}, \{4, -6\}, \{4, -9\}, \{3, -11\}, \{2, -12\}, \{3, -11\}, \{2, -12\}, \{3, -11\}, \{2, -12\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}
7939: };
7940: static XPoint seg5 3208[] = {
7941:
                                                                  \{-2, -3\}, \{-5, -2\}, \{-6, -1\}, \{-7, 1\}, \{-7, 5\}, \{-6, 7\}, \{-5, 8\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}
7942:
                                                                  \{2,9\},\{5,8\},\{6,7\},\{7,5\},\{7,1\},\{6,-1\},\{5,-2\},\{2,-3\},
7943: };
7944: static XPoint seq6 3208[] = {
7945:
                                                                  \{-5, -1\}, \{-6, 1\}, \{-6, 5\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, \{-5, 7\}, 
7946: };
7947: static XPoint seg7 3208[] = {
7948:
                                                                  \{5,7\},\{6,5\},\{6,1\},\{5,-1\},
7949: };
7950: static XPoint seg8 3208[] = {
7951:
                                                                  \{-2, -3\}, \{-4, -2\}, \{-5, 1\}, \{-5, 5\}, \{-4, 8\}, \{-2, 9\},
7952: };
7953: static XPoint seg9 3208[] = {
7954:
                                                                 \{2,9\},\{4,8\},\{5,5\},\{5,1\},\{4,-2\},\{2,-3\},
7955: };
7956: static XPoint *char3208[] = {
7957:
                                                                  seg0 3208, seg1 3208, seg2 3208, seg3 3208, seg4 3208, seg5 3208,
7958:
                                                                  seg6 3208, seg7 3208, seg8 3208, seg9 3208,
7959:
                                                                 NULL,
7960: };
7961: static int char p3208[] = {
                                                                 XtNumber(seg0_3208),XtNumber(seg1_3208),XtNumber(seg2_3208),
7962:
7963:
                                                                 XtNumber(seq3 3208), XtNumber(seq4 3208), XtNumber(seq5 3208),
7964:
                                                                 XtNumber(seg6 3208), XtNumber(seg7 3208), XtNumber(seg8 3208),
7965:
                                                                 XtNumber(seg9 3208),
7966: };
7967: static XPoint seg0 3209[] = {
7968:
                                                                 \{-5,5\},\{-5,6\},\
7969:
                                                                  \{-4,6\},\{-4,5\},\{-5,5\},
7970: };
7971: static XPoint seg1_3209[] = {
7972:
                                                                  \{5, -3\}, \{4, -1\}, \{3, 0\}, \{1, 1\}, \{-1, 1\}, \{-4, 0\}, \{-6, -2\}, \{-7, -5\},
7973:
                                                                  \{-7, -6\}, \{-6, -9\}, \{-4, -11\}, \{-1, -12\}, \{1, -12\}, \{4, -11\}, \{6, -9\}, \{-7, -6\}, \{-6, -9\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -
7974:
                                                                  \{7, -6\}, \{7, 0\}, \{6, 4\}, \{5, 6\}, \{3, 8\}, \{0, 9\}, \{-3, 9\}, \{-5, 8\}, \{-6, 6\},
7975:
                                                                  \{-6,5\},\{-5,4\},\{-4,4\},\{-3,5\},\{-3,6\},\{-4,7\},\{-5,7\},
7976: };
7977: static XPoint seg2 3209[] = {
7978:
                                                                  \{-5, -2\}, \{-6, -4\}, \{-6, -7\}, \{-5, -9\},
7979: };
7980: static XPoint seg3 3209[] = {
7981:
                                                                  \{4, -10\}, \{5, -9\}, \{6, -6\}, \{6, 0\}, \{5, 4\}, \{4, 6\},
7982: };
7983: static XPoint seg4_3209[] = {
7984:
                                                                  \{-1,1\},\{-3,0\},\{-4,-1\},\{-5,-4\},\{-5,-7\},\{-4,-10\},\{-3,-11\},
7985:
                                                                  {-1,-12},
7986: };
7987: static XPoint seg5 3209[] = {
7988:
                                                                  \{1, -12\}, \{3, -11\}, \{4, -9\}, \{5, -6\}, \{5, 1\}, \{4, 5\}, \{3, 7\}, \{2, 8\},
7989:
                                                                  \{0,9\},\
7990: };
```

```
7991: static XPoint *char3209[] = {
7992:
                                              seg0_3209, seg1_3209, seg2_3209, seg3_3209, seg4_3209, seg5_3209,
7993:
                                             NULL,
7994: };
7995: static int char p3209[] = {
7996:
                                             XtNumber(seg0_3209),XtNumber(seg1_3209),XtNumber(seg2_3209),
7997:
                                             XtNumber(seg3 3209), XtNumber(seg4 3209), XtNumber(seg5 3209),
7998: };
7999: static XPoint seg0 3210[] = {
8000:
                                              \{0,6\},\{-1,7\},\{-1,8\},\{0,9\},\{1,9\},\{2,8\},\{2,7\},\{1,6\},
8001:
                                              {0,6},
8002: };
8003: static XPoint seg1 3210[] = {
8004:
                                              \{0,7\},\{0,8\},\{1,8\},\{1,7\},\{0,7\},
8005: };
8006: static XPoint *char3210[] = {
8007:
                                              seg0 3210, seg1 3210,
8008:
                                             NULL,
8009: };
8010: static int char p3210[] = {
8011:
                                             XtNumber(seg0_3210),XtNumber(seg1_3210),
8012: };
8013: static XPoint seg0_3211[] = {
8014:
                                              \{2,8\},\{1,9\},\{0,9\},
8015:
                                              \{-1,8\},\{-1,7\},\{0,6\},\{1,6\},\{2,7\},\{2,10\},\{1,12\},\{-1,13\},
8016: };
8017: static XPoint seg1_3211[] = {
8018:
                                              \{0,7\},\{0,8\},\{1,8\},\{1,7\},\{0,7\},
8019: };
8020: static XPoint seg2 3211[] = {
8021:
                                              \{1,9\},\{2,10\},\
8022: };
8023: static XPoint seg3 3211[] = {
8024:
                                             \{2,8\},\{1,12\},\
8025: };
8026: static XPoint *char3211[] = {
8027:
                                              seg0 3211, seg1 3211, seg2 3211, seg3 3211,
8028:
                                             NULL,
8029: };
8030: static int char p3211[] = {
8031:
                                             XtNumber(seg0_3211), XtNumber(seg1_3211), XtNumber(seg2_3211),
8032:
                                             XtNumber(seg3 3211),
8033: };
8034: static XPoint seg0 3212[] = {
8035:
                                              \{0, -5\}, \{-1, -4\}, \{-1, -3\}, \{0, -2\}, \{1, -2\}, \{2, -3\}, \{0, -2\}, \{1, -2\}, \{2, -3\}, \{2, -3\}, \{2, -3\}, \{2, -3\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\},
8036:
                                              \{2, -4\}, \{1, -5\}, \{0, -5\}, 
8037: };
8038: static XPoint seg1 3212[] = {
8039:
                                              \{0, -4\}, \{0, -3\}, \{1, -3\}, \{1, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{0, -4\}, \{
8040: };
```

```
8041: static XPoint seg2 3212[] = {
8042:
                                                              \{0,6\},\{-1,7\},\{-1,8\},\{0,9\},\{1,9\},\{2,8\},\{2,7\},\{1,6\},
8043:
                                                              {0,6},
8044: };
8045: static XPoint seg3_3212[] = {
8046:
                                                             \{0,7\},\{0,8\},\{1,8\},\{1,7\},\{0,7\},
8047: };
8048: static XPoint *char3212[] = {
8049:
                                                              seg0 3212, seg1 3212, seg2 3212, seg3 3212,
8050:
                                                             NULL,
8051: };
8052: static int char p3212[] = {
8053:
                                                             XtNumber(seg0 3212), XtNumber(seg1 3212), XtNumber(seg2 3212),
8054:
                                                             XtNumber(seg3 3212),
8055: };
8056: static XPoint seg0 3213[] = {
8057:
                                                             \{0, -5\}, \{-1, -4\}, \{-1, -3\},\
8058:
                                                              \{0, -2\}, \{1, -2\}, \{2, -3\}, \{2, -4\}, \{1, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{0, -5\}, \{
8059: };
8060: static XPoint seg1 3213[] = {
8061:
                                                              \{0, -4\}, \{0, -3\}, \{1, -3\}, \{1, -4\}, \{0, -4\},
8062: };
8063: static XPoint seg2 3213[] = {
8064:
                                                              \{2,8\},\{1,9\},\{0,9\},\{-1,8\},\{-1,7\},\{0,6\},\{1,6\},\{2,7\},
8065:
                                                              \{2,10\},\{1,12\},\{-1,13\},\
8066: };
8067: static XPoint seg3 3213[] = {
8068:
                                                              \{0,7\},\{0,8\},\{1,8\},\{1,7\},\{0,7\},
8069: };
8070: static XPoint seg4 3213[] = {
8071:
                                                             \{1,9\},\{2,10\},\
8072: };
8073: static XPoint seg5 3213[] = {
8074:
                                                             \{2,8\},\{1,12\},\
8075: };
8076: static XPoint *char3213[] = {
8077:
                                                              seg0 3213, seg1 3213, seg2 3213, seg3 3213, seg4 3213, seg5 3213,
8078:
                                                             NULL.
8079: };
8080: static int char p3213[] = {
                                                             XtNumber(seg0 3213), XtNumber(seg1 3213), XtNumber(seg2 3213),
8081:
8082:
                                                             XtNumber(seg3 3213), XtNumber(seg4 3213), XtNumber(seg5 3213),
8083: };
8084: static XPoint seg0 3214[] = {
8085:
                                                              \{0, -12\}, \{-1, -11\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{0, -1\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}, \{-1, -9\}
8086: };
8087: static XPoint seg1_3214[] = {
8088:
                                                             \{0, -12\}, \{0, 2\}, \{1, 2\}, \{0, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1, 2\}, \{1
8089: };
8090: static XPoint seg2 3214[] = {
8091:
                                                              \{0, -12\}, \{1, -12\}, \{1, 2\}, 
8092: };
```

```
Listing C.4 Continued
```

```
8093: static XPoint seg3 3214[] = {
8094:
                                                                                                                   \{1, -12\}, \{2, -11\}, \{2, -9\}, \{1, -1\}, \{2, -9\}, \{1, -1\}, \{2, -9\}, \{1, -1\}, \{2, -9\}, \{1, -1\}, \{2, -9\}, \{1, -1\}, \{2, -9\}, \{1, -1\}, \{2, -9\}, \{2, -1\}, \{2, -9\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\}, \{3, -1\},
8095: };
8096: static XPoint seg4 3214[] = {
8097:
                                                                                                                     \{0,6\},\{-1,7\},\{-1,8\},\{0,9\},\{1,9\},\{2,8\},\{2,7\},\{1,6\},
8098:
                                                                                                                     \{0, 6\},\
8099: };
8100: static XPoint seg5 3214[] = {
8101:
                                                                                                                   \{0,7\},\{0,8\},\{1,8\},\{1,7\},\{0,7\},
8102: };
8103: static XPoint *char3214[] = {
8104:
                                                                                                                     seg0 3214, seg1 3214, seg2 3214, seg3 3214, seg4 3214, seg5 3214,
8105:
                                                                                                                   NULL,
8106: };
8107: static int char_p3214[] = {
8108:
                                                                                                                   XtNumber(seg0_3214), XtNumber(seg1_3214), XtNumber(seg2_3214),
8109:
                                                                                                                   XtNumber(seg3 3214), XtNumber(seg4 3214), XtNumber(seg5 3214),
8110: };
8111: static XPoint seg0 3215[] = {
8112:
                                                                                                                   \{-5, -7\}, \{-5, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4, -8\}, \{-4,
8113:
                                                                                                                     \{-4, -6\}, \{-6, -6\}, \{-6, -8\}, \{-5, -10\}, \{-4, -11\}, \{-2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{3, -12\}, \{3, -12\}, \{4, -11\}, \{-2, -12\}, \{4, -11\}, \{-2, -12\}, \{4, -11\}, \{-2, -12\}, \{4, -12\}, \{4, -11\}, \{-2, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12
8114:
                                                                                                                   \{5, -11\}, \{6, -10\}, \{7, -8\}, \{7, -6\}, \{6, -4\}, \{5, -3\}, \{1, -1\}, \{6, -10\}, \{6, -10\}, \{1, -1\}, \{1, -1\}, \{1, -1\}, \{1, -1\}, \{1, -1\}, \{1, -1\}, \{2, -1\}, \{2, -1\}, \{2, -1\}, \{3, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1\}, \{4, -1
8115: };
8116: static XPoint seg1 3215[] = {
8117:
                                                                                                                 \{5, -10\}, \{6, -9\}, \{6, -5\}, \{5, -4\}, \{6, -5\}, \{5, -4\}, \{6, -5\}, \{6, -5\}, \{5, -4\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, \{6, -5\}, 
8118: };
8119: static XPoint seg2 3215[] = {
8120:
                                                                                                                     \{2, -12\}, \{4, -11\}, \{5, -9\}, \{5, -5\}, \{4, -3\}, \{3, -2\}, \{3, -2\}, \{4, -3\}, \{3, -2\}, \{4, -3\}, \{3, -2\}, \{4, -3\}, \{3, -2\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\}, \{4, -3\},
8121: };
8122: static XPoint seg3 3215[] = {
8123:
                                                                                                                   \{0, -1\}, \{0, 2\}, \{1, 2\}, \{1, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0, -1\}, \{0,
8124: };
8125: static XPoint seg4_3215[] = {
8126:
                                                                                                                   \{0,6\},\{-1,7\},\{-1,8\},\{0,9\},\{1,9\},\{2,8\},\{2,7\},\{1,6\},
8127:
                                                                                                                     {0,6},
8128: };
8129: static XPoint seg5 3215[] = {
8130:
                                                                                                                   \{0,7\},\{0,8\},\{1,8\},\{1,7\},\{0,7\},
8131: };
8132: static XPoint *char3215[] = {
8133:
                                                                                                                     seg0 3215, seg1 3215, seg2 3215, seg3 3215, seg4 3215, seg5 3215,
8134:
                                                                                                                   NULL,
8135: };
8136: static int char p3215[] = {
8137:
                                                                                                                   XtNumber(seg0_3215),XtNumber(seg1_3215),XtNumber(seg2_3215),
8138:
                                                                                                                   XtNumber(seg3_3215),XtNumber(seg4_3215),XtNumber(seg5_3215),
8139: };
8140: static XPoint seg0 3216[] = {
8141:
                                                                                                                     \{2, -12\}, \{0, -11\}, \{-1, -9\},\
8142:
                                                                                                                     \{-1, -6\}, \{0, -5\}, \{1, -5\}, \{2, -6\}, \{2, -7\}, \{1, -8\}, \{0, -8\}, \{-1, -7\}, 
8143: };
```

```
8144: static XPoint seg1 3216[] = {
8145:
                                                                                                        \{0, -7\}, \{0, -6\}, \{1, -6\}, \{1, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{0, -7\}, \{
8146: };
8147: static XPoint seg2 3216[] = {
8148:
                                                                                                       \{0, -11\}, \{-1, -7\},\
8149: };
8150: static XPoint seg3 3216[] = {
8151:
                                                                                                       \{-1, -9\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, \{0, -8\}, 
8152: };
8153: static XPoint *char3216[] = {
8154:
                                                                                                        seg0 3216, seg1 3216, seg2 3216, seg3 3216,
8155:
                                                                                                       NULL,
8156: };
8157: static int char p3216[] = {
                                                                                                       XtNumber(seg0 3216),XtNumber(seg1_3216),XtNumber(seg2_3216),
8158:
8159:
                                                                                                       XtNumber(seg3 3216),
8160: };
8161: static XPoint seg0 3217[] = {
8162:
                                                                                                        \{2, -10\}, \{1, -9\}, \{0, -9\}, \{-1, -10\}, \{-1, -11\}, \{0, -12\}, \{-1, -11\}, \{0, -12\}, \{-1, -11\}, \{0, -12\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -11\}, \{-1, -
8163:
                                                                                                        \{1, -12\}, \{2, -11\}, \{2, -8\}, \{1, -6\}, \{-1, -5\},
8164: };
8165: static XPoint seq1 3217[] = {
8166:
                                                                                                        \{0, -11\}, \{0, -10\}, \{1, -10\}, \{1, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, -11\}, \{0, 
8167: };
8168: static XPoint seg2 3217[] = {
8169:
                                                                                                       \{1, -9\}, \{2, -8\},\
8170: };
8171: static XPoint seg3 3217[] = {
8172:
                                                                                                       \{2, -10\}, \{1, -6\},\
8173: };
8174: static XPoint *char3217[] = {
8175:
                                                                                                        seg0_3217, seg1_3217, seg2_3217, seg3_3217,
8176:
                                                                                                       NULL,
8177: };
8178: static int char_p3217[] = {
8179:
                                                                                                       XtNumber(seg0 3217), XtNumber(seg1 3217), XtNumber(seg2 3217),
8180:
                                                                                                       XtNumber(seg3 3217),
8181: };
8182: static XPoint seg0 3218[] = {
8183:
                                                                                                        \{9, -3\}, \{9, -4\}, \{8, -4\}, \{8, -2\}, \{10, -2\}, \{10, -4\},
8184:
                                                                                                        \{9, -5\}, \{8, -5\}, \{7, -4\}, \{6, -2\}, \{4, 3\}, \{2, 6\}, \{0, 8\}, \{-2, 9\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, -2\}, \{0, 
8185:
                                                                                                        \{-6,9\},\{-8,8\},\{-9,6\},\{-9,3\},\{-8,1\},\{-2,-3\},\{0,-5\},\{1,-7\},
8186:
                                                                                                       \{1, -9\}, \{0, -11\}, \{-2, -12\}, \{-4, -11\}, \{-5, -9\}, \{-5, -6\}, \{-4, -3\}, \{-1, -2\}, \{-2, -12\}, \{-2, -12\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{
                                                                                                        \{-2,0\},\{2,5\},\{5,8\},\{7,9\},\{9,9\},\{10,7\},\{10,6\},
8187:
8188: };
8189: static XPoint seg1 3218[] = {
8190:
                                                                                                        \{-7,8\},\{-8,6\},\{-8,3\},\{-7,1\},\{-6,0\},
8191: };
8192: static XPoint seg2_3218[] = {
8193:
                                                                                                       \{0, -5\}, \{1, -9\},\
8194: };
8195: static XPoint seq3 3218[] = {
8196:
                                                                                                       \{1, -7\}, \{0, -11\},\
8197: };
```

```
8198: static XPoint seg4 3218[] = {
8199:
                                                                      \{-4, -11\}, \{-5, -7\},
8200: };
8201: static XPoint seg5 3218[] = {
8202:
                                                                        \{-4, -4\}, \{-2, -1\}, \{2, 4\}, \{5, 7\}, \{7, 8\},
8203: };
8204: static XPoint seg6 3218[] = {
8205:
                                                                         \{-4,9\}, \{-6,8\}, \{-7,6\}, \{-7,3\}, \{-6,1\}, \{-2,-3\}, \{-6,1\}, \{-2,-3\}, \{-2,-3\}, \{-2,-3\}, \{-2,-3\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}, \{-3,-2\}
8206: };
8207: static XPoint seg7 3218[] = {
8208:
                                                                         \{-5, -9\}, \{-4, -5\}, \{-1, -1\}, \{3, 4\}, \{6, 7\}, \{8, 8\}, \{9, 8\}, \{10, 7\},
8209: };
8210: static XPoint *char3218[] = {
8211:
                                                                        seg0_3218, seg1_3218, seg2_3218, seg3_3218, seg4_3218, seg5_3218,
8212:
                                                                         seg6 3218, seg7 3218,
8213:
                                                                        NULL.
8214: };
8215: static int char p3218[] = {
8216:
                                                                        XtNumber(seg0 3218), XtNumber(seg1 3218), XtNumber(seg2 3218),
8217:
                                                                        XtNumber(seg3 3218), XtNumber(seg4 3218), XtNumber(seg5 3218),
                                                                        XtNumber(seq6 3218), XtNumber(seq7 3218),
8218:
8219: };
8220: static XPoint seg0_3219[] = {
8221:
                                                                         \{-2, -16\}, \{-2, 13\}, 
8222: };
8223: static XPoint seg1 3219[] = {
8224:
                                                                        \{2, -16\}, \{2, 13\},\
8225: };
8226: static XPoint seg2 3219[] = {
8227:
                                                                         \{6, -7\}, \{6, -8\}, \{5, -8\}, \{5, -6\}, \{7, -6\}, \{7, -8\}, \{6, -10\}, \{5, -11\},
8228:
                                                                         \{2, -12\}, \{-2, -12\}, \{-5, -11\}, \{-7, -9\}, \{-7, -6\}, \{-6, -4\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-
8229:
                                                                        \{3,0\},\{5,1\},\{6,3\},\{6,6\},\{5,8\},
8230: };
8231: static XPoint seg3 3219[] = {
8232:
                                                                        \{-6, -6\}, \{-5, -4\}, \{-3, -3\}, \{3, -1\}, \{5, 0\}, \{6, 2\}, \{6, 2\}, \{-6, -6\}, \{-5, -4\}, \{-3, -3\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{-1\}, \{
8233: };
8234: static XPoint seg4 3219[] = {
8235:
                                                                        \{-5, -11\}, \{-6, -9\}, \{-6, -7\}, \{-5, -5\}, \{-3, -4\}, \{3, -2\}, \{6, 0\}, \{7, 2\}, \{6, 0\}, \{7, 2\}, \{-3, -4\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-3, -2\}, \{-
8236:
                                                                         \{7,5\},\{6,7\},\{5,8\},\{2,9\},\{-2,9\},\{-5,8\},\{-6,7\},\{-7,5\},\{-7,3\},
8237:
                                                                         \{-5,3\},\{-5,5\},\{-6,5\},\{-6,4\},
8238: };
8239: static XPoint *char3219[] = {
8240:
                                                                         seg0_3219, seg1_3219, seg2_3219, seg3_3219, seg4_3219,
8241:
                                                                        NULL,
8242: };
8243: static int char_p3219[] = {
8244:
                                                                         XtNumber(seg0 3219),XtNumber(seg1 3219),XtNumber(seg2 3219),
8245:
                                                                        XtNumber(seg3 3219), XtNumber(seg4 3219),
8246: };
```

