

国外著名高等院校
信息科学与技术优秀教材

Prentice
Hall

C 语言参考手册 (第五版)

A Reference Manual

FIFTH EDITION

Samuel P. Harbison III ■ Guy L. Steele Jr.

英文版

人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

国外著名高等院校信息科学与技术优秀教材

C 语言参考手册 (第五版)

(英文版)

C A Reference Manual, Fifth Edition

Samuel P. Harbison III

Guy L. Steele Jr.

人 民 邮 电 出 版 社

图书在版编目 (CIP) 数据

C 语言参考手册: 第 5 版 / () 哈比森 (Harbison, S.P.), () 斯蒂尔 (Steele, G.L.) 著.
—北京: 人民邮电出版社, 2003.6

国外著名高等院校信息科学与技术优秀教材

ISBN 7-115-11194-4

I. C... II. ①哈... ②斯... III. C 语言—程序设计—高等学校—教材—英文 IV. TP312
中国版本图书馆 CIP 数据核字 (2003) 第 019941 号

版 权 声 明

English Reprint Edition Copyright © 2003 by PEARSON EDUCATION NORTH ASIA LIMITED and POSTS & TELECOMMUNICATIONS PRESS.

C A Reference Manual, Fifth Edition

By Samuel P. Harbison III Guy L. Steele Jr.

Copyright © 2002

All Rights Reserved.

Published by arrangement with Prentice Hall, Pearson Education, Inc.

The edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative of Hong Kong and Macau).

本书封面贴有 **Pearson Education** (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

国外著名高等院校信息科学与技术优秀教材

C 语言参考手册 (第五版) (英文版)

◆ Samuel P. Harbison III Guy L. Steele Jr.

责任编辑 陈冀康

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

读者热线 010-67132705

北京汉魂图文设计有限公司制作

北京朝阳展望印刷厂印刷

新华书店总店北京发行所经销

◆ 开本: 800×1000 1/16

印张: 34.5

字数: 773 千字

2003 年 6 月第 1 版

印数: 1-3 000 册

2003 年 6 月北京第 1 次印刷

著作权合同登记 图字: 01-2003-0893 号

ISBN 7-115-11194-4/TP·3400

定价: 45.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

内 容 提 要

本书是关于 C 语言的详尽的参考手册。最新的第五版增加了对 C99 标准的介绍，更加符合当前 C 语言学习的需要。

全书共两大部分 24 章。第一部分讨论了 C 语言的所有语言特征，包括词法、预处理机制、声明、类型、表达式、语句以及函数等基本语言特性的介绍。第二部分讨论了 C 语言的标准库，根据它们不同的功能分别详细介绍。为帮助读者理解相关概念，本书在讨论 C 语言及其标准库的各方面问题时，提供了许多实例和解释。许多章里还提供了练习题，书后给出了主要练习的解答。这些练习的主要目的也为帮助读者理解 C 语言的基本机制及其重要细节，其中并不涉及复杂的程序设计技术与问题。

本书可以作为高等院校计算机专业 C 语言课程的教材和参考书，对于一般的 C 程序员也有重要的参考价值。

出版说明

2001 年，教育部印发了《关于“十五”期间普通高等教育教材建设与改革的意见》。该文件明确指出，“九五”期间原国家教委在“抓好重点教材，全面提高质量”方针指导下，调动了各方面的积极性，产生了一大批具有改革特色的新教材。然而随着科学技术的飞速发展，目前高校教材建设工作仍滞后于教学改革的实践，一些教材内容陈旧，不能满足按新的专业目录修订的教学计划和课程设置的需要。为此该文件明确强调，要加强国外教材的引进工作。当前，引进的重点是信息科学与技术 and 生物科学与技术两大学科的教材。要根据专业（课程）建设的需要，通过深入调查、专家论证，引进国外优秀教材。要注意引进教材的系统配套，加强对引进教材的宣传，促进引进教材的使用和推广。