```
8247: static XPoint seg0 3220[] = {
8248:
                                                              \{9, -16\}, \{-9, 16\}, \{-8, 16\}, \{-8, 16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-
8249: };
8250: static XPoint seg1 3220[] = {
8251:
                                                             \{9, -16\}, \{10, -16\}, \{-8, 16\}, \{-8, 16\}, \{-8, 16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}, \{-16\}
8252: };
8253: static XPoint *char3220[] = {
8254:
                                                             seg0_3220, seg1_3220,
8255:
                                                             NULL,
8256: };
8257: static int char p3220[] = {
                                                             XtNumber(seg0 3220),XtNumber(seg1 3220),
8258:
8259: };
8260: static XPoint seg0 3221[] = {
8261:
                                                              \{3, -16\}, \{1, -14\}, \{-1, -11\}, \{-3, -7\}, \{-4, -2\},
8262:
                                                              \{-4,2\},\{-3,7\},\{-1,11\},\{1,14\},\{3,16\},
8263: };
8264: static XPoint seg1 3221[] = {
8265:
                                                              \{-1, -10\}, \{-2, -7\}, \{-3, -3\}, \{-3, 3\}, \{-2, 7\}, \{-1, 10\},
8266: };
8267: static XPoint seg2 3221[] = {
8268:
                                                             \{1, -14\}, \{0, -12\}, \{-1, -9\}, \{-2, -3\}, \{-2, 3\}, \{-1, 9\}, \{0, 12\}, \{1, 14\},
8269: };
8270: static XPoint *char3221[] = {
8271:
                                                             seg0 3221, seg1 3221, seg2 3221,
8272:
                                                             NULL,
8273: };
8274: static int char p3221[] = {
8275:
                                                             XtNumber(seg0 3221), XtNumber(seg1 3221), XtNumber(seg2 3221),
8276: };
8277: static XPoint seg0 3222[] = {
8278:
                                                             \{-3, -16\}, \{-1, -14\}, \{1, -11\}, \{3, -7\}, \{4, -2\}, \{4, 2\}, \{3, 7\}, \{1, 11\}, \{3, -7\}, \{4, -2\}, \{4, 2\}, \{3, 7\}, \{1, 11\}, \{1, -11\}, \{3, -7\}, \{4, -2\}, \{4, 2\}, \{3, 7\}, \{1, 11\}, \{1, -11\}, \{3, -7\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, 
8279:
                                                              \{-1, 14\}, \{-3, 16\},
8280: };
8281: static XPoint seg1_3222[] = {
8282:
                                                             \{1, -10\}, \{2, -7\}, \{3, -3\}, \{3, 3\}, \{2, 7\}, \{1, 10\},\
8283: };
8284: static XPoint seg2 3222[] = {
8285:
                                                              \{-1, -14\}, \{0, -12\}, \{1, -9\}, \{2, -3\}, \{2, 3\}, \{1, 9\}, \{0, 12\}, \{-1, 14\},
8286: };
8287: static XPoint *char3222[] = {
8288:
                                                              seg0 3222, seg1 3222, seg2 3222,
8289:
                                                             NULL,
8290: };
8291: static int char p3222[] = {
8292:
                                                             XtNumber(seg0 3222), XtNumber(seg1 3222), XtNumber(seg2 3222),
8293: };
8294: static XPoint seg0 3223[] = {
8295:
                                                              \{0, -12\}, \{-1, -11\}, \{1, -1\}, \{0, 0\},\
8296: };
8297: static XPoint seg1_3223[] = {
8298:
                                                              \{0, -12\}, \{0, 0\},\
8299: };
```

```
Listing C.4 Continued
```

```
8300: static XPoint seg2_3223[] = {
8301:
                                              \{0, -12\}, \{1, -11\}, \{-1, -1\}, \{0, 0\},\
8302: };
8303: static XPoint seg3 3223[] = {
8304:
                                              \{-5, -9\}, \{-4, -9\}, \{4, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\},
8305: };
8306: static XPoint seg4 3223[] = {
8307:
                                              \{-5, -9\}, \{5, -3\},\
8308: };
8309: static XPoint seg5 3223[] = {
8310:
                                             \{-5, -9\}, \{-5, -8\}, \{5, -4\}, \{5, -3\},
8311: };
8312: static XPoint seg6 3223[] = {
8313:
                                              \{5, -9\}, \{4, -9\}, \{-4, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -3\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -5\}, \{-5, -
8314: };
8315: static XPoint seg7_3223[] = {
8316:
                                              \{5, -9\}, \{-5, -3\},\
8317: };
8318: static XPoint seg8 3223[] = {
8319:
                                              \{5, -9\}, \{5, -8\}, \{-5, -4\}, \{-5, -3\},\
8320: };
8321: static XPoint *char3223[] = {
8322:
                                               seg0 3223, seg1 3223, seg2 3223, seg3 3223, seg4 3223, seg5 3223,
8323:
                                               seg6_3223, seg7_3223, seg8_3223,
8324:
                                              NULL,
8325: };
8326: static int char p3223[] = {
8327:
                                              XtNumber(seg0_3223),XtNumber(seg1_3223),XtNumber(seg2_3223),
8328:
                                               XtNumber(seg3_3223),XtNumber(seg4_3223),XtNumber(seg5_3223),
8329:
                                              XtNumber(seg6_3223),XtNumber(seg7_3223),XtNumber(seg8_3223),
8330: };
8331: static XPoint seg0 3224[] = {
8332:
                                              \{-8, -1\}, \{9, -1\}, \{9, 0\},
8333: };
8334: static XPoint seg1 3224[] = {
8335:
                                              \{-8, -1\}, \{-8, 0\}, \{9, 0\},\
8336: };
8337: static XPoint *char3224[] = {
8338:
                                               seg0 3224, seg1 3224,
8339:
                                              NULL,
8340: };
8341: static int char p3224[] = {
8342:
                                              XtNumber(seg0_3224),XtNumber(seg1_3224),
8343: };
8344: static XPoint seg0 3225[] = {
8345:
                                              \{0, -9\}, \{0, 8\}, \{1, 8\}, 
8346: };
8347: static XPoint seg1 3225[] = {
8348:
                                               \{0, -9\}, \{1, -9\}, \{1, 8\},\
8349: };
```

```
8350: static XPoint seg2 3225[] = {
8351:
                                                 \{-8, -1\}, \{9, -1\}, \{9, 0\},
8352: };
8353: static XPoint seg3 3225[] = {
8354:
                                                \{-8, -1\}, \{-8, 0\}, \{9, 0\}, \{9, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-
8355: };
8356: static XPoint *char3225[] = {
8357:
                                                 seg0_3225, seg1_3225, seg2_3225, seg3_3225,
8358:
                                                NULL,
8359: };
8360: static int char p3225[] = {
                                                XtNumber(seg0 3225), XtNumber(seg1 3225), XtNumber(seg2 3225),
8361:
8362:
                                                XtNumber(seg3 3225),
8363: };
8364: static XPoint seg0 3226[] = {
8365:
                                                 \{-8, -5\}, \{9, -5\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, \{9, -4\}, 
8366: };
8367: static XPoint seg1 3226[] = {
8368:
                                                 \{-8, -5\}, \{-8, -4\}, \{9, -4\},
8369: };
8370: static XPoint seg2 3226[] = {
8371:
                                                \{-8,3\},\{9,3\},\{9,4\},
8372: };
8373: static XPoint seg3 3226[] = {
8374:
                                                \{-8,3\},\{-8,4\},\{9,4\},
8375: };
8376: static XPoint *char3226[] = {
8377:
                                                 seg0 3226, seg1 3226, seg2 3226, seg3 3226,
8378:
                                                NULL,
8379: };
8380: static int char p3226[] = {
8381:
                                                XtNumber(seg0_3226), XtNumber(seg1_3226), XtNumber(seg2_3226),
8382:
                                                XtNumber(seg3 3226),
8383: };
8384: static XPoint seg0_3228[] = {
8385:
                                                 \{-4, -12\}, \{-5, -11\}, \{-5, -5\},
8386: };
8387: static XPoint seg1 3228[] = {
8388:
                                                 \{-4, -11\}, \{-5, -5\},
8389: };
8390: static XPoint seg2 3228[] = {
8391:
                                                \{-4, -12\}, \{-3, -11\}, \{-5, -5\}, 
8392: };
8393: static XPoint seg3 3228[] = {
8394:
                                                 \{5, -12\}, \{4, -11\}, \{4, -5\},\
8395: };
8396: static XPoint seg4_3228[] = {
8397:
                                                \{5, -11\}, \{4, -5\},\
8398: };
8399: static XPoint seg5 3228[] = {
8400:
                                                 \{5, -12\}, \{6, -11\}, \{4, -5\},\
8401: };
```

```
C
```

```
8402: static XPoint *char3228[] = {
8403:
                                                                                       seg0_3228, seg1_3228, seg2_3228, seg3_3228, seg4_3228, seg5_3228,
8404:
                                                                                      NULL,
8405: };
8406: static int char p3228[] = {
8407:
                                                                                      XtNumber(seg0 3228), XtNumber(seg1 3228), XtNumber(seg2 3228),
8408:
                                                                                      XtNumber(seg3 3228), XtNumber(seg4 3228), XtNumber(seg5 3228),
8409: };
8410: static XPoint seg0 3229[] = {
8411:
                                                                                       \{-1, -12\}, \{-3, -11\}, \{-4, -9\}, \{-4, -7\}, \{-3, -5\}, \{-1, -4\}, \{1, -4\}, \{1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1, -4\}, \{-1,
8412:
                                                                                       \{3, -5\}, \{4, -7\}, \{4, -9\}, \{3, -11\}, \{1, -12\}, \{-1, -12\}, 
8413: };
8414: static XPoint seg1 3229[] = {
8415:
                                                                                       \{-1, -12\}, \{-4, -9\}, \{-3, -5\}, \{1, -4\}, \{4, -7\}, \{3, -11\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, 
8416: };
8417: static XPoint seg2 3229[] = {
8418:
                                                                                       \{1, -12\}, \{-3, -11\}, \{-4, -7\}, \{-1, -4\}, \{3, -5\}, \{4, -9\}, \{1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -
8419: };
8420: static XPoint *char3229[] = {
                                                                                       seg0 3229, seg1 3229, seg2 3229,
8421:
8422:
                                                                                      NULL,
8423: };
8424: static int char_p3229[] = {
8425:
                                                                                      XtNumber(seg0 3229), XtNumber(seg1 3229), XtNumber(seg2 3229),
8426: };
8427: static XPoint seg0 3249[] = {
8428:
                                                                                       {-8,8},
8429: };
8430: static XPoint *char3249[] = {
8431:
                                                                                      seg0 3249,
8432:
                                                                                      NULL.
8433: };
8434: static int char p3249[] = {
8435:
                                                                                      XtNumber(seg0 3249),
8436: };
8437: static XPoint seg0 3250[] = {
8438:
                                                                                       \{2, -12\}, \{-1, -11\}, \{-3, -9\}, \{-5, -6\}, \{-6, -3\}, \{-7, 1\}, \{-7, 4\},
8439:
                                                                                       \{-6,7\},\{-5,8\},\{-3,9\},\{-1,9\},\{2,8\},\{4,6\},\{6,3\},\{7,0\},\{8,-4\},
8440:
                                                                                       \{8, -7\}, \{7, -10\}, \{6, -11\}, \{4, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{2, -12\}, \{3, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -
8441: };
8442: static XPoint seg1 3250[] = {
8443:
                                                                                       \{-1, -10\}, \{-3, -8\}, \{-4, -6\}, \{-5, -3\}, \{-6, 1\}, \{-6, 5\}, \{-5, 7\},
8444: };
8445: static XPoint seg2 3250[] = {
8446:
                                                                                       \{2,7\},\{4,5\},\{5,3\},\{6,0\},\{7,-4\},\{7,-8\},\{6,-10\},
8447: };
8448: static XPoint seg3 3250[] = {
8449.
                                                                                       \{2, -12\}, \{0, -11\}, \{-2, -8\}, \{-3, -6\}, \{-4, -3\}, \{-5, 1\}, \{-5, 6\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -1\}, \{-5, -
8450:
                                                                                       \{-4,8\},\{-3,9\},
8451: };
```

```
8452: static XPoint seg4 3250[] = {
8453:
                                                           \{-1,9\},\{1,8\},\{3,5\},\{4,3\},\{5,0\},\{6,-4\},\{6,-9\},\{5,-11\},
8454:
                                                           \{4, -12\},\
8455: };
8456: static XPoint *char3250[] = {
                                                           seg0 3250, seg1 3250, seg2 3250, seg3 3250, seg4 3250,
8457:
8458:
                                                           NULL,
8459: };
8460: static int char p3250[] = {
8461:
                                                           XtNumber(seg0 3250), XtNumber(seg1 3250), XtNumber(seg2 3250),
8462:
                                                           XtNumber(seg3 3250), XtNumber(seg4 3250),
8463: };
8464: static XPoint seg0 3251[] = {
8465:
                                                           \{2, -8\}, \{-3, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{-1, 9\}, \{
8466: };
8467: static XPoint seg1 3251[] = {
8468:
                                                          \{5, -12\}, \{3, -8\}, \{-2, 9\},\
8469: };
8470: static XPoint seg2 3251[] = {
8471:
                                                          \{5, -12\}, \{-1, 9\},\
8472: };
8473: static XPoint seg3 3251[] = {
8474:
                                                           \{5, -12\}, \{2, -9\}, \{-1, -7\}, \{-3, -6\},\
8475: };
8476: static XPoint seq4 3251[] = {
8477:
                                                           \{2, -8\}, \{0, -7\}, \{-3, -6\},\
8478: };
8479: static XPoint *char3251[] = {
8480:
                                                           seg0 3251, seg1 3251, seg2 3251, seg3 3251, seg4 3251,
8481:
                                                           NULL,
8482: };
8483: static int char_p3251[] = {
8484:
                                                           XtNumber(seg0 3251), XtNumber(seg1 3251), XtNumber(seg2 3251),
8485:
                                                           XtNumber(seg3 3251), XtNumber(seg4 3251),
8486: };
8487: static XPoint seg0 3252[] = {
8488:
                                                           \{-3, -7\}, \{-3, -8\}, \{-2, -8\}, \{-2, -6\}, \{-4, -6\}, \{-4, -8\}, \{-3, -10\}, 
8489:
                                                           \{-2, -11\}, \{1, -12\}, \{4, -12\}, \{7, -11\}, \{8, -9\}, \{8, -7\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -
8490:
                                                           \{5, -3\}, \{-5, 3\}, \{-7, 5\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{-9, 9\}, \{
8491: };
8492: static XPoint seg1_3252[] = {
8493:
                                                           \{6, -11\}, \{7, -9\}, \{7, -7\}, \{6, -5\}, \{4, -3\}, \{1, -1\},
8494: };
8495: static XPoint seg2 3252[] = {
8496:
                                                           \{4, -12\}, \{5, -11\}, \{6, -9\}, \{6, -7\}, \{5, -5\}, \{3, -3\}, \{-5, 3\},
8497: };
8498: static XPoint seg3 3252[] = {
8499:
                                                           \{-8,7\},\{-7,6\},\{-5,6\},\{0,7\},\{5,7\},\{6,6\},
8500: };
8501: static XPoint seg4 3252[] = {
8502:
                                                           \{-5,6\},\{0,8\},\{5,8\},
8503: };
```

```
8504: static XPoint seg5_3252[] = {
8505:
                                                        \{-5,6\},\{0,9\},\{3,9\},\{5,8\},\{6,6\},\{6,5\},
8506: };
8507: static XPoint *char3252[] = {
8508:
                                                         seg0 3252, seg1 3252, seg2 3252, seg3 3252, seg4 3252, seg5 3252,
8509:
                                                        NULL,
8510: };
8511: static int char p3252[] = {
8512:
                                                        XtNumber(seg0_3252),XtNumber(seg1_3252),XtNumber(seg2_3252),
8513:
                                                        XtNumber(seg3 3252), XtNumber(seg4 3252), XtNumber(seg5 3252),
8514: };
8515: static XPoint seg0 3253[] = {
8516:
                                                         \{-3, -7\}, \{-3, -8\}, \{-2, -8\}, \{-2, -6\}, \{-4, -6\}, \{-4, -8\}, \{-3, -10\}, 
8517:
                                                         \{-2, -11\}, \{1, -12\}, \{4, -12\}, \{7, -11\}, \{8, -9\}, \{8, -7\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -5\}, \{7, -
8518:
                                                         \{6, -4\}, \{4, -3\}, \{1, -2\},\
8519: };
8520: static XPoint seg1_3253[] = {
8521:
                                                        \{6, -11\}, \{7, -9\}, \{7, -7\}, \{6, -5\}, \{5, -4\},
8522: };
8523: static XPoint seg2_3253[] = {
8524:
                                                        \{4, -12\}, \{5, -11\}, \{6, -9\}, \{6, -7\}, \{5, -5\}, \{3, -3\}, \{1, -2\}, \{5, -11\}, \{6, -9\}, \{6, -7\}, \{5, -5\}, \{3, -3\}, \{1, -2\}, \{1, -2\}, \{2, -12\}, \{3, -3\}, \{1, -2\}, \{3, -3\}, \{1, -2\}, \{3, -3\}, \{3, -3\}, \{1, -2\}, \{3, -3\}, \{3, -3\}, \{3, -3\}, \{3, -3\}, \{3, -3\}, \{3, -3\}, \{3, -3\}, \{3, -3\}, \{3, -3\}, \{3, -3\}, \{3, -3\}, \{4, -2\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3\}, \{5, -3
8525: };
8526: static XPoint seg3 3253[] = {
8527:
                                                         \{-1, -2\}, \{1, -2\}, \{4, -1\}, \{5, 0\}, \{6, 2\}, \{6, 5\}, \{5, 7\}, \{3, 8\},
8528:
                                                         \{0,9\},\{-3,9\},\{-6,8\},\{-7,7\},\{-8,5\},\{-8,3\},\{-6,3\},\{-6,5\},
8529:
                                                         \{-7,5\},\{-7,4\},
8530: };
8531: static XPoint seg4 3253[] = {
8532:
                                                        \{4,0\},\{5,2\},\{5,5\},\{4,7\},
8533: };
8534: static XPoint seg5 3253[] = {
8535:
                                                        \{1, -2\}, \{3, -1\}, \{4, 1\}, \{4, 5\}, \{3, 7\}, \{2, 8\}, \{0, 9\},\
8536: };
8537: static XPoint *char3253[] = {
8538:
                                                         seg0 3253,seg1 3253,seg2 3253,seg3 3253,seg4 3253,seg5 3253,
8539:
                                                        NULL,
8540: };
8541: static int char p3253[] = {
                                                        XtNumber(seg0 3253),XtNumber(seg1_3253),XtNumber(seg2_3253),
8542:
8543:
                                                        XtNumber(seg3 3253), XtNumber(seg4 3253), XtNumber(seg5 3253),
8544: };
8545: static XPoint seg0 3254[] = {
8546:
                                                         \{5, -8\},\
8547:
                                                         \{0,9\},\{2,9\},\
8548: };
8549: static XPoint seg1 3254[] = {
8550:
                                                         \{8, -12\}, \{6, -8\}, \{1, 9\}, \{6, -8\}, \{1, 9\}, \{1, 9\}, \{2, -8\}, \{2, -8\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{3, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, 
8551: };
8552: static XPoint seg2 3254[] = {
8553:
                                                         \{8, -12\}, \{2, 9\},\
8554: };
```

```
8555: static XPoint seg3 3254[] = {
8556:
                                                                                       \{8, -12\}, \{-8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{8, 3\}, \{
8557: };
8558: static XPoint *char3254[] = {
8559:
                                                                                       seg0_3254, seg1_3254, seg2_3254, seg3_3254,
8560:
                                                                                       NULL,
8561: };
8562: static int char p3254[] = {
                                                                                       XtNumber(seg0 3254), XtNumber(seg1 3254), XtNumber(seg2 3254),
8563:
8564:
                                                                                     XtNumber(seg3 3254),
8565: };
8566: static XPoint seg0 3255[] = {
8567:
                                                                                     \{-1, -12\}, \{-6, -2\}, 
8568: };
8569: static XPoint seg1 3255[] = {
8570:
                                                                                       \{-1, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9,
8571: };
8572: static XPoint seg2 3255[] = {
8573:
                                                                                       \{-1, -11\}, \{7, -11\}, 
8574: };
8575: static XPoint seg3 3255[] = {
8576:
                                                                                     \{-2, -10\}, \{3, -10\}, \{7, -11\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9, -12\}, \{9,
8577: };
8578: static XPoint seg4 3255[] = {
8579:
                                                                                     \{-6, -2\}, \{-5, -3\}, \{-2, -4\}, \{1, -4\}, \{4, -3\}, \{5, -2\}, \{6, 0\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6, 3\}, \{6,
8580:
                                                                                       \{5,6\},\{3,8\},\{-1,9\},\{-4,9\},\{-6,8\},\{-7,7\},\{-8,5\},\{-8,3\},
8581:
                                                                                       \{-6,3\},\{-6,5\},\{-7,5\},\{-7,4\},
8582: };
8583: static XPoint seg5 3255[] = {
8584:
                                                                                       \{4, -2\}, \{5, 0\}, \{5, 3\}, \{4, 6\}, \{2, 8\}, \{2, 8\}, \{4, 6\}, \{2, 8\}, \{2, 8\}, \{2, 8\}, \{2, 8\}, \{3, 8\}, \{4, 6\}, \{2, 8\}, \{3, 8\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4, 6\}, \{4,
8585: };
8586: static XPoint seg6 3255[] = {
8587:
                                                                                       \{1, -4\}, \{3, -3\}, \{4, -1\}, \{4, 3\}, \{3, 6\}, \{1, 8\}, \{-1, 9\},
8588: };
8589: static XPoint *char3255[] = {
                                                                                       seg0 3255, seg1 3255, seg2 3255, seg3 3255, seg4 3255, seg5 3255,
8590:
8591:
                                                                                       seg6 3255,
8592:
                                                                                     NULL,
8593: };
8594: static int char p3255[] = {
8595:
                                                                                     XtNumber(seg0 3255),XtNumber(seg1 3255),XtNumber(seg2 3255),
8596:
                                                                                       XtNumber(seg3 3255),XtNumber(seg4 3255),XtNumber(seg5 3255),
8597:
                                                                                     XtNumber(seg6 3255),
8598: };
8599: static XPoint seg0 3256[] = {
8600:
                                                                                       \{7, -8\}, \{7, -9\}, \{6, -9\}, \{6, -7\}, \{8, -7\}, \{8, -9\}, \{7, -11\}, \{5, -12\},
8601:
                                                                                       \{2, -12\}, \{-1, -11\}, \{-3, -9\}, \{-5, -6\}, \{-6, -3\}, \{-7, 1\}, \{-7, 4\},
8602:
                                                                                       \{-6,7\},\{-5,8\},\{-3,9\},\{0,9\},\{3,8\},\{5,6\},\{6,4\},\{6,1\},\{5,-1\},
8603:
                                                                                       \{4, -2\}, \{2, -3\}, \{-1, -3\}, \{-3, -2\}, \{-4, -1\}, \{-5, 1\},
8604: };
8605: static XPoint seg1 3256[] = {
8606:
                                                                                       \{-2, -9\}, \{-4, -6\}, \{-5, -3\}, \{-6, 1\}, \{-6, 5\}, \{-5, 7\},
8607: };
```

```
Listing C.4 Continued
```

```
8608: static XPoint seg2 3256[] = {
8609:
                                                                                                   \{4,6\},\{5,4\},\{5,1\},\{4,-1\},
8610: };
8611: static XPoint seg3 3256[] = {
8612:
                                                                                                    \{2, -12\}, \{0, -11\}, \{-2, -8\}, \{-3, -6\}, \{-4, -3\}, \{-5, 1\}, \{-5, 6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -6\}, \{-5, -
8613:
                                                                                                    \{-4,8\},\{-3,9\},
8614: };
8615: static XPoint seg4 3256[] = {
8616:
                                                                                                    \{0,9\},\{2,8\},\{3,7\},\{4,4\},\{4,0\},\{3,-2\},\{2,-3\},
8617: };
8618: static XPoint *char3256[] = {
8619:
                                                                                                    seg0 3256, seg1 3256, seg2 3256, seg3 3256, seg4 3256,
8620:
                                                                                                   NULL,
8621: };
8622: static int char p3256[] = {
8623:
                                                                                                   XtNumber(seg0 3256), XtNumber(seg1 3256), XtNumber(seg2 3256),
8624:
                                                                                                   XtNumber(seg3 3256), XtNumber(seg4 3256),
8625: };
8626: static XPoint seg0_3257[] = {
8627:
                                                                                                    \{-4, -12\}, \{-6, -6\}, 
8628: };
8629: static XPoint seg1_3257[] = {
8630:
                                                                                                   \{9, -12\}, \{8, -9\}, \{6, -6\}, \{2, -1\}, \{0, 2\}, \{-1, 5\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{
8631: };
8632: static XPoint seg2 3257[] = {
8633:
                                                                                                    \{0,1\},\{-2,5\},\{-3,9\},
8634: };
8635: static XPoint seg3 3257[] = {
8636:
                                                                                                   \{6, -6\}, \{0, 0\}, \{-2, 3\}, \{-3, 5\}, \{-4, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-2, 9\}, \{-
8637: };
8638: static XPoint seg4 3257[] = {
8639:
                                                                                                   \{-5, -9\}, \{-2, -12\}, \{0, -12\}, \{5, -9\}, \{-5, -9\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2
8640: };
8641: static XPoint seg5_3257[] = {
8642:
                                                                                                    \{-3, -11\}, \{0, -11\}, \{5, -9\},\
8643: };
8644: static XPoint seg6 3257[] = {
8645:
                                                                                                    \{-5, -9\}, \{-3, -10\}, \{0, -10\}, \{5, -9\}, \{7, -9\}, \{8, -10\}, \{9, -12\}, \{9, -12\}, \{1, -10\}, \{1, -10\}, \{2, -10\}, \{2, -10\}, \{2, -10\}, \{3, -10\}, \{3, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -10\}, \{4, -
8646: };
8647: static XPoint *char3257[] = {
8648:
                                                                                                   seg0_3257, seg1_3257, seg2_3257, seg3_3257, seg4_3257, seg5_3257,
8649:
                                                                                                    seg6 3257,
8650:
                                                                                                   NULL,
8651: };
8652: static int char p3257[] = {
8653:
                                                                                                   XtNumber(seg0 3257), XtNumber(seg1 3257), XtNumber(seg2 3257),
8654:
                                                                                                    XtNumber(seg3 3257),XtNumber(seg4 3257),XtNumber(seg5 3257),
8655:
                                                                                                   XtNumber(seg6 3257),
8656: };
8657: static XPoint seg0 3258[] = {
8658:
                                                                                                    \{1, -12\}, \{-2, -11\}, \{-3, -10\}, \{-4, -8\}, \{-4, -5\}, \{-3, -3\}, \{-1, -2\},
8659:
                                                                                                    \{2, -2\}, \{5, -3\}, \{7, -4\}, \{8, -6\}, \{8, -9\}, \{7, -11\}, \{5, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{2, -2\}, \{2, -2\}, \{3, -3\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, -2\}, \{4, 
8660: };
```

```
8661: static XPoint seg1 3258[] = {
8662:
                             \{3, -12\}, \{-2, -11\}, \}
8663: };
8664: static XPoint seg2 3258[] = {
8665:
                            \{-2, -10\}, \{-3, -8\}, \{-3, -4\}, \{-2, -3\},
8666: };
8667: static XPoint seg3 3258[] = {
8668:
                            \{-3, -3\}, \{0, -2\},\
8669: };
8670: static XPoint seq4 3258[] = {
8671:
                            \{1, -2\}, \{5, -3\},\
8672: };
8673: static XPoint seg5 3258[] = {
8674:
                            \{6, -4\}, \{7, -6\}, \{7, -9\}, \{6, -11\},\
8675: };
8676: static XPoint seg6_3258[] = {
8677:
                            \{7, -11\}, \{3, -12\},\
8678: };
8679: static XPoint seg7 3258[] = {
8680:
                            \{1, -12\}, \{-1, -10\}, \{-2, -8\}, \{-2, -4\}, \{-1, -2\},
8681: };
8682: static XPoint seg8 3258[] = {
8683:
                             \{2, -2\}, \{4, -3\}, \{5, -4\}, \{6, -6\}, \{6, -10\}, \{5, -12\},
8684: };
8685: static XPoint seg9 3258[] = {
8686:
                             \{-1, -2\}, \{-5, -1\}, \{-7, 1\}, \{-8, 3\}, \{-8, 6\}, \{-7, 8\}, \{-4, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0, 9\}, \{0,
8687:
                             \{4,8\},\{5,7\},\{6,5\},\{6,2\},\{5,0\},\{4,-1\},\{2,-2\},
8688: };
8689: static XPoint seg10 3258[] = {
8690:
                             \{0, -2\}, \{-5, -1\},\
8691: };
8692: static XPoint seg11 3258[] = {
8693:
                             \{-4, -1\}, \{-6, 1\}, \{-7, 3\}, \{-7, 6\}, \{-6, 8\},
8694: };
8695: static XPoint seg12 3258[] = {
8696:
                            \{-7,8\},\{-2,9\},\{4,8\},
8697: };
8698: static XPoint seq13 3258[] = {
8699:
                            \{4,7\},\{5,5\},\{5,2\},\{4,0\},
8700: };
8701: static XPoint seg14 3258[] = {
8702:
                            \{4, -1\}, \{1, -2\},\
8703: };
8704: static XPoint seg15 3258[] = {
8705:
                             \{-1, -2\}, \{-3, -1\}, \{-5, 1\}, \{-6, 3\}, \{-6, 6\}, \{-5, 8\}, \{-4, 9\},
8706: };
8707: static XPoint seg16 3258[] = {
8708:
                            \{0,9\},\{2,8\},\{3,7\},\{4,5\},\{4,1\},\{3,-1\},\{2,-2\},
8709: };
8710: static XPoint *char3258[] = {
8711:
                             seg0 3258, seg1 3258, seg2 3258, seg3 3258, seg4 3258,
8712:
                             seg5 3258, seg6 3258, seg7 3258, seg8 3258, seg9 3258,
```

```
Listing C.4 Continued
```

```
8713:
                         seg10 3258, seg11 3258, seg12 3258, seg13 3258,
8714:
                         seg14 3258, seg15 3258, seg16 3258,
8715:
                        NULL,
8716: };
8717: static int char p3258[] = {
8718:
                        XtNumber(seq0 3258), XtNumber(seq1 3258), XtNumber(seq2 3258),
8719:
                        XtNumber(seg3 3258), XtNumber(seg4 3258), XtNumber(seg5 3258),
8720:
                        XtNumber(seg6_3258), XtNumber(seg7_3258), XtNumber(seg8_3258),
8721:
                        XtNumber(seg9 3258), XtNumber(seg10 3258), XtNumber(seg11 3258),
8722:
                         XtNumber(seg12 3258), XtNumber(seg13 3258), XtNumber(seg14 3258),
8723:
                        XtNumber(seg15_3258),XtNumber(seg16_3258),
8724: };
8725: static XPoint seg0 3259[] = {
8726:
                         \{6, -4\}, \{5, -2\}, \{4, -1\}, \{2, 0\}, \{-1, 0\}, \{-3, -1\}, \{-4, -2\}, \{-5, -4\},
8727:
                         \{-5, -7\}, \{-4, -9\}, \{-2, -11\}, \{1, -12\}, \{4, -12\}, \{6, -11\}, \{7, -10\}, \{-5, -7\}, \{-4, -9\}, \{-2, -11\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -
8728:
                         \{8, -7\}, \{8, -4\}, \{7, 0\}, \{6, 3\}, \{4, 6\}, \{2, 8\}, \{-1, 9\}, \{-4, 9\},
8729:
                         \{-6,8\},\{-7,6\},\{-7,4\},\{-5,4\},\{-5,6\},\{-6,6\},\{-6,5\},
8730: };
8731: static XPoint seg1_3259[] = {
8732:
                        \{-3, -2\}, \{-4, -4\}, \{-4, -7\}, \{-3, -9\},
8733: };
8734: static XPoint seg2 3259[] = {
8735:
                        \{6, -10\}, \{7, -8\}, \{7, -4\}, \{6, 0\}, \{5, 3\}, \{3, 6\},\
8736: };
8737: static XPoint seg3 3259[] = {
8738:
                        \{-1,0\},\{-2,-1\},\{-3,-3\},\{-3,-7\},\{-2,-10\},\{-1,-11\},\{1,-12\},
8739: };
8740: static XPoint seg4_3259[] = {
8741:
                         \{4, -12\}, \{5, -11\}, \{6, -9\}, \{6, -4\}, \{5, 0\}, \{4, 3\}, \{3, 5\}, \{1, 8\},
8742:
                         \{-1,9\},\
8743: };
8744: static XPoint *char3259[] = {
8745:
                         seg0 3259, seg1 3259, seg2 3259, seg3 3259, seg4 3259,
8746:
                        NULL,
8747: };
8748: static int char p3259[] = {
8749:
                        XtNumber(seg0 3259), XtNumber(seg1 3259), XtNumber(seg2 3259),
8750:
                        XtNumber(seg3 3259), XtNumber(seg4 3259),
8751: };
8752: static XPoint seg0 3260[] = {
8753:
                        \{-2,6\},\{-3,7\},\{-3,8\},\{-2,9\},\{-1,9\},\{0,8\},\{0,7\},\{-1,6\},
8754:
                         \{-2, 6\},\
8755: };
8756: static XPoint seg1 3260[] = {
8757:
                         \{-2,7\},\{-2,8\},\{-1,8\},\{-1,7\},\{-2,7\},
8758: };
8759: static XPoint *char3260[] = {
8760:
                         seg0 3260, seg1 3260,
8761:
                         NULL,
8762: };
```

```
8763: static int char p3260[] = {
8764:
                                                           XtNumber(seg0_3260),XtNumber(seg1_3260),
8765: };
8766: static XPoint seg0 3261[] = {
8767:
                                                            \{-1,9\},\{-2,9\},\{-3,8\},
8768:
                                                            \{-3,7\},\{-2,6\},\{-1,6\},\{0,7\},\{0,9\},\{-1,11\},\{-2,12\},\{-4,13\},
8769: };
8770: static XPoint seg1 3261[] = {
8771:
                                                            \{-2,7\},\{-2,8\},\{-1,8\},\{-1,7\},\{-2,7\},
8772: };
8773: static XPoint seg2 3261[] = {
8774:
                                                           \{-1,9\},\{-1,10\},\{-2,12\},
8775: };
8776: static XPoint *char3261[] = {
8777:
                                                           seg0 3261, seg1 3261, seg2 3261,
8778:
                                                           NULL,
8779: };
8780: static int char p3261[] = {
8781:
                                                           XtNumber(seg0_3261), XtNumber(seg1_3261), XtNumber(seg2_3261),
8782: };
8783: static XPoint seg0 3262[] = {
8784:
                                                           \{1, -5\}, \{0, -4\}, \{0, -3\}, \{1, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{2, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{3, -2\}, \{
8785:
                                                            \{3, -3\}, \{3, -4\}, \{2, -5\}, \{1, -5\},\
8786: };
8787: static XPoint seq1 3262[] = {
8788:
                                                           \{1, -4\}, \{1, -3\}, \{2, -3\}, \{2, -4\}, \{1, -4\},
8789: };
8790: static XPoint seg2 3262[] = {
8791:
                                                           \{-2,6\}, \{-3,7\}, \{-3,8\}, \{-2,9\}, \{-1,9\}, \{0,8\}, \{0,7\}, \{-1,6\}, \{-2,6\}, \{-2,6\}, \{-3,7\}, \{-3,8\}, \{-2,9\}, \{-1,9\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}
8792:
                                                            \{-2, 6\},\
8793: };
8794: static XPoint seg3 3262[] = {
8795:
                                                           \{-2,7\},\{-2,8\},\{-1,8\},\{-1,7\},\{-2,7\},
8796: };
8797: static XPoint *char3262[] = {
8798:
                                                            seg0 3262, seg1 3262, seg2 3262, seg3 3262,
                                                           NULL,
8799:
8800: };
8801: static int char p3262[] = {
8802:
                                                           XtNumber(seg0 3262), XtNumber(seg1 3262), XtNumber(seg2 3262),
8803:
                                                           XtNumber(seg3 3262),
8804: };
8805: static XPoint seg0 3263[] = {
                                                            \{1, -5\}, \{0, -4\}, \{0, -3\},\
8806:
8807:
                                                            \{1, -2\}, \{2, -2\}, \{3, -3\}, \{3, -4\}, \{2, -5\}, \{1, -5\},\
8808: };
8809: static XPoint seg1_3263[] = {
8810:
                                                           \{1, -4\}, \{1, -3\}, \{2, -3\}, \{2, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{1, -4\}, \{
8811: };
8812: static XPoint seg2 3263[] = {
8813:
                                                            \{-1,9\},\{-2,9\},\{-3,8\},\{-3,7\},\{-2,6\},\{-1,6\},\{0,7\},\{0,9\},
8814:
                                                            \{-1,11\},\{-2,12\},\{-4,13\},
8815: };
```

```
Listing C.4 Continued
```

```
8816: static XPoint seg3 3263[] = {
                                                                           \{-2,7\},\{-2,8\},\{-1,8\},\{-1,7\},\{-2,7\},
8817:
8818: };
8819: static XPoint seg4 3263[] = {
8820:
                                                                              \{-1,9\},\{-1,10\},\{-2,12\},
8821: };
8822: static XPoint *char3263[] = {
                                                                              seg0_3263, seg1_3263, seg2_3263, seg3_3263, seg4_3263,
8823:
8824:
                                                                             NULL,
8825: };
8826: static int char_p3263[] = {
8827:
                                                                              XtNumber(seg0 3263), XtNumber(seg1 3263), XtNumber(seg2 3263),
8828:
                                                                             XtNumber(seg3_3263),XtNumber(seg4_3263),
8829: };
8830: static XPoint seg0 3264[] = {
8831:
                                                                             \{4, -12\}, \{3, -12\}, \{2, -11\}, \{0, 2\},\
8832: };
8833: static XPoint seg1 3264[] = {
8834:
                                                                              \{4, -11\}, \{3, -11\}, \{0, 2\},\
8835: };
8836: static XPoint seg2 3264[] = {
8837:
                                                                             \{4, -11\}, \{4, -10\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, 
8838: };
8839: static XPoint seg3 3264[] = {
8840:
                                                                             \{4, -12\}, \{5, -11\}, \{5, -10\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}, \{0, 2\}
8841: };
8842: static XPoint seg4 3264[] = {
8843:
                                                                              \{-2,6\},\{-3,7\},\{-3,8\},\{-2,9\},\{-1,9\},\{0,8\},\{0,7\},\{-1,6\},
8844:
                                                                              \{-2,6\},\
8845: };
8846: static XPoint seg5 3264[] = {
8847:
                                                                           \{-2,7\},\{-2,8\},\{-1,8\},\{-1,7\},\{-2,7\},
8848: };
8849: static XPoint *char3264[] = {
                                                                              seg0 3264, seg1 3264, seg2 3264, seg3 3264, seg4 3264,
8850:
8851:
                                                                              seg5 3264,
8852:
                                                                             NULL,
8853: };
8854: static int char_p3264[] = {
8855:
                                                                             XtNumber(seg0 3264), XtNumber(seg1 3264), XtNumber(seg2 3264),
8856:
                                                                             XtNumber(seg3_3264),XtNumber(seg4_3264),XtNumber(seg5_3264),
8857: };
8858: static XPoint seg0_3265[] = {
8859:
                                                                              \{-3, -7\}, \{-3, -8\}, \{-2, -8\}, \{-2, -6\}, \{-4, -6\}, \{-4, -8\}, \{-3, -10\},
8860:
                                                                              \{-2, -11\}, \{1, -12\}, \{5, -12\}, \{8, -11\}, \{9, -9\}, \{9, -7\}, \{8, -5\}, \{9, -7\}, \{8, -5\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -7\}, \{9, -
8861:
                                                                             \{7, -4\}, \{5, -3\}, \{1, -2\}, \{-1, -1\}, \{-1, 1\}, \{1, 2\}, \{2, 2\}, \{2, 2\}, \{-1, -1\}, \{-1, 1\}, \{-1, 1\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-1, 2\}, \{-
8862: };
8863: static XPoint seg1 3265[] = {
8864:
                                                                              \{3, -12\}, \{8, -11\},\
8865: };
```