邓小平同志早在 1977 年就明确指出：“要引进外国教材，吸收外国教材中有益的东西。”随着我国加入 WTO，信息产业的国际竞争将日趋激烈，我们必须尽快培养出大批具有国际竞争能力的高水平信息技术人才。教材是一个很关键的问题，国外的一些优秀教材不但内容新，而且还提供了很多新的研究方法和思考方式。引进国外原版教材，可以促进我国教学水平的提高，提高学生的英语水平和学习能力，保证我们培养出的学生具有国际水准。

为了贯彻中央“科教兴国”的方针，配合国内高等教育教材建设的需要，人民邮电出版社约请有关专家反复论证，与国外知名的教材出版公司合作，陆续引进一些信息科学与技术优秀教材。第一批教材针对计算机专业的主干核心课程，是国外著名高等院校所采用的教材，教材的作者都是在相关领域享有盛名的专家教授。这些教材内容新，反映了计算机科学技术的最新发展，对全面提高我国信息科学与技术的教学水平必将起到巨大的推动作用。

出版国外著名高等院校信息科学与技术优秀教材的工作将是一个长期的、坚持不懈的过程，我社网站（www.ptpress.com.cn）上介绍了我们首批陆续推出的图书的详细情况，后续教材的引进和出版情况我们会及时在网上发布，敬请关注。希望广大教师和学生将使用中的意见和建议及时反馈给我们，我们将根据您的反馈不断改进我们的工作，推出更多更好的引进版信息科学与技术教材。

人民邮电出版社

2001 年 12 月

序

本书为 2002 年出版的 *C A Reference Manual*，它的先前版本由于其准确性和易用性，一直受到 C 程序设计工作者和教授 C 语言程序设计的教师们的广泛推崇。这个新版本保持了原有的风格和特色，在原书基本框架下增加了许多内容，特别是增加了对最新的 C99 语言标准的介绍。熟悉程序设计语言的人都应该知道作者之一的 Guy L. Steele Jr.，他是一位很有影响的计算机科学家，也是最有名的语言手册写作者，他独立或与人合作撰写了多种语言的手册，包括著名的 Scheme、Common Lisp 和 Java 等。另一作者 Samuel P. Harbison III 也是 C 语言的专家。

本书可以看作是一本“带标注的 C 语言手册”，类似于国外常见的其他名为“Annotated XX Reference Manual”的书籍。这种书基本上是按语言手册的方式组织内容，分门别类地介绍一种语言的各方面特征。但它们又与纯粹的手册不同，在介绍语言特征时，作者常根据情况加入一些说明性的实例，提供一些背景情况的介绍或者解释。当然，这些实例和解释的基本目的是帮助读者理解这一语言。本书是一本完整的 C 语言手册，内容分为两部分。第一部分讨论了 C 语言的所有语言特征，也包括了对预处理机制的详尽介绍。第二部分讨论了 C 语言的标准库。为帮助读者理解相关概念，本书在讨论 C 语言及其标准库的各方面问题时，提供了许多实例和解释。许多章里还提供了练习题，书后给出了主要练习的解答。这些练习的主要目的也为帮助读者理解 C 语言的基本机制及其重要细节，并不涉及复杂的程序设计技术与问题。

本书的另一个特点就是将 K&R C、ANSI C 和 C99 放在同一个框架里，互相对照着一起介绍。在讨论 C 语言的各种特征时，作者都介绍了 C 语言的不同“标准”的规定。这种组织方式有助于读者看清不同“标准”的变化和 C 语言的发展。书中还包含了 C 语言中的一些方面与 C++ 标准的兼容性问题，可以帮助读者理解这两种关系密切的语言之间的细节差异。

如前所述，这一新版本的最重要修订就是包含了对 C99 标准的完整介绍，包括新标准引入的新的关键字和基本数据类型，变长数组和变长数组参数的定义和使用，数组、结构和联合成员初始化的选定描述方式等许多重要变化。在讨论标准库的第二部分里，也详细介绍了 C99 标准库各方面的新情况。虽然 C99 标准已