```
8866: static XPoint seg2 3265[] = {
8867:
                                                                                                                        \{7, -11\}, \{8, -9\}, \{8, -7\}, \{7, -5\}, \{6, -4\}, \{4, -3\}, \{6, -4\}, \{4, -3\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{7, -4\}, \{8, -4\}, \{8, -4\}, \{7, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, \{6, -4\}, 
8868: };
8869: static XPoint seg3 3265[] = {
8870:
                                                                                                                        \{5, -12\}, \{6, -11\}, \{7, -9\}, \{7, -7\}, \{6, -5\}, \{5, -4\}, \{1, -2\}, \{0, -1\},
8871:
                                                                                                                        \{0,1\},\{1,2\},\
8872: };
8873: static XPoint seg4 3265[] = {
8874:
                                                                                                                        \{-2,6\}, \{-3,7\}, \{-3,8\}, \{-2,9\}, \{-1,9\}, \{0,8\}, \{0,7\}, \{-1,6\}, \{-2,6\}, \{-3,7\}, \{-3,8\}, \{-2,9\}, \{-1,9\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}, \{-3,8\}
8875:
                                                                                                                        \{-2,6\},\
8876: };
8877: static XPoint seg5_3265[] = {
8878:
                                                                                                                      \{-2,7\},\{-2,8\},\{-1,8\},\{-1,7\},\{-2,7\},
8879: };
8880: static XPoint *char3265[] = {
8881:
                                                                                                                        seg0 3265, seg1 3265, seg2 3265, seg3 3265, seg4 3265,
8882:
                                                                                                                        seg5_3265,
                                                                                                                      NULL,
8883:
8884: };
8885: static int char p3265[] = {
8886:
                                                                                                                      XtNumber(seg0 3265), XtNumber(seg1 3265), XtNumber(seg2 3265),
8887:
                                                                                                                      XtNumber(seg3 3265),XtNumber(seg4_3265),XtNumber(seg5_3265),
8888: };
8889: static XPoint seg0 3266[] = {
8890:
                                                                                                                        \{5, -12\}, \{3, -11\}, \{2, -10\}, 
8891:
                                                                                                                        \{1, -8\}, \{1, -6\}, \{2, -5\}, \{3, -5\}, \{4, -6\}, \{4, -7\}, \{3, -8\}, \{2, -8\}, \{2, -8\}, \{3, -8\}, \{2, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{3, -8\}, \{
8892: };
8893: static XPoint seg1 3266[] = {
8894:
                                                                                                                      \{3, -11\}, \{2, -9\}, \{2, -8\}, \{2, -8\}, \{2, -8\}, \{2, -8\}, \{2, -8\}, \{2, -8\}, \{2, -8\}, \{2, -8\}, \{2, -8\}, \{3, -11\}, \{2, -8\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -
8895: };
8896: static XPoint seg2 3266[] = {
8897:
                                                                                                                      \{2, -7\}, \{2, -6\}, \{3, -6\}, \{3, -7\}, \{2, -7\}, \{2, -7\}, \{2, -7\}, \{2, -7\}, \{2, -7\}, \{2, -7\}, \{2, -7\}, \{2, -7\}, \{2, -7\}, \{3, -7\}, \{2, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{3, -7\}, \{
8898: };
8899: static XPoint *char3266[] = {
8900:
                                                                                                                        seg0_3266, seg1_3266, seg2_3266,
8901:
                                                                                                                      NULL,
8902: };
8903: static int char p3266[] = {
8904:
                                                                                                                      XtNumber(seg0 3266), XtNumber(seg1 3266), XtNumber(seg2 3266),
8905: };
8906: static XPoint seg0 3267[] = {
8907:
                                                                                                                        \{4, -9\}, \{3, -9\}, \{2, -10\}, \{2, -10\}, \{2, -10\}, \{3, -9\}, \{2, -10\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{3, -9\}, \{4, -9\}, \{3, -9\}, \{4, -9\}, \{4, -9\}, \{3, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{3, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9\}, \{4, -9
8908:
                                                                                                                        \{2, -11\}, \{3, -12\}, \{4, -12\}, \{5, -11\}, \{5, -9\}, \{4, -7\}, \{3, -6\}, \{1, -5\},
8909: };
8910: static XPoint seg1 3267[] = {
8911:
                                                                                                                      \{3, -11\}, \{3, -10\}, \{4, -10\}, \{4, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, -11\}, \{3, 
8912: };
8913: static XPoint seg2 3267[] = {
8914:
                                                                                                                      \{4, -9\}, \{4, -8\}, \{3, -6\},\
8915: };
8916: static XPoint *char3267[] = {
8917:
                                                                                                                        seg0 3267, seg1 3267, seg2 3267,
8918:
                                                                                                                      NULL,
8919: };
```
Listing C.4 Continued

```
8920: static int char_p3267[] = {
8921:
                                                                                      XtNumber(seg0 3267), XtNumber(seg1 3267), XtNumber(seg2 3267),
8922: };
8923: static XPoint seg0 3268[] = {
8924:
                                                                                        \{10, -3\}, \{10, -4\}, \{9, -4\}, \{9, -2\}, \{11, -2\}, \{11, -4\}, \{10, -5\}, \{9, -5\}, \{9, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}, \{10, -5\}
8925:
                                                                                        \{7, -4\}, \{5, -2\}, \{0, 6\}, \{-2, 8\}, \{-4, 9\}, \{-7, 9\}, \{-10, 8\}, \{-11, 6\},
8926:
                                                                                        \{-11,4\},\{-10,2\},\{-9,1\},\{-7,0\},\{-2,-2\},\{0,-3\},
8927:
                                                                                        \{2, -5\}, \{3, -7\}, \{3, -9\}, \{2, -11\}, \{0, -12\}, \{-2, -11\}, \{-3, -9\}, \{-2, -11\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, -9\}, \{-3, 
8928:
                                                                                        \{-3, -6\}, \{-2, 0\}, \{-1, 3\}, \{0, 5\}, \{2, 8\}, \{4, 9\}, \{6, 9\}, \{7, 7\}, \{7, 6\},
8929: };
8930: static XPoint seg1 3268[] = {
8931:
                                                                                      \{-6,9\},\{-10,8\},
8932: };
8933: static XPoint seg2 3268[] = {
8934:
                                                                                      \{-9,8\},\{-10,6\},\{-10,4\},\{-9,2\},\{-8,1\},\{-6,0\},
8935: };
8936: static XPoint seg3 3268[] = {
8937:
                                                                                      \{-2, -2\}, \{-1, 1\}, \{2, 7\}, \{4, 8\},
8938: };
8939: static XPoint seq4 3268[] = {
8940:
                                                                                        \{-7,9\},\{-8,8\},\{-9,6\},\{-9,4\},\{-8,2\},\{-7,1\},\{-5,0\},\{0,-3\},
8941: };
8942: static XPoint seq5 3268[] = {
8943:
                                                                                        \{-3, -6\}, \{-2, -3\}, \{-1, 0\}, \{1, 4\}, \{3, 7\}, \{5, 8\}, \{6, 8\}, \{7, 7\}, \{5, 8\}, \{6, 8\}, \{7, 7\}, \{6, 8\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\}, \{7, 7\},
8944: };
8945: static XPoint *char3268[] = {
                                                                                        seg0 3268, seg1 3268, seg2 3268, seg3 3268, seg4 3268,
8946:
8947:
                                                                                        seg5_3268,
8948:
                                                                                      NULL,
8949: };
8950: static int char p3268[] = {
8951:
                                                                                      XtNumber(seg0 3268), XtNumber(seg1 3268), XtNumber(seg2 3268),
8952:
                                                                                      XtNumber(seg3_3268),XtNumber(seg4_3268),XtNumber(seg5_3268),
8953: };
8954: static XPoint seg0_3269[] = {
8955:
                                                                                      \{2, -16\}, \{-6, 13\}, 
8956: };
8957: static XPoint seg1 3269[] = {
8958:
                                                                                      \{7, -16\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1, 13\}, \{-1,
8959: };
8960: static XPoint seg2 3269[] = {
                                                                                        \{8, -7\}, \{8, -8\}, \{7, -8\}, \{7, -6\}, \{9, -6\}, \{9, -8\}, \{8, -10\}, \{7, -11\},
8961:
8962:
                                                                                        \{4, -12\}, \{0, -12\}, \{-3, -11\}, \{-5, -9\}, \{-5, -6\}, \{-4, -4\}, \{-2, -2\}, \{-3, -11\}, \{-2, -2\}, \{-3, -11\}, \{-2, -2\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -11\}, \{-3, -
8963:
                                                                                        \{4,1\},\{5,3\},\{5,6\},\{4,8\},
8964: };
8965: static XPoint seg3 3269[] = {
8966:
                                                                                        \{-4, -6\}, \{-3, -4\}, \{4, 0\}, \{5, 2\},
8967: };
```

```
8968: static XPoint seg4 3269[] = {
                                          \{-3, -11\}, \{-4, -9\}, \{-4, -7\}, \{-3, -5\}, \{3, -2\}, \{5, 0\}, \{6, 2\}, \{6, 5\},
8969:
8970:
                                          \{5,7\},\{4,8\},\{1,9\},\{-3,9\},\{-6,8\},\{-7,7\},\{-8,5\},\{-8,3\},
8971:
                                          \{-6,3\},\{-6,5\},\{-7,5\},\{-7,4\},
8972: };
8973: static XPoint *char3269[] = {
8974:
                                          seg0 3269, seg1 3269, seg2 3269, seg3 3269, seg4 3269,
8975:
                                         NULL.
8976: };
8977: static int char p3269[] = {
8978:
                                         XtNumber(seg0 3269), XtNumber(seg1 3269), XtNumber(seg2 3269),
                                         XtNumber(seg3 3269),XtNumber(seg4 3269),
8979:
8980: };
8981: static XPoint seg0 3270[] = {
8982:
                                          \{13, -16\}, \{-13, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}
8983: };
8984: static XPoint seg1_3270[] = {
8985:
                                          \{13, -16\}, \{14, -16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}, \{-12, 16\}
8986: };
8987: static XPoint *char3270[] = {
8988:
                                          seg0 3270, seg1 3270,
8989:
                                          NULL,
8990: };
8991: static int char p3270[] = {
8992:
                                         XtNumber(seg0_3270),XtNumber(seg1_3270),
8993: };
8994: static XPoint seq0 3271[] = {
8995:
                                          \{8, -16\}, \{6, -15\}, \{3, -13\}, \{0, -10\}, \{-2, -7\},\
8996:
                                          \{-4, -3\}, \{-5, 1\}, \{-5, 6\}, \{-4, 10\}, \{-3, 13\}, \{-1, 16\},
8997: };
8998: static XPoint seg1 3271[] = {
8999:
                                          \{1, -10\}, \{-1, -7\}, \{-3, -3\}, \{-4, 2\}, \{-4, 10\},
9000: };
9001: static XPoint seg2 3271[] = {
9002:
                                          \{8, -16\}, \{5, -14\}, \{2, -11\}, \{0, -8\}, \{-1, -6\}, \{-2, -3\}, \{-3, 1\},
9003:
                                          \{-4, 10\},\
9004: };
9005: static XPoint seg3 3271[] = {
9006:
                                          \{-4,2\},\{-3,11\},\{-2,14\},\{-1,16\},
9007: };
9008: static XPoint *char3271[] = {
9009:
                                          seg0 3271, seg1 3271, seg2 3271, seg3 3271,
9010:
                                          NULL,
9011: };
9012: static int char p3271[] = {
9013:
                                         XtNumber(seg0 3271), XtNumber(seg1 3271),
9014:
                                         XtNumber(seg2_3271),XtNumber(seg3_3271),
9015: };
9016: static XPoint seg0_3272[] = {
9017:
                                         \{1, -16\}, \{3, -13\}, \{4, -10\}, \{5, -6\},\
9018:
                                          \{5, -1\}, \{4, 3\}, \{2, 7\}, \{0, 10\}, \{-3, 13\}, \{-6, 15\}, \{-8, 16\},
9019: };
```

C

Listing C.4 Continued

```
9020: static XPoint seg1 3272[] = {
9021:
                                                                                   \{4, -10\}, \{4, -2\}, \{3, 3\}, \{1, 7\}, \{-1, 10\},\
9022: };
9023: static XPoint seg2 3272[] = {
9024:
                                                                                      \{1, -16\}, \{2, -14\}, \{3, -11\}, \{4, -2\},\
9025: };
9026: static XPoint seg3 3272[] = {
9027:
                                                                                     \{4, -10\}, \{3, -1\}, \{2, 3\}, \{1, 6\}, \{0, 8\}, \{-2, 11\}, \{-5, 14\}, \{-8, 16\},
9028: };
9029: static XPoint *char3272[] = {
9030:
                                                                                     seg0 3272, seg1 3272, seg2 3272, seg3 3272,
9031:
                                                                                     NULL,
9032: };
9033: static int char_p3272[] = {
9034:
                                                                                     XtNumber(seg0 3272), XtNumber(seg1 3272),
9035:
                                                                                     XtNumber(seg2 3272),XtNumber(seg3 3272),
9036: };
9037: static XPoint seg0 3273[] = {
9038:
                                                                                      \{2, -12\}, \{1, -11\}, \{3, -1\}, \{2, 0\},\
9039: };
9040: static XPoint seg1 3273[] = {
9041:
                                                                                     \{2, -12\}, \{2, 0\},\
9042: };
9043: static XPoint seg2 3273[] = {
9044:
                                                                                     \{2, -12\}, \{3, -11\}, \{1, -1\}, \{2, 0\},\
9045: };
9046: static XPoint seg3 3273[] = {
9047:
                                                                                     \{-3, -9\}, \{-2, -9\}, \{6, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\},
9048: };
9049: static XPoint seg4_3273[] = {
9050:
                                                                                     \{-3, -9\}, \{7, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, 
9051: };
9052: static XPoint seg5 3273[] = {
9053:
                                                                                      \{-3, -9\}, \{-3, -8\}, \{7, -4\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\}, \{7, -3\},
9054: };
9055: static XPoint seg6 3273[] = {
9056:
                                                                                     \{7, -9\}, \{6, -9\}, \{-2, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -3\}, \{-3, -
9057: };
9058: static XPoint seg7 3273[] = {
9059:
                                                                                     \{7, -9\}, \{-3, -3\},\
9060: };
9061: static XPoint seg8_3273[] = {
9062:
                                                                                      \{7, -9\}, \{7, -8\}, \{-3, -4\}, \{-3, -3\},
9063: };
9064: static XPoint *char3273[] = {
9065:
                                                                                      seg0 3273, seg1 3273, seg2 3273, seg3 3273,
9066:
                                                                                      seg4 3273, seg5 3273, seg6 3273, seg7 3273, seg8 3273,
9067:
                                                                                     NULL,
9068: };
```

```
9069: static int char p3273[] = {
9070:
                                                              XtNumber(seg0_3273),XtNumber(seg1_3273),XtNumber(seg2_3273),
9071:
                                                              XtNumber(seg3 3273), XtNumber(seg4 3273), XtNumber(seg5 3273),
9072:
                                                              XtNumber(seg6 3273), XtNumber(seg7 3273), XtNumber(seg8 3273),
9073: };
9074: static XPoint seg0 3274[] = {
9075:
                                                              \{-8, -1\}, \{9, -1\}, \{9, 0\},\
9076: };
9077: static XPoint seg1 3274[] = {
9078:
                                                              \{-8, -1\}, \{-8, 0\}, \{9, 0\},
9079: };
9080: static XPoint *char3274[] = {
9081:
                                                              seg0 3274, seg1 3274,
9082:
                                                              NULL,
9083: };
9084: static int char p3274[] = {
9085:
                                                             XtNumber(seg0_3274), XtNumber(seg1_3274),
9086: };
9087: static XPoint seg0 3275[] = {
9088:
                                                             \{0, -9\}, \{0, 8\}, \{1, 8\},\
9089: };
9090: static XPoint seq1 3275[] = {
9091:
                                                               \{0, -9\}, \{1, -9\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1, 8\}, \{1
9092: };
9093: static XPoint seg2 3275[] = {
9094:
                                                              \{-8, -1\}, \{9, -1\}, \{9, 0\},
9095: };
9096: static XPoint seg3 3275[] = {
9097:
                                                              \{-8, -1\}, \{-8, 0\}, \{9, 0\}, \{9, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-8, 0\}, \{-
9098: };
9099: static XPoint *char3275[] = {
9100:
                                                               seg0_3275, seg1_3275, seg2_3275, seg3_3275,
9101:
                                                              NULL,
9102: };
9103: static int char_p3275[] = {
                                                              XtNumber(seg0 3275),XtNumber(seg1 3275),
9104:
9105:
                                                              XtNumber(seg2 3275),XtNumber(seg3 3275),
9106: };
9107: static XPoint seg0 3276[] = {
9108:
                                                               \{-8, -5\}, \{9, -5\}, \{9, -4\},\
9109: };
9110: static XPoint seg1 3276[] = {
9111:
                                                              \{-8, -5\}, \{-8, -4\}, \{9, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, -4\}, \{-8, 
9112: };
9113: static XPoint seg2 3276[] = {
9114:
                                                              \{-8,3\},\{9,3\},\{9,4\},
9115: };
9116: static XPoint seg3 3276[] = {
9117:
                                                              \{-8,3\},\{-8,4\},\{9,4\},
9118: };
9119: static XPoint *char3276[] = {
9120:
                                                               seg0 3276, seg1 3276, seg2 3276, seg3 3276,
9121:
                                                              NULL,
9122: };
```

C