经过几年了，但现有书籍中完整地讨论这一标准的还很少见，本书的引进将能填补国内在这方面的空白，可供关心 C99 的专业工作者和教师学生参考。

作为 C 语言参考书，这本书是非常有价值的。我个人就觉得手边很需要这样一本参考手册，以便在编程和教学工作中随时查阅。目前市场上见到的与 C 语言有关的书籍中许多为教科书（也另有些定位不清的书籍），由于其写作目的和用途定位，不少书籍中并没有涵盖 C 语言的全部机制。许多书籍是从教学需要，或者从某种实用需要出发，介绍了 C 语言中一些方面的情况。这样一本参考书定会使学习 C 语言程序设计的人们得到很大帮助，也能给在教学、科研和系统开发中使用 C 语言的专业工作者们提供许多信息。

最后还想就本书的使用提一点建议：如果要将它用于自学或者课程，最好能配以适当的讨论程序设计问题的教材，这样才能更好地发挥它的作用。

裘宗燕

北京大学数学学院信息科学系

2003 年 2 月

Preface

This text is a reference manual for the C programming language. Our aim is to provide a complete and precise discussion of the language, the run-time libraries, and a style of C programming that emphasizes correctness, portability, and maintainability.

We expect our readers to already understand basic programming concepts, and many will be experienced C programmers. In keeping with a reference format, we present the language in a bottom-up order: lexical structure, preprocessor, declarations, types, expressions, statements, functions, and run-time libraries. We have included many cross-references in the text so that readers can begin at any point.

This Fifth Edition now includes a complete description of the latest international C standard, ISO/IEC 9899:1999 (C99). I have been careful to indicate which features of the language and libraries are new in C99 and point out how C99 differs from the previous standard, C89. This is now the only book that serves as a reference for all the major versions of the C language: traditional C, the 1989 C Standard, the 1995 Amendment and Corrigenda to C89, and now the 1999 C Standard. It also covers the Clean C subset of Standard C and Standard C++. Although there is much new material in C99, I have not changed the chapter and section organization of the book significantly, so readers familiar with previous editions will not have problems finding the information they need.

This book originally grew out of our work at Tartan, Inc. developing a family of C compilers for a range of computers—from micros to mainframes. We wanted the compilers to be well documented, provide precise and helpful error diagnostics, and generate exceptionally efficient object code. A C program that compiles correctly with one compiler must compile correctly under all the others insofar as the hardware differences allow.

In 1984, despite C's popularity, we found that there was no description of C precise enough to guide us in designing the new compilers. Similarly, no existing description was precise enough for our programmer/customers, who would be using compilers that analyzed C programs more thoroughly than was the custom at that time. In this text, we have been especially sensitive to language features that affect program clarity, object code efficiency, and the portability of programs among different environments.

WEB SITE

We encourage readers to visit the book's Web site: **CAResourceManual.com**. We'll post example code, expanded discussions, clarifications, and links to more C resources.

ACKNOWLEDGMENTS

In preparing this Fifth Edition, I want to especially acknowledge the critical help I received from Rex Jaeschke, former chairman of NCITS J11; Antoine Trux of Helsinki, Finland; and Steve Adamczyk, the founder of Edison Design Group.

For assistance with previous editions, I would like to thank Jeffrey Esakov, Alan J. Filipski, Frank J. Wagner, Debra Martin, P. J. Plauger, and Steve Vinoski. Other help came from Aurelio Bignoli, Steve Clamage, Arthur Evans, Jr., Roy J. Fuller, Morris M. Kessan, George V. Reilly, Mark Lan, Mike Hewett, Charles Fischer, Kevin Rodgers, Tom Gibb, David Lim, Stavros Macrakis, Steve Vegdahl, Christopher Vickery, Peter van der Linden, and Dave Wilson. Also Michael Angus, Mady Bauer, Larry Breed, Sue Broughton, Alex Czajkowski, Robert Firth, David Gaffney, Steve Gorman, Dennis Hamilton, Chris Hanna, Ken Harrenstien, Rex Jaeschke, Don Lindsay, Tom MacDonald, Peter Nelson, Joe Newcomer, Kevin Nolish, David Notkin, Peter Plamondon, Roger Ray, Larry Rosler, David Spencer, and Barbara Steele.