```
Listing C.4 Continued
```

```
9123: static int char p3276[] = {
9124:
                                                                         XtNumber(seg0_3276),XtNumber(seg1_3276),
9125:
                                                                           XtNumber(seg2 3276), XtNumber(seg3 3276),
9126: };
9127: static XPoint seg0 3278[] = {
9128:
                                                                           \{-2, -12\}, \{-3, -11\}, \{-5, -5\}, \{-2, -12\}, \{-3, -11\}, \{-1, -12\}, \{-1, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\}, \{-2, -12\},
9129: };
9130: static XPoint seg1 3278[] = {
9131:
                                                                        \{-2, -11\}, \{-5, -5\},
9132: };
9133: static XPoint seg2_3278[] = {
9134:
                                                                           \{-2, -12\}, \{-1, -11\}, \{-5, -5\},
9135: };
9136: static XPoint seg3 3278[] = {
9137:
                                                                           \{8, -12\}, \{7, -11\}, \{5, -5\},\
9138: };
9139: static XPoint seg4 3278[] = {
9140:
                                                                         \{8, -11\}, \{5, -5\},\
9141: };
9142: static XPoint seg5_3278[] = {
9143:
                                                                           \{8, -12\}, \{9, -11\}, \{5, -5\},\
9144: };
9145: static XPoint *char3278[] = {
9146:
                                                                            seg0 3278, seg1 3278, seg2 3278, seg3 3278, seg4 3278,
9147:
                                                                            seg5_3278,
9148:
                                                                           NULL,
9149: };
9150: static int char p3278[] = {
9151:
                                                                           XtNumber(seg0_3278), XtNumber(seg1_3278), XtNumber(seg2_3278),
9152:
                                                                           XtNumber(seg3 3278), XtNumber(seg4 3278), XtNumber(seg5 3278),
9153: };
9154: static XPoint seg0 3279[] = {
9155:
                                                                            \{1, -12\}, \{-1, -11\}, \{-2, -9\}, \{-2, -7\}, \{-1, -5\}, \{1, -4\},
9156:
                                                                            \{3, -4\}, \{5, -5\}, \{6, -7\}, \{6, -9\}, \{5, -11\}, \{3, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{2, -11\}, \{3, -12\}, \{1, -12\}, \{1, -12\}, \{2, -11\}, \{3, -12\}, \{1, -12\}, \{1, -12\}, \{1, -12\}, \{2, -11\}, \{2, -12\}, \{3, -12\}, \{3, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}, \{4, -12\}
9157: };
9158: static XPoint seg1 3279[] = {
9159:
                                                                           \{1, -12\}, \{-2, -9\}, \{-1, -5\}, \{3, -4\}, \{6, -7\}, \{5, -11\}, \{1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -12\}, \{-1, -1
9160: };
9161: static XPoint seg2 3279[] = {
9162:
                                                                           \{3, -12\}, \{-1, -11\}, \{-2, -7\}, \{1, -4\}, \{5, -5\}, \{6, -9\}, \{3, -12\}, \{-1, -11\}, \{-2, -7\}, \{-1, -4\}, \{-2, -5\}, \{-1, -11\}, \{-2, -7\}, \{-1, -4\}, \{-2, -5\}, \{-1, -11\}, \{-2, -7\}, \{-1, -4\}, \{-2, -5\}, \{-1, -12\}, \{-2, -7\}, \{-1, -4\}, \{-2, -7\}, \{-1, -4\}, \{-2, -7\}, \{-1, -4\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-2, -7\}, \{-
9163: };
9164: static XPoint *char3279[] = {
9165:
                                                                   seg0_3279, seg1_3279, seg2_3279,
9166:
                                                                   NULL,
9167: };
9168: static int char p3279[] = {
9169:
                                                                   XtNumber(seg0 3279),XtNumber(seg1 3279),XtNumber(seg2 3279),
9170: };
9171:
9172: /*
9173: * a full char set (and charP's)
9174: */
```

9175:	int * VCharPSe	t[] = {		
9176:	char_p501,	char_p502,	char_p503,	char_p504,
9177:	char_p505,	char_p506,	char_p507,	char_p508,
9178:	char_p509,	char_p510,	char_p511,	char_p512,
9179:	char_p513,	char_p514,	char_p515,	char_p516,
9180:	char p517,	char p518,	char p519,	char p520,
9181:	char p521,	char p522,	char p523,	char p524,
9182:	char p525,	char p526,	char p551,	char p552,
9183:	char p553,	char p554,	char p555,	char p556,
9184:	char p557,	char p558,	char p559,	char p560,
9185:	char_p561,	char_p562,	char_p563,	char_p564,
9186:	char p565,	char p566,	char p567,	char p568,
9187:	char p569,	char p570,	char p571,	char p572,
9188:	char p573,	char p574,	char p575,	char p576,
9189:	char p601,	char p602,	char p603,	char p604,
9190:	char_p605,	char_p606,	char p607,	char_p608,
9191:	char p609,	char p610,	char p611,	char p612,
9192:	char p613,	char p614,	char p615,	char p616,
9193:	char p617,	char p618,	char p619,	char p620,
9194:	char p621,	char p622,	char p623,	char p624,
9195:	char p625,	char p626,	char p651,	char p652,
9196:	char p653,	char p654,	char p655,	char p656,
9197:	char p657,	char p658,	char p659,	char p660,
9198:	char p661,	char p662,	char p663,	char p664,
9199:	char p665,	char_p666,	char p667,	char p668,
9200:	char p669,	char p670,	char p671,	char p672,
9201:	char p673,	char p674,	char p675,	char p676,
9202:	char p699,	char p700,	char p701,	char p702,
9203:	char p703,	char p704,	char p705,	char p706,
9204:	char p707,	char p708,	char p709,	char p710,
9205:	char p711,	char p712,	char p713,	char p714,
9206:	char p715,	char p717,	char p718,	char p719,
9207:	char p720,	char p721,	char p722,	char p723,
9208:	char_p724,	char_p725,	char_p726,	char_p730,
9209:	char_p731,	char_p733,	char_p734,	char_p804,
9210:	char_p834,	char_p840,	char_p844,	char_p845,
9211:	char_p847,	char_p850,	char_p855,	char_p866,
9212:	char_p999,	char_p2219,	char_p2223,	char_p2224,
9213:	char_p2225,	char_p2226,	char_p2229,	char_p2241,
9214:	char_p2242,	char_p2246,	char_p2262,	char_p2271,
9215:	char_p2273,	char_p2275,	char_p2750,	char_p2751,
9216:	char_p2752,	char_p2753,	char_p2754,	char_p2755,
9217:	char_p2756,	char_p2757,	char_p2758,	char_p2759,
9218:	char_p2761,	char_p2762,	char_p2763,	char_p2764,
9219:	char_p2765,	char_p2766,	char_p2767,	char_p2768,
9220:	char_p2769,	char_p2770,	char_p2771,	char_p2772,
9221:	char_p2773,	char_p2778,	char_p3001,	char_p3002,
9222:	char_p3003,	char_p3004,	char_p3005,	char_p3006,
9223:	char_p3007,	char_p3008,	char_p3009,	char_p3010,
9224:	char_p3011,	char_p3012,	char_p3013,	char_p3014,
9225:	char_p3015,	char_p3016,	char_p3017,	char_p3018,
9226:	char_p3019,	char_p3020,	char_p3021,	char_p3022,
9227:	char_p3023,	char_p3024,	char_p3025,	char_p3026,

C

Listing C.4 Continued

9228:	char p3051.	char p3052.	char p3053.	char p3054.
9229:	char_p3055.	char_p3056.	char_p3057.	char p3058.
9230:	char p3059.	char p3060.	char p3061.	char p3062.
9231:	char p3063.	char p3064.	char p3065.	char p3066.
9232:	char p3067.	char p3068.	char p3069.	char p3070.
9233:	char p3071.	char p3072.	char p3073.	char p3074.
9234:	char p3075,	char p3076,	char p3101,	char p3102,
9235:	char_p3103,	char p3104,	char p3105,	char p3106,
9236:	char_p3107,	char p3108,	char p3109,	char p3110,
9237:	char p3111,	char p3112,	char p3113,	char p3114,
9238:	char p3115,	char p3116,	char p3117,	char p3118,
9239:	char_p3119,	char_p3120,	char_p3121,	char_p3122,
9240:	char_p3123,	char_p3124,	char_p3125,	char_p3126,
9241:	char_p3151,	char_p3152,	char_p3153,	char_p3154,
9242:	char_p3155,	char_p3156,	char_p3157,	char_p3158,
9243:	char_p3159,	char_p3160,	char_p3161,	char_p3162,
9244:	char_p3163,	char_p3164,	char_p3165,	char_p3166,
9245:	char_p3167,	char_p3168,	char_p3169,	char_p3170,
9246:	char_p3171,	char_p3172,	char_p3173,	char_p3174,
9247:	char_p3175,	char_p3176,	char_p3199,	char_p3200,
9248:	char_p3201,	char_p3202,	char_p3203,	char_p3204,
9249:	char_p3205,	char_p3206,	char_p3207,	char_p3208,
9250:	char_p3209,	char_p3210,	char_p3211,	char_p3212,
9251:	char_p3213,	char_p3214,	char_p3215,	char_p3216,
9252:	char_p3217,	char_p3218,	char_p3219,	char_p3220,
9253:	char_p3221,	char_p3222,	char_p3223,	char_p3224,
9254:	char_p3225,	char_p3226,	char_p3228,	char_p3229,
9255:	char_p3249,	char_p3250,	char_p3251,	char_p3252,
9256:	char_p3253,	char_p3254,	char_p3255,	char_p3256,
9257:	char_p3257,	char_p3258,	char_p3259,	char_p3260,
9258:	char_p3261,	char_p3262,	char_p3263,	char_p3264,
9259:	char_p3265,	char_p3266,	char_p3267,	char_p3268,
9260:	char_p3269,	char_p3270,	char_p3271,	char_p3272,
9261:	char_p3273,	char_p3274,	char_p3275,	char_p3276,
9262:	char_p3278,	char_p3279	};	
9263:	<pre>#endif /* VEC_I</pre>	FONTS_INC_H	* /	

Index

Symbols

& (ampersand), 80, 524 && (and symbol), 54 * (asterisk), 70 * (asterisk) wildcard, 13 */ (asterisk slash), 84 @ (at sign), 36 \ (backslash), 33 [] (brackets), 7, 532 { } body markers, 77 : (colon), 24, 35 { } (curly braces), 47, 59, 61 . (current directory), 14 - (dash), 532 \$ (dollar sign), 33 .. (double dot), 11 >> (double greater than symbol), 46 " " (double quotation marks), 93, 532 ;; (double semicolon), 534 ... (ellipses), 63 = (equal sign), 526 = = (equal to symbol), 53 >= (greater than or equal to symbol), 53 > (greater than symbol),

45.53 - (hyphen), 6-7 <= (less than or equal to symbol), 53 < (less than symbol), 53 != (not equal to symbol), 53 ! (not symbol), 54, 63 || (or symbol), 54 .. (parent directory), 14 % (percent sign), 93, 482, 524 . (period), 14 | (pipe symbol), 45 # (pound sign), 32, 39, 91 ; (semicolon), 47, 61, 527 . (single dot), 11 ' ' (single quotation marks), 93, 532 /* (slash asterisk), 84 // (slash slash), 84 ~ (tilde), 12 #! token, 531 _ (underscore), 67 16-bit processor, 20 2D Graphical Editor, 203-204 2d-editor root directory, 540

32-bit processors, 20 80286 processors, 20

A

-a flag, 14, 527 [a-d] wildcard, 13 aborting creation modes, 382 abs function, 214 access (group), 16 accessing data by reference, 69, 78-81 action fields, 317, 337 actions, 52, 303 active, 398 assigning, 169, 336 copy, 317 gx_line, 382 invoking, 336 move, 396 scale, 396 Activate function, 303 activate_objs function, 333 active action, 398 active colormaps, 185 active objects, 316

add PostScript operator, 484 add2Nums function, 89 adding objects to linked lists, 387 points to arrays, 383, 388 addresses, memory, 70. See also pointers alias command, 527 aliases, 526-527 allocated memory, returning, 106 allocating colors, 184-187 memory, 103-107, 236 allowable colors, 510 AllPlanes argument, 499 American National **Standards Institute** (ANSI), 55 American Standard Code for Information Interchange (ASCII), 54-55 ampersand (&), 80, 524 analysis (lexical), 56 and symbol (&&), 54 angle1 field, 511 angle2 field, 511 angles (theta), 255 angular deflection (polar coordinate system), 255 **ANSI** (American National Standard Institute), 55 ANSI C, 25. See also C programming language application interfaces, 151-158 applications, 119 creating windows in, 123 Graphics Editor. See Graphics Editor

structure, 262-263 X, 126-127 X Window, 301 X-based application interfaces, creating, 151-158 components, 143 connecting to X server; 144 buttons, creating, 159-162 managing windows, 175-176 processing events, 176 xcalc, 119-121 applications cursor, updating, 338-339 ApplicationShell widget, 133-135, 141 creating, 150 resources, 134 apply_delta function, 245, 399, 469 apply_delta_<direction> function, 247 apply_delta_bottom_side function, 240-242 apply_delta_left_side function, 240-242 apply_delta_right_side function, 240-242 apply_delta_top_side function, 240-242 apply_scale function, 455-457, 471-472 applying source pixels, 198 apropos command, 7 arc angles, 511 arc object, 314-315, 363, 511 bounds, calculating, 206-207

copying, 434 creating, 422-427 data file, 366 deselecting, 430-431 drawing, 427-428 erasing, 427-428 finding, 428 handles, creating, 429-430 location, 225 moving, 431-432 restoring, 435-436 saving, 435 scaling, 432-433 selecting, 428 arc PostScript operator, 484 arc_bounding_handles function. 430 arc_copy function, 434 arc_draw method, 427 arc_erase method, 427 arc find function, 219-221 arc_find method, 428 arc_move function, 432 arc save function, 435 arc_scale function, 243-244, 432-433 arc_select method, 428-429 arcs apply_delta_<direction> function, 247 apply_delta function, 245 aspect ratio, 220 Graphics Editor, moving, 231-232 intersections, calculating, 218-223 rotating, 252 Arg data type, 144-145 argc argument, 86

args, 92. See also arguments argv argument, 86 argument count, 86 argument lists, 146 argument value, 86 arguments. See also parameters AllPlanes, 499 argc, 86 argv, 86 casting, 93 getopt function, 292 required, 92 separating, 93 **XtVaGetApplicationReso** urces function, 295 XYPixmap, 499 ZPixmap, 499 argv argument, 86 armed widgets, 142 array elements, referencing, 172 array of points, 211-212 arrays, 60, 85, 95, 384 determining size, 95 elements, 60 gxDrawIcons, 380, 412 gxDrawIcons array, 422 GXFontP, 311 length, 60, 311 mask bits, 329 objForm, 375 plain simplex, 445-447 plain simplex p, 445-447 point, 381 points adding, 383, 388 managing memory, 387 rubber_line->pts, 383

XPoint, 307 **XtVaGetApplicationReso** urces function, 292, 294 Arrow creation icon, 415 Arrow objects, creating, 415-419 arrow_from_box function, 417-418 arrays (gxDrawIcons), declaring, 168-169 ASCII (American Standard **Code for Information** Interchange), 54-55 ASCII file format, 359 ASCII files, writing, 359-361 aspect ratio (arcs), 220 assigning actions, 336 ClipMask, 330 copy method, 320 data types, 70-71 variables, 322 assigning actions, 169 assignments, combining, 98 asterisk (*), 70 asterisk slash (*/), 84 asterisk wildcard (*), 13 asynchronous, 301 asynchronous handling of events (X Server), 301 Athena widget set, 128-130 ApplicationShell widget, 133-135 Box widget, 135-136 Command widget, 139-140 Core widget, 130-133 Form widget, 136-138 Label widget, 138-139

attributes, 509 automatic variables, 78-79 axes, major/minor, 220

B

back-ups, 13 backslash (\), 33 base names, 35 bash (Bourne Again shell), 12, 525, 531 bash command, 12 begin PostScript operator, **484** bg command, 524 binary file fomat, 357-359 binary numbering systems, 19 bind PostScript operator, 484 bit-mapped data, 167 bitmap clients (colors), 164 bits, 71-73 bodies, 36 conditional, creating, 41 ifeg, 42 body markers ({ }), 77 Boolean, 132 bounds, 203-209 bounds method, 317 Bourne Again shell (bash), 12, 525, 531 Bourne shell (sh). See sh (Bourne shell) Box objects converting to Line object, 415 creating, 411-414

Box widget, 135-136 comparing to XmRowColumn widget, 141 resizing, 160 Box widget resources, 136 brackets ([]), 7 **BSD UNIX, 8** bubble sort, 50 buffers, 94, 374 bugs, 95 built-in functions, 90-91 **Bus Errors**, 27 button widgets, placing, 127 ButtonPress event, 116, 119-121, 208-209, 331, 334, 381-382, 516 buttonpress_update function, 335-336 ButtonRelease event, 116, 334, 338-339, 381-383 buttonrelease_update function, 338-339 buttons, 127 create_buttons function, 159-162, 283-284 creating, 159-162 X-based applications, creating, 159-162 bytes, ordering, 21

C

C compiler (cc) command, 25 C Compiler (cc), 18-19 C compilers, 25, 86 -c flag, 26 C programming language, 25, 69, 320 comment tokens, 84 error reporting, 368 fgets function, 374 getopt function, 290 arguments, 292 code example, 291-292 syntax, 290 variables, 290-291 history of, 83-84 PostScript, translating into, 480-481 standardization of, 25 string functions, 99-100 string library, 97 variables, declaring, 69-70 C shell (csh). See csh (C shell) C++ programming language, 320 calc apply scale function, 470-471 calculating arc intersections, 218-223 bounds (objects), 203-207 deltas, 425 foci, 221 line intersections, 212-216 point intersections, 211-218 slope, 216-218 calculations of slope, comparing, 410 Callback function, 132 callbacks, 300-303 calling create_line function, 386 procedures, 302

calloc function, 106 canvases (Graphics Editor), creating, 154-158 capabilities, print (Graphics Editor), 495 capability of printer, determining, 491-494 Cartesian coordinate system, 254 case statements, 58-60, 534 case tables, 534 case-sensitivity (Linux), 18 casts (float keyword), 217 casting arguments, 93 between data types, 93-94 variables, 217 cc (C Compiler), 18-19 cc (C compiler) command, 25 cd (change directory) command, 18 changing file format, 356-357 char (character) data type, 71-74 character, 310 character bitmap data (grid_bits), 349 character conditions, 54 character data type (char), 71-74 character definitions, 309-313 character encoding, 357 characters % (percent), inserting comments, 482 strokes, 309

charpath PostScript operator. 485 child windows, 122 chmod (change mode) command, 17-18 circles. 221 **CISC** (Complex **Instruction Set** Computer), 21 classes (widgets), 129 clauses, else, 320 client data, 158 client/server model. 114 clients, 115 bitmap (colors), 164 X, 116-119 clip PostScript operator, 484 ClipMask, assigning, 330 ClipOrigin, specifying, 330 clipping graphics pipeline, 198 windows, 122 closepath PostScript operator, 485 close_dialog function, 371 CLUT (color look up table), 198 cntrl manager function, 283-284 code machine, 19 object, 18 op, 19 planning application structure, 262-263 preventing multiple inclusions of same file, 269 source, 18-19 code bodies, 87-88 defining, 87

main function, 86-88 marking, 87 code listings activate_objs function, 333 apply_delta function, 238-239, 245-246, 469 apply_delta_<direction> function. 239-243 apply_scale function, 455-456, 471-472 arc object bounds calculation, 206 arc objects deselecting, 430 drawing/erasing, 427-428 selecting, 428 arc_copy function, 434 arc find function, 219-221 arc_move function, 231-232, 431-432 arc_save function, 435 arc scale function, 243-244, 432-433 arcs creating handles, 429-430 deselecting, 430 Arg and XtSetArg, using, 145 bounds structure, 203 Box objects, converting to Line objects, 414-415 buttonpress_update function, 335-336 buttonrelease_update function, 338-339 calc apply scale function, 470-471

changes necessary to the function Main in gxMain.c for invoking the function initializeGX. 354 changes to be made to the create canvas routine in gxGraphics.c, 348 .cshrc file, 530 common object data structure, 315-316 copy function source code, 319-320 copy_gxtext function, 460-461 create_arc function, 426 create_buttons function, 159-160 create_gxtext, 447-448 create_icons function definition, 169-170 create_line function, 386 create_pixmap function, 171-172 create_text function, 459 creating arrow from box, 417-418 CursorMode enumeration for inclusion in the gxGraphics.h header file, 351 deactivate_objs function, 333 delete function, 321-322 determining if new vertex should be stored, 409 doPrintTo function, 505 double quote character definition, 447 draw_erase function, 462 draw_manager and drawAreaEventProc functions, 173

drawAreaEventProc function. 334 exclamation point character definition, 446-447 filenames, retrieving from user input, 369-370 find_graphic function, 332 find_handle function, 340 freeGXText function. 450-451 get_bounds function, 458 get_creation_text function, 441-442 get_extents function, 457 getopt function, 291 getWidgetPathCompone nts function, 516-517 Global Cursor References for Inclusion in the gxGraphics.c Source File, 351 GNUmakefile, 542 graphic line object structure definition, 306 graphic text object structure definition, 308 Graphics Editor gxArc.c file, 288-289 gxGraphics.c file, 271-275 gxGraphics.h file, 268 gxGx.c file, 285-287 gxIcons.h file, 276-282 gxMain.c file, 265-266 gxProtos.h file, 269-270 gxText.c file, 289 Graphics Editor canvas, creating, 154-155 Graphics Editor data file format, 365

Graphics Editor project (GNUmakefile), 31-32 gray-scaling color raster data, 492-493 grid.xbm file, contents, 349 gx_add_obj Function For Inclusion In gxGx.c, 346 gx_allocate_gc Function For Inclusion In gxGx.c, 347-348 gx_arc function, 422-423 gx_arrow function, 416-417 gx_box function, 412-413 gx_create_obj Function For Inclusion in gxGx.c, 343-344 gx_draw_handles function, 327-328 gx_erase_handles function, 331 gx_line function, 380-381 gx_load function, 373 gx_pencil function, 406-407 gx_print function, 496 gx_save function, 368 gx_text function, 438 GXArc data type definition, 315 gxArc.c file, 549 gxArcLoad function, 435 gxBitmaps.h file, 552-555 gxDrawIcons array declaration, 168-169 GxDrawText function, 443 gxGraphics.c file, 544-547 gxGraphics.h file, 551 gxGx.c file, 548-549

gxIcons.h, 551-552 gxLine.c file, 288, 549-550 gxMain.c file, 543 gxProtos.h file, 555 GXRubberLine function, 384 GxStatusBar. creating/managing, 174-175 gxText.c, 550 GXTextLoad function, 474 hard coding resources, 152 help configuration file entry (sample), 518 HelpAppMainLoop function, 514 icon definitions, 165-168 image color use, determining, 500-501 initializeGX Function for the gxGraphics.c File, 352 line object bounds calculation. 204-205 Line objects copying, 400-401 creating handles, 392-393 deselecting, 394 moving, 395-396 scaling, 398-399 selecting, 392 line-separated tagged file format, 360 line_find function, 390 line_move function, 226-231 line scale function, 235-238

lines drawing/erasing, 388-389 restoring from files, 402-403 saving to files, 402 make.defines file, 38-39, 540-541 menu.sh. 536-537 motionnotify_update function, 340 near_segment function, 212-216 next_char min function, 454 null_func Routine For Inclusion In gxGx.c, 345 objects loading into Editor, 374 parsing, 326 objects selection, testing, 208-209 PCX file header format, 358 place_creation_text function, 439-440 points/segments, determining if selected, 391 point_equal_point function, 384-385 position-specific ASCII data file entries, reading/writing, 361 PostScript comments, 489-490 PostScript Prolog, 499 process_event function, 334-335 processKeyEvent function, 515 .profile files, 527-528

refresh function source code. 325 reset_font_pts function, 451-452 reset_pts function, 451 rotate_point function, 249-251 script_simplex.h font definition, 557-558 set cursor function to be added to gxGraphics.c, 350 temporary box points, updating, 413 text bounding handles, 464, 466 text_copy function, 472-473 text_deselect function, 466-467 text_find function, 463-464 text_move function, 467-468 text_save function, 473 text_scale function, 470 text_select function, 464 token-separated file format, 360 traversing editor objects for saving, 372 triplex_bold.h font definition, 558-559 triplex_bold_italic.h font definition, 559-560 update_arc function, 424-425 update_gxtext function, 444 update_line function, 387 update_obj function, 335

update_pencil function, 407 vector fonts (character definition for letter A), 309 vector_chars.h header file, 561-740 vfonts/simplex.h header file, 445-446 write_ps function, 502 write_ps_data function, 504 XGCValuesStructure, 181 XImage, creating, 497-498 XtVaAppInitialize function, 146 XtVaCreateManageWidg et function, 152 **XtVaGetApplicationReso** urces function, 294-295 codes, op (Operational), 20 colon (:), 24, 35 color extensions (PostScript), 492-494 color images, converting to gray-scaled images, 492-494 color look up table (CLUT), 198 colorimage function, 493-494 colormaps, 133, 184-185, 500 active, 185 installing, 185 parsing, 500-502 colors allocating, 184-185 bitmap clients, 164 Graphics Editor, 509-510

named, 186	bg, 524	man -k, 7
names, 186	cc (C compiler), 25	man man, 6
specifying, 184	cd (change directory), 18	mkdir, 10-12
XAllocNameColor func-	chmod (change mode),	more, 45
tion, 185-187	17-18	moveto, 481
colortogray function, 494	ср (сору), 17, 43	mv (move), 17
combining	csh, 11	passwd, 524
assignments, 98	echo, 36	perror, 368
declarations, 98	env, 40	printenv, 526
file format strategies,	exit, 538	ps (process status), 524
364-366	export, 526	read, 538
command completion,	expr, 535	rm, 17
specifying names for, 12	external, 11	rm -r, 12
command lines, parsing	fg, 524	rmdir, 12-13
getopt function, 290-292	find, 46-47	scalefont, 481
XtVaGetApplicationReso	findfont, 481	setenv, 526
urces function, 292-295	gcc, 25	setfont, 481
command mode, 23	дсс -с, 25-27	sh, 11
commands, 23-24	gcc -D, 28	shell history, repeating,
moving to edit mode, 23	gcc -I, 28-29	529
command shell environ-	gcc -l, 29-30	sleep, 538
ments, 525	gcc -0, 27	startx, 115
command shells, 11,	gcc -q, 27	stat, 505
523-524. See also shells	gcc -W, 27-28	tcsh, 11
Bourne Again shell (bash),	gmake, 43	UNIX (aliases), 526-527
525 D 1 11 (1) 522	grep, 44-45	vi, 84
Bourne shell (sh), 523	groups, 16	xcalc, 117
C shell (csh), 524-525	history, 529	xclock, 117
controlling jobs, 524	id, 16	xconsole, 117
Korn shell (ksh), $524-525$	if, 532	xdm, 115, 117
I C shell (tcsh), 524	include, 37	xdpyinfo, 117
Command widget,	internal, 11, 526	xedit, 117
139-140, 302	jobs, 524	xev, 117
Comparing to YmPushButton widget	ksh, 11	xeyes, 117
142	link (ln), 44	xfontsel, 117
resources 139-140	ln, 44	xinit, 115
commands	lp, 505	xload, 117
alias 527	ls, 14, 527	xman, 117
anronos 7	ls -al, 14	xmessage, 117
hash. 12	man, 6-7	xset, 117
Subiry 12		xterm, 117

748

comment tokens, 84 comments, 32, 482, 489-490 common object data structure, 315-318 **Common Object** Structure, modifying, 367 common-object save and restore, 367-373 communicating events, 301 comparing gx_arrow function to gx_box function, 417 slope calculations, 410 strings, 97-98 compile process, 22 compiler directives, 107 define, 108 include, 91-92 compilers (C), 25, 86 compiling source file, 22 complex shape parameters, 193 **Complex Instruction Set** Computer (CISC), 21 components buttons, 127 X-based applications, 143 computer processor instruction sets. 20-21 conditional bodies, creating, 41 conditions, 52 characters, 54 else, 534 evaluating, 52 flipped, 414 numeric, 53-54 configuration files (make), 32

configuring widgets, 147-148 connecting to X server X-based applications, 144-151 XtAppInitialize function, 144 XtVaAppInitialize function, 144-151 Constraint widget resources, 137 contents (storage locations), 79 context-sensitive help, 513-514 controlling jobs (command shells), 524 conversion routines (line_from_box), 413 converting color images to gray-scaled images, 492-494 convex shape parameters, 193 coordinate systems, 252-253 Drawables, 253 negative numbers, 254 polar, 254-256 rectangular, 254 coordinates (discrete), 253 copy action, 317 copy (cp) command, 17, 43 Copy icon, 319 copy method, 320 copy_gxtext function, 460-461 copying data (memcpy function), 237 Line objects, 400-401

objects, 316, 319-320, 434 Text objects, 472-473 Core widget, 130-133, 141 cp (copy) command, 17, 43 create arc function, 426-427 create_buttons function, 159-162, 283-284 create_canvas function, 275-276, 296-297 create canvas routine, 348 create_gxtext function, 444 code example, 447-449 GXText structure, 449-450 create_icons function, 169-170, 276, 283 create_line function, 382, 385-386, 415 create_pixmap function, 171-172 create_status function, 276 create_text function, 459-461 creating application interfaces (X-based applications), 151-158 buttons (X-based applications), 159-162 conditional bodies, 41 dialog widgets, 370 directories, 10-12 functions, 300 Graphic Context, 329, 389 graphic types, 317 GXLine data structures, 386 handles Arc object, 429-430 Line objects, 392-394

750 creating

line objects, 385 linked lists, 318 objects Arc, 422-427 Arrow, 415-419 Box, 411-414 Line, 388 Pencil, 406-410 Pixmaps, 329 PostScript Prolog, 499 symbolic links, 43-44 windows in an application, 123 XImage, 497-499 creating_line function, 410 creation icons Arrow, 415 Pencil, 406 creation modes, aborting, 382 csh (C shell), 11, 524-525 environment files, 530-531 variables, defining, 526 csh command, 11 .cshrc file, 530-531 curly braces ({ }), 47, 59, 61 current directory (.), 14 Cursor variables, 351 cursorfonts.h header file, 349, 353 CursorMode data type (set_cursor function), 350 cursors as state indicators, 349-350 initializeGX function, 351-354 set_cursor function, 349-351

curveto PostScript operator, 485 Cut icon, 320 cutting objects, 316

D

-d (destination) parameter, 506 -D flag, 28 dash (-), 532 data, 69 accessing by reference, 69, 78-81 bit-mapped, 167 client, 158 memcpy function, copying, 237 overflowing, 73 passing to functions, 68 data field, 315-317 data files, 366 data grouping, 21 data integrity (scaling objects), 234 data representation, 19 data structures, 318 common object, 315-318 GXArc, 314 GXLine, 364, 386 GXObj, 367 GXText, 314 stacks, 481 XRectangle rubber_box, 413 data types, 66 Arg, 144-145 assigning, 70-71 casting between, 93-94

character (char), 71-74 CursorMode (set_cursor function), 350 defining, 77-78 double, 76 **FILE**, 96 float, 76 GXFont, 311, 314 GXIconData, 162 GxLine, 307 GxLinePtr, 307 GXText, 308 integer (int), 74-76 long, 76 short, 76 variable, promoting, 93-94 XtArgVal, 144 de-referencing variables, 79 deactivate_objs function, 332-333 debug flag, 27 debugging objects, 27 shell scripts, 538 X Window applications, 301 decimal numbering systems, 19 decision trees, 50-51 declarations, 64 combining, 98 forward, 28 implicit, 90 declaring functions, 64-65 gxDrawIcons arrays, 168-169 pointers, 78 printf function, 91-92

variables, 39-40, 87 C, 69-70 shell scripts, 532 decorations. 118 deduction, 9 def PostScript operator, 483-485 default shells, 524 define directive, 108 defining code bodies, 87 data types, 77-78 event loops (contextsensitive help), 514-516 functions, 65, 68 procedures, 483 spaces, 313 structures, 77 variables (command shell environments), 525-526 definitions, 65, 165-168 delete function, 321-322 deleting objects, 321-325 delimiters (header files), 267 delimiting, 33 deltas, 425 dependencies, 27, 35-36 depth, 133 descriptions command widget resources, 140 Core widget resources, 132 geometric, 199 Label widget resources, 138-139 deselect method, 317, 394, 430 deselecting objects Arc, 430-431

Line, 394-395 Text, 464-467 designator fields, 15 designators (location), 225 destination (-d) parameter, 506 destroying widgets, 371 determining equal points, 385 pixels (graphics pipeline), 198 point selection, 390-391 printer capability, 491-494 size of arrays, 95 values (action fields), 337 dialog widgets, creating, 370 dictionary, 483 dictionary stack (PostScript), 481-483 dimensions, 133 directing output, 504-506 direction (scaling objects), 233 directives, 28, 107 compiler, 107 define, 108 include, 91-92 HNDL SIZE, 329 directories, 10 creating, 10-12 current (.), 14 include, 550 object (i86-Linux), 540 parent (..), 14 removing, 12-13 requesting, 14 root (2d-editor), 540 source (src), 540

disabling synchronous behavior, 301 Disarm function, 303 discrete coordinates, 253 displays, 114 div PostScript operator, 485 do while loop, 63-64 dollar sign (\$), 33 doPrintTo function, 505 double data type, 76 double dot (..), 11 double greater than symbol (>>), 46 double quotation marks (""), 93, 532 double quote character definition example, 447 double semicolon (;;), 534 draw method, 317, 325, 388 draw_erase function, 388-389, 462-463 draw_manager function, 173, 283-284, 296-297 draw_mgr_func function pointer, 334 draw_mgr_func global variables, 345 drawables functions, 182 pixmaps, 182 systems coordinate, 253 windows, 182 drawAreaEventProc event handler. 334 drawAreaEventProc function, 173 drawing handles, 327, 329-330

objects Arc, 427-428 interactive mode, 228 Line, 388-390 Text, 461-463 vector text objects, 307-309 drawing functions, 334 dup PostScript operator, 485 dynamic memory allocation, 105

E

echo command, 36 edit mode, moving to from command mode, 23 editors, 22 GNU Emacs, 22 vi, 22-24 command mode, 23 command mode commands, 23-24 edit mode, 23 last-line mode, 23-24 last-line mode commands, 24 moving from command mode to edit mode, 23 starting, 23 elements, 60, 172 ellipses (...), 63 foci, 222 rotating, 251 else clause, 320 else conditions, 534 else statement, 52 employing bounds

(objects), 208-209 empty functions, 367 enabling synchronous behavior. 301 enclosing filenames, 92 end of body markers (endif), 41 end PostScript operator, 485 end-of-line marker, 33 endif end of body marker, 41 endless loop, 62 Enlightenment window manager, 118 EnterNotify event, 116 entry points, 299-300 env command, 40 environment, 40 environment files, 527 Bourne shell, 527-529 C shell, 530-531 environment variables, 40 equal points, 382, 385 equal sign (=), 526 equal to symbol (= =), 53equations, linear, 217 erase method, 317, 322, 388 erasing handles, 330-331 objects Arc, 427-428 Line, 388-390 Text objects, 461-463 errno global variable, 369 errno variable, 369 error reporting (C), 368 error.h header file, 369 errors, 301 Bus, 27 logic, 28

minimizing (floatingpoint calculation), 399 rounding (scaling objects), 234 semantic, 28, 65 syntax, 28, 65 evaluating conditions, 52-53 strings, 56 event handlers, 300-303, 334 event hooks, 302 event loops (context-sensitive help), defining, 514-516 event masks (X), 157-158 event references (validity), testing, 381 event-driven programming, 302 event-loops, 300 events, 115-116, 299-301 asynchronous handling by X Server, 301 ButtonPress, 116, 119-121, 208-209, 331, 334, 381-382, 516 ButtonRelease, 116, 334, 338-339, 381-383 communicating, 301 EnterNotify, 116 Expose, 116, 122-124 generating, 301-302 keyboard, 303 KeyPress, 116, 516 LeaveNotify, 116, 119 MotionNotify, 339, 381-383 non-null, 230 nonmaskable, 158 NULL, 382

files

PointerMotion, 116, 334 windows, 127 X-based applications, 176 exch PostScript operator, 485 exclamation points (character definition example), 446-447 execute permission, 15-17 executing files, 17 exit command, 538 ExntX global variable, 399 ExntY global variable, 399 explicit path, 11 explicit tests (non-null events), 230 explicitly managing memory, 100-101 export command, 526 Expose event, 116, 122-124 expr command, 535 extensions .h, 28 .o. 20 extern variables, 104-105, 369 external commands, 11 external dependency, 27

F

-f flag, 37 families (UNIX), 8 fg command, 524 fgets function, 374 fi keyword, 532 field summary (XGCValues), 188-189

fields, 357

action, 317, 337 angle1, 511 angle2, 511 data, 315-317 line_width, 510 next, 318, 323 next field, 345 num_pts, 388 selected, 316 vpts, 314 x, 314 v, 314 file formats ASCII. 359 binary, 357-359 position-specific, 360-361 strategy, 364 tagged, 359-360 FILE data type, 96 file format strategies, 363-366 file pointers, 95 file systems (Linux), navigating, 18 filename extensions, 357-358 filenames enclosing, 92 string integrity, 371 files, 357 ASCII, writing, 359-361 configuration, 32 .cshrc, 530-531 data, 366 directing output to, 504-506 environment, 527 Bourne shell, 527-529 C shell, 530-531

executing, 17 formatting, 355-357 generating, 26 grid.xbm, 349 gxArc.c, 549 gxBitmaps.h, 552 gxGraphics.c, 544 gxGraphics.h, 550 gxGx.c, 547 gxIcon.h, 551 gxLine.c, 549 gxMain.c, 542 gxProtos.h, 555 gxText.c, 550 header, 28, 91-92 cursorfonts.h, 349, 353 delimiters, 267 error.b. 369 Graphics Editor; 266-267 gxGraphics.b, 268-269, 329, 349 gxIcons.b , 276-284 gxProtos.b, 268-271 locating, 28-29 path components, 267 vector_chars.h, 446, 557, 560 X11/Xaw/Dialog.h, 370 X11/Xaw/StringDefs.h, 370 hidden, 14, 527 including, 37 make, 37. See also GNUmakefile, Makefile make.defines, 37, 40-41, 540-541 object, 19-20 generating, 35 platform specific, 20-21 storing, 41-43

753

PostScript ps suffix, 479 viewing, 488-489 PostScript page definition, writing, 502-504 preventing multiple inclusions of same file, 269 .profile, 527-529 source, 21-22 compiling, 22 invoking functions, 27 storing, 41-43 fill PostScript operator, 485 find command, 46-47 find method, 317, 390 find_graphic function, 332, 335 find_handle function, 337, 340 find_handle routine, 337, 340 findfont command, 481 findfont PostScript operator, 485 finding handles, 340-341 objects Arc, 428 Line, 390-392 Text objects, 463-464 fix_x variable, 414 fix_y variable, 414 FixedX variable, 399 FixedY variable, 399 flags, 6 -a, 14, 527 -c, 26 -D, 28 -f, 37

-g, 16 -I, 28, 44, 92 -k. 7 -l, 14, 28-30, 527 -n, 44 -o, 27 -q, 27 -r, 12 -s, 524 -v, 45 -W, 25-27 -x, 538 found, 341 multiple, specifying, 14 tile, 388-389, 427 flipped conditions, 414 flipping objects, 414 float data type, 76 float keyword (cast), 217 floating-point calculations, minimizing errors, 399 foci calculating, 221 defined, 221 ellipses, 222 font sets, 311-312 fonts hershey, 309 PostScript, specifying, 480 scalable, 309 vector. 309-310 character definitions, 309-313 vector chars.h header file, 446 fopen function, 95-97 for loop, 60-62, 535-536 for PostScript operator, 486

Form widget, 136-138, 155 comparing to XmForm widget, 141 resources, 136 format string, 92 formatted lines, positionspecific, 365 formatted output, 94 formatting files, 355-357 text. 22 formatting tokens, 92-93 forward declarations, 28 found flag, 341 fprintf function, 94 frames, 79 free function, 106 freeGXText function, 450-451 freeing memory, 106 nested pointers, 243 function calls, 536-537 function pointers (draw_mgr_func), 334 function types (GC), 182-183 functionality (widgets), 130 functions, 64 abs, 214 Activate, 303 activate_objs, 333 add2Nums, 89 apply_delta, 239, 245, 399, 469 apply_delta_<direction>, 241-243, 247 apply_delta_bottom_side, 240-242 apply_delta_left_side, 240-242

apply_delta_right_side, 240-242 apply_delta_top_side, 240-242 apply_scale, 455-457, 471-472 arc_bounding_handles, 430 arc_copy, 434 arc_find, 219-221 arc_move, 432 arc_save, 435 arc_scale, 243-244, 432-433 arrow_from_box, 417-418 built-in, 90-91 buttonpress_update, 335-336 buttonrelease_update, 338-339 calc_apply_scale, 470-471 Callback, 132 calloc, 106 close_dialog, 371 cntrl_manager, 283-284 colorimage, 493-494 colortogray, 494 copy_gxtext, 460-461 create_arc, 426-427 create_buttons, 159-162, 283-284 create_canvas, 275-276, 296-297 create_gxtext, 444 code example, 447-449 GXText structure. 449-450 create_icons, 169-170, 276, 283 create_line, 382, 385-386, 410, 415

create_pixmap, 171-172 create_status, 276 create_text, 459-461 creating, 300 deactivate objs, 332-333 declaring, 64-65 defining, 65, 68 delete, 321-322 Disarm. 303 doPrintTo, 505 draw erase, 388-389, 462-463 draw_manager, 173, 283-284, 296-297 drawables, 182 drawAreaEventProc, 173 empty, 367 fgets, 374 find_graphic function, 332, 335 find_handle, 337, 340 fopen, 95-97 fprintf, 94 free, 106 freeGXText, 450-451 GC (GXxor), 228 get_bounds, 394, 458-459 get_creation_text, 439-443 get_extents, 457-458 get_image, 497-499 getopt, 290 arguments, 292 code example, 291-292 syntax, 290 variables, 290-291 getWidgetPath, 516-518 getWidgetPathComponents, 516-518 graphic primitives, 189 gx_add_obj, 346

gx_allocate_gc, 347-348, 389 gx_arc, 334, 423 gx_arrow, 334, 417 gx_box, 334, 412-413 gx_copy, 320-322 gx_create_obj, 343-345, 367 gx_draw handles, 327 gx_line, 334, 380-383 gx_load, 367, 373 gx_new_vertex, 408 gx_pencil, 334, 406-407 gx_print, 496-497 gx_refresh, 320, 325-326 gx_save, 367-368 gx_text, 334, 438-439 gxArcLoad, 436 GXDrawText, 313, 443-444 gxGetFileName, 368-370 GxIcon, 169 gxLineLoad, 402-403 gxLoadCommon, 375 gxLoadObjs, 374 GXRubberLine, 382-384 gxSaveCommon, 373 gxSaveObj, 368 gxSaveObjs, 372 GXTextLoad function, 474-475 HelpAppMainLoop, 514-515 initializeGX, 351-354 invoking, 27, 300 line_copy, 400-401 line_find, 390 line_from_box, 415 line_move, 396 line_save, 402 line_scale, 242

load, 361 main, 69, 85-86 code body, 86-88 parameter list, 85 parameters, 86 return type, 85 malloc, 105-106 memcpy, 237 memory allocation, 106-107 motionnotify_update, 339-340 names, 66-67 naming, 67 near_segment, 212-216, 391 next_char_min, 454 parameters, 64-68. See also arguments parse_all_objects, 326 passing data to, 68 perror, 368 place_creation_text, 439-441 point_equal_event, 382 point_equal_point, 384-385 point_selected, 390-391 printf, 91, 146 declaring, 91-92 formatted output, 94 formatting tokens, 92-93 type checking, 93 processKeyEvent, 515 process_event, 334-335, 396 prototyping, 65-66 realloc, 106 recursive, 327, 372 reset_font_pts, 451-453

reset_pts, 451 restore, 361 restoring, 366 return types, 66 saving, 366 segment_selected, 390-391 set_color_data, 500-501 set cursor, 349-351 sprintf (formatted output), 94 strcat, 99 strchr, 100 strcmp, 97-99 strcpy, 99 strdup, 100, 107 strerror, 369 string, 99-100 strlen, 100 strncat, 99 strncmp, 99 strncpy, 99 strstr. 100 stub, 367 text_bounding_handles, 464-466 text_copy, 472-473 text_deselect, 466-467 text_find, 463-464 text_move, 467-468 text_save, 473-474 text_scale, 470 text_select, 464 trackingEvent, 516 type checking, 65 update_arc, 425 update_box, 413-414 update_gxtext, 444 update_line, 387 update_obj, 335

update_pencil, 407-408 write_ps, 502-503 write_ps_color_conv, 492-494.503 write_ps_data, 503 write_ps_prolog, 499 XAllocNameColor, allocating color, 185-187 XCGValues line width field, 187 tile field, 188 values, 182-184 XDefineCursor, 351 XDrawArc, 192-193 XDrawLine, 191, 212, 410 XDrawPoint, 190-191 XDrawRectangle, 191 XFill, 193 XGCValues, 188-189 XGetImage, 498 Xlib, 180-184, 301 Xt, 180 XtAddCallback, 161 XtAddEventHandler, 156, 276 XtAppInitialize, 144 XtAppMainLoop, 176, 285, 299 XtAppNextEvent, 300 XtAugmentTranslations, 304 XtDispatchEvent, 300 XtMalloc, 408 XtNew, 236 XtParseTranslationTable, 304 XtRealloc, 408 XtVaAppInitialize, 144-151