Some of the original example programs in this book were inspired by algorithms appearing in the following works:

- Beeler, Michael, Gosper, R. William, and Schroepfel, Richard, *HAKMEM*, AI Memo 239 (Massachusetts Institute of Technology Artificial Intelligence Laboratory, February 1972);
- Bentley, Jon Louis, *Writing Efficient Programs* (Prentice-Hall, 1982);
- Bentley, Jon Louis, "Programming Pearls" (monthly column appearing in *Communications of the ACM* beginning August 1983);
- Kernighan, Brian W., and Ritchie, Dennis M., *The C Programming Language* (Prentice-Hall, 1978);
- Knuth, Donald E., *The Art of Computer Programming* Volumes 1–3 (Addison-Wesley, 1968, 1969, 1973, 1981); and
- Sedgewick, Robert, *Algorithms* (Addison-Wesley, 1983).

We are indebted to these authors for their good ideas.

The use of *I* instead of *we* in this Preface reflects that Guy Steele's work load has prevented him from being an active contributor to recent editions. The text still reflects his clear and rigorous analysis of the C language, but he cannot be held responsible for any new problems in this edition.

C: A Reference Manual is now over 17 years old. To all our readers: Thank you!

Sam Harbison
Pittsburgh, PA
harbison@CAREferenceManual.com

List of Tables

Table 2-1	Graphic characters	12
Table 2-2	ISO trigraphs	15
Table 2-3	Operators and separators	21
Table 2-4	Keywords in C99	23
Table 2-5	Types of integer constants	27
Table 2-6	Assignment of types to integer constants	28
Table 2-7	Character escape codes	36
Table 2-8	Additional C++ keywords	39
Table 3-1	Preprocessor commands	44
Table 3-2	Predefined macros	52
Table 4-1	Identifier scopes	75
Table 4-2	Overloading classes	78
Table 4-3	Storage class specifiers	83
Table 4-4	Default storage class specifiers	84
Table 4-5	Function declarators	101
Table 4-6	Form of initializers	104
Table 4-7	Interpretation of top-level declarations	115
Table 5-1	C types and categories	124
Table 5-2	Values defined in <code>limits.h</code>	127
Table 5-3	Values defined in <code>float.h</code>	134
Table 5-4	IEEE floating-point characteristics	135
Table 6-1	Memory models on early PCs	186
Table 6-2	Permitted casting conversions	194
Table 6-3	Allowable assignment conversions	195
Table 6-4	Conversion rank	196
Table 6-5	Usual unary conversions (choose first that applies)	197
Table 6-6	Usual binary conversions (choose first that applies)	199
Table 7-1	Nonarray expressions that can be lvalues	204
Table 7-2	Operators requiring lvalue operands	204
Table 7-3	C operators in order of precedence	205
Table 7-4	Binary operator expressions	227
Table 7-5	Conditional expression 2nd and 3rd operands (pre-Standard)	245
Table 7-6	Conditional expression 2nd and 3rd operands (Standard C)	245
Table 7-7	Assignment operands	247

Contents

PART 1 The C Language	1
1 Introduction	3
1.1 The Evolution of C	3
1.2 Which Dialect of C Should You Use?	6
1.3 An Overview of C Programming	7
1.4 Conformance	8
1.5 Syntax Notation	9
2 Lexical Elements	11
2.1 Character Set	11
2.2 Comments	18
2.3 Tokens	20
2.4 Operators and Separators	20
2.5 Identifiers	21
2.6 Keywords	23
2.7 Constants	24
2.8 C++ Compatibility	38
2.9 On Character Sets, Repertoires, and Encodings	39
2.10 Exercises	41
3 The C Preprocessor	43
3.1 Preprocessor Commands	43
3.2 Preprocessor Lexical Conventions	44
3.3 Definition and Replacement	46
3.4 File Inclusion	59
3.5 Conditional Compilation	61
3.6 Explicit Line Numbering	66
3.7 Pragma Directive	67
3.8 Error Directive	69
3.9 C++ Compatibility	70
3.10 Exercises	71