XtVaCreateManageWidg et, 151-158 XtVaGetApplicationReso urces, 290-292 arguments, 295 arrays, 292-294 code example, 294-295 parameters, 292-294 syntax, 292

G

-g flag, 16 garbage, 78-80 GC (Graphics Context), 179, 316 creating, 329, 347-348, 389 function types, 182-183 GXxor function, 228 rubberGC global, 353 source pixels value, 198 tile values, 462 tiling, 348-349 Xlib function, 180-184 Xt function, 180 gcc (GNU C Compiler), 20 gcc -c command, 25-27 gcc -D command, 28 gcc -I command, 28-29 gcc -l command, 29-30 gcc -o command, 27 gcc -q command, 27 gcc -W command, 27-28 gcc command, 25 general linear equations, 217 generating events, 301-302 object files, 26, 35

geometric descriptions, 199 get_bounds function, 394, 458-459 get_creation_text function, 439-443 get_extents function, 457-458 get_image function, 497-499 getopt function, 290 arguments, 292 code example, 291-292 syntax, 290 variables, 290-291 getWidgetPath function, 516-518 getWidgetPathComponent s function, 516-518 GhostScript PostScript files, viewing, 488-489 prologs, 489-490 Web site, 489 **GLOBAL** section (gxGraphics.h), 353 GLOBAL variable, 269 global variables, 88, 104 draw_mgr_func, 345 errno, 369 ExntX, 399 ExntY, 399 OrigX, 399 OrigY, 399 updating, 337 Xdebug, 301 globals (rubberGC), 353 gmake command, 43 gmake utility, 31-33 GNU C Compiler (gcc), 20 **GNU Emacs editor**, 22

GNUmakefile, 31, 541-542 comments, 32 include command, 37 targets, 33-35 body, 36 dependencies, 35-36 variables, 32-33 Graphics Context. See GC graphic primitives, 179 XFillRectangle, 330 Xlib XDFill function, 193 XDrawArc function, 192-193 XDrawLine function, 191 XDrawPoint function, 190-191 XDrawPoints function, 190 XDrawRectangle function, 191 graphic primatives functions, 189 graphic types, creating, 317 graphical user interfaces. See user interfaces graphics raster, 199-200 vector, 199 X Window System, 197 **Graphics Editor** arc angles, 511 arcs, moving, 231-232 attributes, 509 cntrl manager function, 283-284 color, 509-510 create_buttons function, 283

create_canvas function, 275-276, 296-297 create_icons function, 276, 283 create_status function, 276 cursors as state indicators, 349-354 draw_manager function, 283-284, 296-297 features list, 263-264 Graphic Context creating, 347-349 gx_allocate_gc function, 347-348 tiling, 348-349 line attributes, 510-511 moving graphics, 225-226 lines, 226-231 objects creating, 343-347 linked lists, 346-347 print capability, 495 project structure, 244, 262-263 user interface gxArc.c file, 288-289 gxGraphics.c file, 271-276 gxGraphics.h header file, 268-269 gxGx.c file, 285-288 gxIcons.h header file, 276-284 gxMain.c file, 265-271, 284-285 gxProtos.b header file, 268-271 gxText.c file, 289 beader files, 266-267

XtAddEventHandler function, 276 XtAppMainLoop function, 285 Graphics Editor application, 305 Graphics Editor canvas, creating, 154-158 graphics pipeline, 197-198 gray-scaled images, converting color images to, 492-494 greater than or equal to symbol (>), 53 greater than symbol (>=), 45, 53 grep command, 44-45 grestore PostScript operator, 486 grid_bits character bitmap data, 349 grid.xbm file, 349 group access, 16 group access fields, 16 group permission, 16 groups command, 16 gsave PostScript operator, **486** gx_add_obj function, 346 gx_allocate_gc function, 347-348, 389 gx_allocate_gc utility, 329 gx_arc function, 334, 423 gx_arrow function, 334, 417 gx_box function, 334, 412-413, 417 gx_copy function, 320-322 gx_create_obj function, 343-345, 367 gx_delete routine, 322

gx_draw_handles function, 327 gx_line action, 382 gx_line function, 334, 380-383 gx_load function, 367, 373 gx_new_vertext function, 408 gx_pencil function, 334, 406-407 gx_print function, 496-497 gx_refresh function, 320, 325-326 gx_save function, 367-368 gx_text function, 334, 438-439 GxActiveHandle variable, 337 GXArc data structure, 314 gxArc.c file, 549 gxArcLoad function, 436 gxArcLoad method, 375 gxBitmaps.h file, 552 gxDrawIcons array, 168-169, 380, 412, 422 GXDrawText function, 313, 443-444 GXFont data type, 311, 314 GXFontP array, 311 gxGetFileName function, 368-370 gxGraphics file, 544 gxGraphics.h file, 353, 550 gxGraphics.h header file, 329, 349 gxGx.c file, 547 gxGx.c source file, 367 **GxHOME** variable, 41 **GxIcon function**, 169 GxIconData data type, 162 gxIcons.h file, 551 GXLine data structure, 364, 386 GxLine data type, 307 GXLine structure, 226, 306, 401 gxLine.c file, 549 gxLineLoad function, 402-403 gxLineLoad method, 375 GxLinePtr data type, 307 gxLoadCommon function, 375 gxLoadObjs function, 374 gxMain.c file, 542 GXObj data structure, 367 gxObjHeader value, 324 gxObjHeader variable, 322, 346 gxObjs value, 324 gxObjs variable, 324 gxProtos.h file, 555 **GXRubberLine function**, 382-384 gxSaveCommon function, 373 gxSaveObjs function, 368, 372 gxSaveObjs routine, 372 GxSRCDIR variable, 41-42 GxStatusBar, 174-175 GxStatusBar variable, 161 GXText data structure, 314 GXText data type, 308 gxText.c file, 550 GXTextLoad function, 474-475 gxTextLoad method, 375 GXxor (GC function), 228

Η

.h extension, 28 handle bit data array, 329 handles creating Arc object, 429-430 Line objects, 392-394 drawing, 327-330 erasing, 330-331 finding, 340-341 objects, scaling, 233 selecting, 337 Text objects, 464-466 handling events asynchronously, 301 hard-coded resources, 152 header files, 28, 91-92 cursorfonts.h, 349, 353 delimiters, 267 error.h, 369 Graphics Editor, 266-267 gxGraphics.h, 268-269 gxIcons.b, 276-284 gxProtos.b, 268-271 gxGraphic.h, 349 gxGraphics.h, 329 locating, 28-29 path components, 267 vector_chars.h, 446, 557, 560 X11/Xaw/Dialog.h, 370 X11/Xaw/StringDefs.h, 370 heads (linked lists), 318 heap, 105-106 help (context-sensitive), 513-514 HelpAppMainLoop function, 514-515

hershey fonts, 309. See also vector fonts hexadecimal numbering systems, 19 hidden files, 14, 527 hierarchy (windows), 121-122 history C. 83-84 X Window System, 113-114 history command, 529 HNDL SIZE directive, 329 home directories, referencing, 12 hooks, 300-302 hyphen (-), 6-7

I

-I flag, 28, 44, 92 i86-Linux object directory, 540 icon definitions, 165-168 icons Copy, 319 create icons function, 276, 283 Cut, 320 Latex Line, 380 Load, 373 pixmap, creating, 162-169, 171-172 Save, 367 id command, 16 if command, 532 if else statements, 57-58 if PostScript operator, 486 if statements, 51-52, 532

ifdef keyword, 41 ifeg body, 42 ifelse PostScript operator, 486 ifndef keyword, 41 images color, converting to grayscale, 492-494 raster (modifications), 200 immediate graphic nature of X, 122-123 implicit declarations, 90 implicit test (non-null events), 230 implicitly managing memory, 100-101 in scope, 88 include command, 37 include compiler directive, 91-92 include directory, 550 including files, 37 index PostScript operator, 486 indirect resources, 140 induction, 9 indx variable, 384 initialization level (initlevel), 115 initializeGX function. 351-354 initializing object methods, 367 variables, 78-80, 98 initlevel, 115 input, optional, 7 inserting comments (PostScript), 482 installing colormaps, 185 instance hierarchy (widgets), 133

instantiation (widgets), 130 instruction sets CISC (Complex Instruction Set Computer), 21 RISC (Reduced Instruction Set Computer), 20 int (integer data type), 69, 74-76 integrity data, 234 strings, 371 interactive modes, 228 interfaces applications, creating, 151-158 feature lists, 264 user aesthetics, 264-265 feature lists, 263 Graphics Editor features, 263-264 gxArc.c file (Graphics Editor), 288-289 gxGraphics.c file (Graphics Editor), 271-276 gxGx.c file (Graphics Editor), 285-288 gxMain.c file (Graphics Editor), 265-271, 284-285 gxText.c file (Graphics Editor), 289 intuitiveness, 264-265 planning layout, 264-265 internal commands, 11, 526 intersections, calculating arcs, 218-223

lines, 212-216 points, 211-218 inverse search, 45 invoking actions, 336 functions, 27, 300

J-K

job number, 524 jobs, controlling, 524 jobs command, 524 jump points, 19 -k flag, 7 KActivate translation, 303 Kernighan, Brian, 25 keyboard events, 303 KeyPress event, 116, 516 keywords fi, 532 float, 217 ifdef, 41 ifndef, 41 static, 103-104, 227 this, 320 KHelp translation, 303 ksh (Korn shell), 11, 524-525, 531 ksh command, 11

L

-l flag, 14, 28-30, 527 Label widgets, 138-139 resources, 138-139 XmLable, 142 last in, first out (LIFO) lists, 481 last-line mode, 24 Latex Line icon, 380 layout (user interfaces), 262. See also structure aesthetics, 264-265 feature lists, 263-264 Graphics Editor features, 263-264 gxArc.c file (Graphics Editor), 288-289 gxGraphics.c file (Graphics Editor), 271-276 gxGraphics.h header file (Graphics Editor), 268-269 gxGx.c file (Graphics Editor), 285-288 gxIcons.h header file (Graphics Editor), 276-284 gxMain.c file (Graphics Editor), 265-271, 284-285 gxProtos.h header file (Graphics Editor), 268-271 gxText.c file (Graphics Editor), 289 header files (Graphics Editor), 266-267 intuitiveness, 264-265 planning, 264-265 LeaveNotify event, 116, 119 length arrays, 60, 311 handle bit data array, 329 less than or equal to symbol (<=), 53

less than symbol (<), 53 lexical analysis, 56, 98-99 libraries, 29 locating, 30 names, determining, 29 LIFO (last in, first out) lists, 481 line attributes, 510 Graphics Editor, 510 line style, 511 line width, 510-511 line intersections, calculating, 212-216 Line object, 305 bounds, calculating, 204-207 converting Box objects to, 415 copying, 400-401 creating, 385, 388 deselecting, 394-395 drawing/erasing, 388-390 finding, 390-392 handles, creating, 392-394 moving, 395, 397 point-array base, 305 restoring, 401-403 saving, 401-402 scaling, 398-400 selecting, 392, 395 tagged line, 365 line_copy function, 400-401 line_find function, 390 line_find method, 392 line_from_box conversion routine, 413 line_from_box function, 415 line_move function, 396 line_move method, 395

line_save function, 402 line_save method, 403 line_scale method, 242, 398-399 line_width field, 187, 510 linear equations, 217 LineDashed value, 511 LineDoubleDashed value, 511 lines Graphics Editor, 226-231 multiple, wrapping, 33 objects, scaling, 235-238 rotating, 248-251 slope, 216 tagged, 365 LineSolid value, 511 lineto PostScript operator, 486 link (ln) command, 44 linked lists, 318, 346-347 adding objects, 387 creating, 318 heads, 318 pointers, managing, 323-324 links, 15, 43-44, 542 Linux. 8 case-sensitivity, 18 file system, navigating, 18 make utility, 24-25 X Clients, 117 listing (ls) command, 14, 527 lists arguments, 146 feature lists (user interfaces), 263-264 LIFO, 481

linked, 318, 346-347 adding objects, 387 creating, 318 heads, 318 managing pointers, 323-324 parameter, 67, 85 resource, 144 variable argument, 144 XtCallbacks, 133 little-endian, 21 In (link) command, 44 load function, 361 Load icon. 373 locating header files, 28-29 libraries, 30 locations (objects), 225 location designator, 225 logic errors, 28 long data type, 76 look and feel, 128 loops, 60 do while, 63-64 endless, 62 for, 60-62, 535-536 while, 62-63, 535 lp command, 505 ls (listing) command, 14, 527

M

machine code, 19 macros safe, 146 unsafe, 146 XtSetArg, 145 magic numbers, 361 main function, 69, 85-86 code body, 86-88 parameter list, 85 parameters, 86 return type, 85 major axes, 220 make files, 37. See also **GNUmakefile**, Makefile make utility, 24, 30-31 cc command, 25 Makefile, 31 body, 36 comments, 32 include command, 37 targets, 33-36 variables, 32-33 make.defines file, 37, 40-41, 540-541 Makefile, 31 comments, 32 include command, 37 targets, 33-35 body, 36 dependencies, 35-36 variables, 32-33 malloc function, 105-106 man command, 6-7 man -k command, 7 man man command, 6 managing GxStatusBar, 174-175 memory, 100-103 explicitly, 100-101 implicitly, 100-101 points array, 387 Pencil objects, 410 pointers (linked lists), 323-324 status (objects), 331

widgets, 153, 371 windows, 175-176 manipulating pointers, 70, 80 strings, 100 variable values, 157 manual (man) command, 6-7 mapping windows, 121 margin of error (TOLER-ANCE), 385 markers, end-of-line, 33 marking code bodies, 87 mask bits array, 329 masks, 181 memcpy function, 237 memory allocated, returning, 106 allocating, 103 freeing, 106 heap, 105-106 managing, 100-103 points array, managing, 387 re-allocating, 388 XtNew function, 236 memory addresses, 70. See also pointers memory allocation dynamic, 105 functions, 106-107 memory errors, 103 memory leaks, 106-107 methods, 317 arc_draw, 427 arc_erase, 427 arc_find, 428 arc_select, 428-429 bounds, 317 copy, 320

deselect, 317, 394, 430 draw, 317, 325, 388 erase, 317, 322, 388 find, 317, 390 gxArcLoad, 375 gxLineLoad, 375 gxTextLoad, 375 line find, 392 line_move, 395 line_save, 403 line scale, 398-399 move, 317 scale, 317, 337, 398 select, 317, 333, 392 minimizing errors in floating-point calculations, 399 minor axes, 220 mkdir command, 10-12 models, client/server, 114 modes command commands, 23-24 moving to edit mode, 23 creation, aborting, 382 fopen function, 97 interactive, 228 last-line, 24 modifications (raster images), 200 modifiers, unsigned, 73-76 modifying Common **Object Structure**, 367 modules, 27, 64. See also functions more command, 45 Motif widget sets, 128, 140-142 MotionNotify event, 339, 381-383

motionnotify_update function. 339-340 move action, 396 move (mv) command, 17 move method, 317 moveto command, 481 moveto PostScript operator, 486 moving arcs, 231-232 Line objects, 395-397 lines, 226-231 objects, 316 Arc, 431-432 Graphics Editor; 225-226 Text objects, 467-469 mul PostScript operator, 486 multiple flags, specifying, 14 multiple lines, wrapping, 33 mv (move) command, 17

Ν

-n flag, 44 named colors, 186 names colors, 186 functions, 66-67 libraries, determining, 29 resource (structures), 295 specifying for command completion, 12 naming functions, 67 variables, 32 navigating file systems (Linux), 18 projects, 9-10 near_segment function, 212-216, 391 negative numbers (coordinate system), 254 nested pointers, freeing, 243 newpath PostScript operator, 486 next_char_min function, 454 next field, 318, 323, 345 nodes, 318, 346. See also data structures non-null events, 230 nonconvex shape parameters. 193 nonmaskable events, 158 not equal to symbol (!=), 53 not symbol (!), 54, 63 **NULL**, 96 NULL events, passing, 382 NULL pointers, 318 null termination, 95 null func routine, 345 num handles test, 330 numbering systems, 19 numeric conditions evaluating, 53 symbols, 53-54 num pts field, 388

0

.o extension, 20 -o flag, 27

object code, 18 object directories (i86-Linux), 540 object files, 19-20 generating, 26, 35 platform specific, 20-21 storing, 41-43 object methods, 367 object-specific save and restore, 373-375 objects Graphics Editor, 203 actions, assigning to, 336 active, 316 adding to linked lists, 387 arc, 314-315, 363, 511 Arc copying, 434 creating, 422-427 data file, 366 deselecting, 430-431 drawing, 427-428 erasing, 427-428 finding, 428 handles, creating, 429-430 moving, 431-432 restoring, 435-436 saving, 435 scaling, 432-433 selecting, 428 arcs location, 225 moving, 231-232 Arrow, 415-419 bounds 2D Graphical Editor; 204 calculating, 203-207 employing, 208-209

Box, 411 converting to Line object, 415 creating, 411-414 copying, 316, 319-320 creating, 343-347 gx_add_obj function, 346 gx_create_obj function, 343-345 linked lists, 346-347 null_func routine, 345 cutting, 316 debugging, 27 deleting, 321-325 flipping, 414 Graphics Editor, 225-226 handles drawing, 327-330 erasing, 330-331 scaling, 233 interactive mode drawing, 228 Line, 305 copying, 400-401 creating, 385, 388 deselecting, 394-395 drawing/erasing, 388-390 finding, 390-392 handles, creating, 392-394 moving, 395, 397 point-array based, 305 restoring, 401, 403 saving, 401-402 scaling, 398-400 selecting, 392, 395 tagged line, 365 lines, scaling, 235-238 moving, 316

navigation, 334-341 parsing, 326-327 Pencil creating, 406-410 managing, 410 point-array (location), 225 refreshing, 325-326 resizing, 316 rotating, 511 rounded errors, scaling, 234 rubber-banding, 228 saving, 372 scaling, 232-233, 337 data integrity, 234 direction, 233 selecting, 306, 316, 331-333 status, managing, 331 template, 411 Text apply_scale function, 455-457 copy_gxtext function, 460-461 copying, 472-473 create_gxtext function, 444, 447-450 create_text function, 459-461 data file, 366 double quote character definition, 447 drawing, 461-463 erasing, 461-463 exclamation point character definition, 446-447 finding, 463-464 freeGXText function, 450-451

get_bounds function, 458-459 get_creation text function, 439-443 get_extents function, 457-458 gx_text function example, 438-439 GxDrawText function, 443-444 bandles, 464, 466 moving, 467-469 next_char_min function, 454 place creation text function example, 439-441 plain simplex array, 445-447 plain simplex p array, 445-447 reset_font_pts function, 451-453 reset_pts function, 451 restoring, 474-475 saving, 473-474 scaling, 469-472 selecting/deselecting, 464-467 update gxtext function, 444 user navigation, 334-341 vector text, 307 objForm array, 375 objForm buffer, 374 octal numbering systems, 19 offset, 19 op (Operational) codes, 19-20 opaque, 96

operand stack (PostScript), 481 operators, PostScript, 483 add. 484 arc, 484 begin, 484 bind, 484 charpath, 485 clip, 484 closepath, 485 curveto, 485 def, 483 def, 485 div, 485 dup, 485 end, 485 exch, 485 fill, 485 findfont, 485 for, 486 grestore, 486 gsave, 486 if, 486 ifelse, 486 index, 486 lineto, 486 moveto, 486 mul, 486 newpath, 486 pop, 486 restore, 486 rlineto, 487 rmoveto, 487 rotate, 487 save, 487 scale, 487 scalefont, 487 setfont, 487 setgray, 487 setlinewidth, 487

show, 487 showpage, 488 stroke, 488 sub, 488 translate, 488 optional input, 7 or symbol (||), 54 ordering bytes, 21 organizing projects, 9-10 origin (windows), 122 OrigX global variable, 399 OrigY global variable, 399 out of scope, 88 outline fonts, 309. See also vector fonts output directing, 504-506 formatted, 94 redirecting, 45-46 overflowing data, 73 owner access fields, 15 owner access permissions, 15

P

packages, 29 page definition files (PostScript), writing, 502-504 parameter lists, 67, 85 parameters, 6, 65-68. See *also* arguments -d (destination), 506 cd command, 18 fopen function, 97 main function, 86 shape, 193 XEvent, 326

Xt command-line, 148 **XtVaGetApplicationReso** urces function, 292-294 parent directory (..), 14 parent windows, 121-122 parse_all_objects function, 326 parsing Colormap, 500-502 command lines getopt function, 290-292 XtVaGetApplicationReso urces function, 292-295 objects, 326-327 passing data to functions, 68 NULL events, 382 rubber_line variables, 382 passwd command, 524 path variable, 40-42 paths explicit, 11 path components in header files, 267 relative, 11 widgets, 516-518 .pcx filename extension, 358 Pencil creation icon, 406 Pencil object creating, 406-410 managing, 410 Pentium processors, 20 percent sign (%), 93, 482, 524 performing tests, 532 period (.), 14 permissions, 13-15 designator fields, 15 execute, 15-17

group, 16 group access fields, 16 owner access, 15 owner access fields, 15 read, 17 world. 16 world access fields, 16 write, 15-17 perror command, 368 perror function, 368 pipe symbol (1), 45 pixels, 198 pixmap icons, creating, 162-169, 171-172 Pixmaps, 182, 329 place_creation_text function, 439-441 placing button widgets, 127 widgets, 155 plain simplex array, 445-447 plain simplex p array, 445-447 plain text, 21 planning projects, 8 platform specific object files, 20-21 platforms, support, 42 point array, 381 point intersections, calculating, 211-216 point size, specifying, 480 point slope linear equations, 217 point-array-based line objects, 305 point-array objects, 225 point_equal_event function, 382

point_equal_point function. 384-385 point_selected function, 390-391 PointerMotion event, 116, 334 pointers, 69-70, 78 declaring, 78 file. 95 linked lists, managing, 323-324 manipulating, 70, 80 nested, freeing, 243 NULL, 318 returning, 102 stack, 20 variable, 70 points adding to arrays, 383, 388 arrays of, 211-212, 387 determining if selected, 390-391 equal, 382, 385 intersections, calculating, 218 rotating, 248-251 tracking number, 388 polar coordinate system, 254-256 pop PostScript operator, 486 portability of X applications, 126 position-specific file format, 360-361 Position-Specific file format strategies, 364-366 position-specific formatted line, 365 **POSIX-conformant sys**tems, 301

```
PostScript, 479-480
   C, translating, 480-481
   color extensions, 492-494
   comments, 482, 489-490
   files
      GhostScript, viewing,
        488-489
      ps suffix, 479
   operators, 483
      add, 484
      arc, 484
      begin, 484
      bind, 484
      charpath, 485
      clip, 484
      closepath, 485
      curveto, 485
      def, 483-485
      div. 485
      dup, 485
      end, 485
      exch, 485
      fill, 485
      findfont, 485
      for; 486
      grestore, 486
      gsave, 486
      if, 486
      ifelse, 486
      index, 486
      lineto, 486
      moveto, 486
      mul, 486
      newpath, 486
      pop, 486
      restore, 486
      rlineto, 487
      rmoveto, 487
      rotate, 487
      save, 487
```

scale, 487 scalefont, 487 setfont, 487 setgray, 487 setlinewidth, 487 show, 487 showpage, 488 stroke, 488 sub, 488 translate, 488 printing, 480 programming, 482-483 stacks, 481 dictionary stack, 481-483 operand stack, 481 PostScript page definition file, writing, 502-504 PostScript Prolog, creating, 499 pound sign (#), 32, 39, 91 preventing multiple inclusions of same file in code, 269 print capability of **Graphics Editor**, 495 printenv command, 526 printer capability, determining, 491-494 directing output to, 504-506 printf function, 91, 146 declaring, 91-92 formatted output, 94 formatting tokens, 92-93 type checking, 93 printing (PostScript), 480 procedures calling, 302 defining, 483

process status (ps) command, 524 processing events ButtonRelease, 383 MotionNotify, 383 widows, 127 X-based applications, 176 processKeyEvent function, 515 processors, 20 process_event function, 334-335, 396 .profile file, 527-529 program bugs, 95 programming event-driven, 302 PostScript, 482-483 programming construct, 50 programming languages. See individual programming languages **Project Athena**, 113 project structure (Graphics Editor), 244 projects navigating, 9-10 organizing, 8-10 planning, 8 prologs (GhostScript), 489-490 promoting variable data types, 93-94 prototypes, 28, 65 prototyping functions, 65-66 ps (process status) command, 524 ps suffix (PostScript files), 479

Q-R

-q flag, 27 quoting variables, 532 -r flag, 12 radius, rho, 255, 425 range check (near segment), 215 raster, 200 raster graphics, 199-200 re-allocating memory, 388 read command, 538 read permissions, 15-17 realloc function, 106 records, 357 rectangular coordinate system, 254 recursive functions, 327, 372 redirecting output, 45-46 redirection, 45 **Reduced Instruction Set** Computer (RISC), 20 references accessing data by, 69, 78-81 events, testing validity, 381 referencing array elements, 172 home directories, 12 refreshing objects, 325-326 relating widgets to text, 518-519 relationships (architecture), 127 relative path, 11 removing directories, 12-13

repeating commands (shell history), 529 replies, 301 requesting directories, 14 requests, 301 required arguments, 92 reset_font_pts function, 451-453 reset_pts function, 451 resizing Box widget, 160 objects, 316 resource list, 144 resource names, 295 resource values, specifying, 153 resources, 128 ApplicationShell widget, 134 Box widget, 136 Command widget, 139-140 Constraint widget, 137 Core widget, 131-132 Form widget, 136 hard-coded. 152 indirect, 140 Label widgets, 138-139 Shell widget, 135 ToplevelShell widget, 134 widgets, 130 restore function, 361 restore PostScript operator, 486 restoring Arc objects, 435-436 functions, 366 Line objects, 401-403 Text objects, 474-475 return statement, 68-69

return types, 66 main function, 85 parameter list, 85 void, 68 returning allocated memory, 106 pointers, 102 values, 102 rho radius, 255 **RISC** (Reduced Instruction Set Computer), 20 Ritchie, Dennis, 25 rlineto PostScript operator, 487 rm command, 12, 17 rmdir command, 12-13 rmoveto PostScript operator, 487 root directories (2deditor), 540 rotate PostScript operator, 487 rotating arcs. 252 ellipses, 251 lines, 248-251 objects, 511 points, 248-251 rounding errors (scaling objects), 234 routines create_canvas, 348 find_handle, 337, 340 gx_delete, 322 gxSaveObjs, 372 line_from_box, 413 null_func routine, 345 rubber-banding object, 228 rubber_arc structure, 423 rubber_line->pts array, 383

rubber_line structure, 381-383 rubber_line variables, passing, 382 rubberGC globals, 353 running X clients, 117

S

-s flag, 524 safe macros, 146 save and restore common-object, 367-373 object-specific, 373-375 Save icon, 367 save PostScript operator, 487 saving functions, 366 objects, 372 Arc, 435 Line, 401-402 Text, 473-474 tagged lines, 365 scalable fonts, 309. See also vector fonts scalar value, 255 scale action, 233, 396 scale method, 317, 337, 398 scale PostScript operator, 487 scalefont command, 481 scalefont PostScript operator, 487 scaling objects, 232-233, 337 Arc, 432-433 data integrity, 234

direction, 233 handles, 233 Line, 398-400 lines, 235-238 rounding errors, 234 Text objects, 469-472 scope, 40, 88-90 screens, 133 scripting (Bourne shell), 531 segmentation violations, 79,95 segment_selected function, 390-391 select method, 317, 333, 392 selected field, 316 selecting handles, 337 objects, 306, 316, 331-333 Arc, 428 Line, 392, 395 Text, 464-467 widget sets, 128 semantic errors, 28, 65 semicolon (;), 47, 61, 527 separating arguments, 93 server requests, 119 servers, X, 114-116, 119 asynchronous handling of events, 301 connecting to, 144-151 starting, 115 set_color_data function, 500-501 set_cursor function, 349-351 setenv command, 526 setfont command, 481 setfont PostScript operator, 487

setgray PostScript operator, 487 setlinewidth PostScript operator, 487 setting sign bits, 73 sh (Bourne shell), 11, 523, 531 environment files, 527-529 scripting, 531 testing, 533-534 variables, defining, 525-526 shape parameters, 193 shell history, repeating commands, 529 shell scripts debugging, 538 variables, declaring, 532 Shell widget resources, 135 shells. See also command shells Bourne, 11, 523, 531 defining variables, 525-526 environment files, 527-529 scripting, 531 testing, 533-534 Bourne Again (bash), 12, 525, 531 C (csh), 11, 524-525 defining variables, 526 environment files, 530-531 command, 524 default, 524 Korn (ksh), 524-525, 531 TC (tcsh), 11, 524 UNIX, 11-12
short data type, 76 show PostScript operator, 487 showpage PostScript operator, 488 side functions, 241 sign bit, 71-73 simple italic-style vector font set, 557 single dot (.), 11 single quotation marks (' '), 93, 532 sizes (point), specifying, 480 slash asterisk (/*), 84 slash slash (//), 84 sleep command, 538 slope calculating, 216-218 lines, 216 slope calculations, comparing, 410 slope intercept linear equations, 217 source code, 18-19 source directories (src), 540 source files, 21-22 compiling, 22 gxGx.c, 367 invoking functions, 27 storing, 41-43 source pixels graphics pipeline, applying, 198 values (GC), 198 spaces, 36, 313 specifying ClipOrigin, 330 colors, 184

multiple flags, 14 names for command completion, 12 point size, 480 resource values, 153 typeface, 480 widget paths, 149 widgets, 153 sprintf function, 94 square braces ([]), 532 src source directory, 540 stack, 78 stack pointer, 20 stacks data structures dictionary stack, 481-483 operand stack, 481 LIFO lists, 481 PostScript, 481 standard error, 91 standard out, 91 standardization of C programming language, 25 starting vi editor, 23 X Server, 115 startx command, 115 stat command, 505 state indicators, using cursors as, 349-354 initializeGX function, 351-354 set_cursor function, 349-351 statements case, 58-60, 534 else, 52 if, 51-52, 532 if else, 57-58 return, 68-69 switch, 59

static keyword, 103-104, 227 static variables, 104, 227 status (objects), managing, 331 stderr. 91 stdout, 91 storage location contents, 79 storing object files, 41-43 source files, 41-43 values, 79 streat function, 99 strchr function, 100 strcmp function, 97-99 strcpy function, 99 strdup function, 100, 107 strerror function, 369 string compare functions, 97-99 string functions, 99-100 string library (C), 97 strings, 56, 85 comparing, 97-98 evaluating, 56 format, 92 integrity for use as filename, 371 lexical analysis, 98-99 manipulating, 100 strlen function, 100 strncat function, 99 strncmp function, 99 strncpy function, 99 stroke PostScript operator, 488 strokes (characters), 309 strstr function, 100

structure, 262. See also layout applications Graphics Editor example, 262-263 planning, 262-263 defining, 77 GXLine, 226, 306, 401 project (Graphics Editor), 244 resource names, 295 rubber_arc, 423 rubber_line, 381-383 XArc, 363-364, 425 XColor, 186 XGCValues, 181 stub functions, 367 sub PostScript operator, 488 sub-shells, 525 sum variable, 70 super-classes (widgets), 130 support for platforms, 42 switch statements, 59 symbolic links, 43-44, 542 synchronous behavior, enabling/disabling, 301 syntax getopt function, 290 **XtVaGetApplicationReso** urces function, 292 syntax errors, 28, 65 systems, coordinate, 252-253 Drawables, 253 negative numbers, 254 polar, 254-256 rectangular, 254 SYSV UNIX, 8

Т

Tab window manager, 118 tables, case, 534 tabs, 36 tagged file format, 359-360 tagged lines Line object, 365 saving, 365 **Tagged-File Format** strategies, combining with Position-Specific file format strategies, 364-366 TARGET variable, 40-42, 541 targets, 33-35, 41 body, 36 dependencies, 35-36 tcsh (TC shell), 11, 524 tcsh command, 11 template objects, 411. See also Box objects, Arrow objects terminal emulator, 91. See also windows terminals, 91 testing Bourne shells, 533-534 validity (event references), 381 tests explicit (non-null events), 230 implicit (non-null events), 230 num handles, 330 performing, 532 text formatting, 22 relating widgets to, 518-519

Text objects apply_scale function, 455-457 copy_gxtext function, 460-461 copying, 472-473 create_gxtext function, 444 code example, 447-449 GXText structure, 449-450 create_text function, 459-461 data file, 366 double quote character definition, 447 drawing, 461-463 erasing, 461-463 exclamation point character definition, 446-447 finding, 463-464 freeGXText function. 450-451 get_bounds function, 458-459 get_creation_text function, 439-443 get_extents function, 457-458 gx_text function, 438-439 GxDrawText function, 443-444 handles, 464, 466 moving, 467-469 next_char_min function, 454 place_creation_text function, 439-441 plain simplex array, 445-447 plain simplex p array, 445-447

reset_font_pts function, 451-453 reset_pts function, 451 restoring, 474-475 saving, 473-474 scaling, 469-472 selecting/deselecting, 464, 466-467 update_gxtext function, 444 text_bounding_handles function, 464-466 text_copy function, 472-473 text_deselect function, 466-467 text find function, 463-464 text_move function, 467-468 text_save function, 473-474 text scale function, 470 text_select function, 307-309, 464 theta angles, 255 third-party packages, 90 this keyword, 320 tilde (~), 12 tile field (XGCValues function), 188 tile flag, 388-389, 427 tile values, 389, 462 tiling, 348-349 tokens #!. 531 formatting, 92-93 **TOLERANCE** variable. 385, 214-215 toplevel window, 122 ToplevelShell widget, 134 Towers of Hanoi, 34 tracking number of points, 388

trackingEvent function, 516 translate PostScript operator, 488 translating C into Postscript, 480-481 source code, 18-19 translation tables, 303-304 translations, 133, 300, 303 triplex bold italic vector font set, 559 triplex bold vector font set, 558 type checking functions, 65 printf function, 93 typeface, specifying, 480

U

underscore (_), 67 Unicode, 55 UNIX command shells, 523-524 Bourne Again shell (bash), 525 Bourne shell (sh), 523 C shell (csh), 524-525 environments, 525 Korn shell (ksh), 524-525 TC shell (tcsh), 524 commands, aliases, 526-527 **UNIX** families, 8 UNIX shells, 11-12 unmapping windows, 121 unsafe macros, 146 unsigned modifier, 73-76 update_arc function, 425

update_box function, 413-414 update_gxtext function, 444 update_line function, 387 update_obj function, 335 update_pencil function, 407-408 updating applications cursor, 338-339 global variables, 337 rubber_line structures, 383 user interfaces aesthetics, 264-265 feature lists, 263-264 Graphics Editor gxArc.c file, 288-289 gxGraphics.c file, 271-276 gxGx.c file, 285-288 gxMain.c file, 265-271, 284-285 gxText.c file, 289 Graphics Editor features, 263-264 intuitiveness, 264-265 planning layout, 264-265 user navigation (objects), 334-341 userid, 12 utilities gmake, 31 gx allocate gc, 329 make, 24, 30-31 cc command, 25 Makefile. See Makefile

V

-v flag, 45 valid rubber_line structures, 383 validity, testing, 381 value masks, 181 values, 69 action fields, determining, 337 gxObjHeader, 324 gxObjs, 324 LineDashed, 511 LineDoubleDashed, 511 LineSolid, 511 resource, specifying, 153 returning, 102 scalar, 255 source pixels, 198 storing, 79 tile, 389, 462 variables, manipulating, 157 XGCValues function, 182 variable argument list, 144 variable data types, promoting, 93-94 variable pointers, 70 variable values, manipulating, 157 variables, 32-33 assigning, 322 automatic, 78-79 casting, 217 command shell environments, defining, 525-526 Cursor variables, 351 de-referencing, 79 declaring, 39-40, 69-70, 87 environment, 40 errno, 369

extern, 104-105, 369 FixedX, 399 FixedY, 399 fix_x, 414 fix_y, 414 getopt function, 290-291 GFStatusBar, 161 GLOBAL, 269 global, 88, 104 errno, 369 ExntX, 399 ExntY, 399 OrigX, 399 OrigY, 399 updating, 337 Xdebug, 301 GxActiveHandle, 337 GxHOME, 41 gxObjHeader, 322, 346 gxObjs, 324 GxSRCDIR, 41-42 indx, 384 initializing, 78-80, 98 naming, 32 path, 40-42 quoting, 532 rubber_line, passing, 382 scope, 88-90 shell scripts, declaring, 532 static, 104, 227 sum, 70 TARGET, 40-42, 541 TOLERANCE, 214-215 types, 66 vpath, 42-43 vector font sets, 557 simple italic-style, 557 triplex bold, 558 triplex bold italic, 559 vector fonts, 309-310 character definitions, 309-313

chars.h header file, 446 vector graphics, 199 vector text objects, 307-309 vectors, 199, 309 vector_chars.h header file, 446, 557, 560 vertex, 199 vi command, 84 vi editor, 22-24 modes command, 23-24 edit, 23 last-line, 23-24 moving from command mode to edit mode, 23 starting, 23 viewing PostScript files (GhostScript), 488-489 void return types, 68 vpath variable, 42-43 vpts field, 314

W

-W flag, 25-27 Web sites, GhostScript, 489 what-you-see-is-what-youget (WYSIWYG), 506 while loop, 62-63, 535 widget paths, specifying, 149 widget sets, 127 Athena. See Athena widget set Motif, 128, 140-142 selecting, 128 widgets, 126-129, 300 ApplicationShell, 133-135, 141 creating, 150 resources, 134

armed, 142 Box, 135 resizing, 160 resources, 136 buttons, placing, 127 callbacks, 302-303 classes, 129 Command, 139-140, 302 configuring, 147-148 Constraint, 137 Core, 130-133, 141 destroying, 371 dialog, creating, 370 event handlers, 302-303 Form, 136-138, 155 functionality, 130 instance hierarchy, 133 instantiation, 130 Label, 138-139 managing, 153, 371 paths, 149, 516-518 placing, 155 relating to text, 518-519 relationships, 127 resources, 128-130 Shell, 135 specifying, 153 super-classes, 130 ToplevelShell, 134 translation tables, 303-304 XmForm, 141 XmLabel, 142 XMPushButton, 142 XmRowColumn, 141 wildcards, 12-13 *, 13 [a-d], 13 window decorations, 118 window managers, 118 windows, 91, 119-121 child, 122

clipping, 122 creating in an application, 123 drawables, 182 events, processing, 127 hierarchy, 121-122 mapping, 121 origin, 122 parent, 121-122 toplevel, 122 unmapping, 121 X applications, 127, 175-176 word size, 21 world access fields, 16 world permissions, 16 wrapping multiple lines, 33 write permission, 15-17 write_ps function, 502-503 write_ps_color_conv function, 492-494, 503 write_ps_data function, 503 write_ps_prolog function, 499 writing ASCII files position-specific file format, 360-361 tagged file format, 359-360 PostScript page definition file, 502-504 WYSIWYG (what-yousee-is-what-you-get), 506

Х

X applications, 115, 126-127 X Clients, 116-119 X Display Manager, 115 X event masks, 157-158 x field, 314 -x flag, 538 X Server, 114-116, 119 asynchronous handling of events, 301 connecting to X-based applications, 144-151 XtAppInitialize, 144 XtVaAppInitialize, 144-151 graphics pipeline, 197 starting, 115 X Server requests, 119 X Toolkit Intrinsics. See XT X Window applications, 301 X Window server. See X Server X Window System graphics, 197 history, 113-114 limitations, 224 X-based applications application interfaces, creating, 151-158 buttons, creating, 159-162 components, 143 events, processing, 176 windows, managing, 175-176 X server, connecting to, 144 X11/Xaw/Dialog.h header file, 370 X11/Xaw/StringDefs.h header file, 370 XAllocNameColor function, 185-187 XArc structure, 363-364, 425

xcalc application, 119-121 xcalc command, 117 **XCGValues** function. 183-184 line_width field, 187 tile field, 188 values. 182 xclock command, 117 **XColor structure**, 186 xconsole command, 117 Xdebug global variable, 301 **XDefineCursor** function, 351 xdm command, 115-117 xdpvinfo command, 117 XDrawArc function, 192-193 XDrawLine function, 191, 212, 410 **XDrawPoint function**, 190-191 XDrawRectangle function, 191 xedit command, 117 xev command, 117 XEvent parameter, 326 xeves command, 117 XFill function, 193 XFillRectangle graphic primitive, 330 xfontsel command, 117 **XGCValues** function, 188-189 XGCValues structure, 181 XGetImage function, 498 XImage, creating, 497-499 xinit command, 115 Xlib (X library) graphic primatives XDrawArc function, 192-193

XDrawLine function, 191 XDrawPoint function, 190-191 XDrawRectangle function, 191 XFill functions, 193 relationships, 127 Xlib function, 180-184, 301 xload command, 117 xman command, 117 xmessage command, 117 XmForm widget, 141 XmLabel widget, 142 XmPushButton widget, 142 XmRowColumn widget, 141 XmRwColumn widget, 141 XPoint array, 307 XRectangle rubber_box data structure, 413 xset command, 117 Xt (X Toolkit Intrinsics), **XtVaGetApplication** Resources, 125, 290-292, 300 arguments, 295 arrays, 292-294 code example, 294-295 parameters, 292-294 relationships, 127 syntax, 292 XtAddEventHandler function, 276 Xt command-line parameters, 148 Xt function, 180 XtAddCallback function, 161 XtAddEventHandler, 303

XtAddEventHandler function. 156 XtAddEventHandler functions, 276 **XtAppInitialize function**, 144 XtAppMainLoop function, 176, 285, 299 XtAppNextEvent function, 300 XtArgVal data type, 144 **XtAugmentTranslations** function, 304 XtCallback lists, 133 **XtDispatchEvent function**, 300 xterm command, 117 XtMalloc function, 408 XtNew function, 236 **XtParseTranslationTable** function, 304 **XtRealloc function**, 408 XtSetArg macro, 145 XtVaAppInitialize function, 144-151 XtVaCreateManagedWidg et function, 151-158 **XtVaGetApplicationResour** ces functions, 290-292 arguments, 295 arrays, 292-294 code example, 294-295 parameters, 292-294 syntax, 292 XYPixmap argument, 499

Y-Z

y fields, 314 ZPixmap argument, 499