4 Declarations**73**

- 4.1 Organization of Declarations 74
- 4.2 Terminology 75
- 4.3 Storage Class and Function Specifiers 83
- 4.4 Type Specifiers and Qualifiers 86
- 4.5 Declarators 95
- 4.6 Initializers 103
- 4.7 Implicit Declarations 113
- 4.8 External Names 113
- 4.9 C++ Compatibility 116
- 4.10 Exercises 119

5 Types**123**

- 5.1 Integer Types 124
- 5.2 Floating-Point Types 132
- 5.3 Pointer Types 136
- 5.4 Array Types 140
- 5.5 Enumerated Types 145
- 5.6 Structure Types 148
- 5.7 Union Types 160
- 5.8 Function Types 165
- 5.9 The Void Type 168
- 5.10 Typedef Names 168
- 5.11 Type Compatibility 172
- 5.12 Type Names and Abstract Declarators 176
- 5.13 C++ Compatibility 178
- 5.14 Exercises 179

6 Conversions and Representations**181**

- 6.1 Representations 181
- 6.2 Conversions 188
- 6.3 The Usual Conversions 194
- 6.4 C++ Compatibility 200
- 6.5 Exercises 201

7 Expressions**203**

- 7.1 Objects, Lvalues, and Designators 203
- 7.2 Expressions and Precedence 204
- 7.3 Primary Expressions 207
- 7.4 Postfix Expressions 210
- 7.5 Unary Expressions 219
- 7.6 Binary Operator Expressions 227
- 7.7 Logical Operator Expressions 243
- 7.8 Conditional Expressions 244
- 7.9 Assignment Expressions 246
- 7.10 Sequential Expressions 249

7.11	Constant Expressions	250
7.12	Order of Evaluation	253
7.13	Discarded Values	255
7.14	Optimization of Memory Accesses	256
7.15	C++ Compatibility	257
7.16	Exercises	258
8	Statements	259
8.1	General Syntactic Rules for Statements	260
8.2	Expression Statements	260
8.3	Labeled Statements	261
8.4	Compound Statements	262
8.5	Conditional Statements	264
8.6	Iterative Statements	266
8.7	Switch Statements	274
8.8	Break and Continue Statements	277
8.9	Return Statements	279
8.10	Goto Statements	280
8.11	Null Statements	281
8.12	C++ Compatibility	282
8.13	Exercises	282
9	Functions	285
9.1	Function Definitions	286
9.2	Function Prototypes	289
9.3	Formal Parameter Declarations	295
9.4	Adjustments to Parameter Types	298
9.5	Parameter-Passing Conventions	299
9.6	Agreement of Parameters	300
9.7	Function Return Types	301
9.8	Agreement of Return Types	302
9.9	The Main Program	303
9.10	Inline Functions	304
9.11	C++ Compatibility	306
9.12	Exercises	307
PART 2	The C Libraries	309
10	Introduction to the Libraries	311
10.1	Standard C Facilities	312
10.2	C++ Compatibility	313
10.3	Library Headers and Names	316
11	Standard Language Additions	325
11.1	NULL, ptrdiff_t, size_t, offsetof	325
11.2	EDOM, ERANGE, EILSEQ, errno, strerror, perror	327
11.3	bool, false, true	329

11.4	va_list, va_start, va_arg, va_end	329
11.5	Standard C Operator Macros	333
12	Character Processing	335
12.1	isalnum, isalpha, iscntrl, iswalnum, iswalpha, iswcntrl	336
12.2	iscsym, iscsymf	338
12.3	isdigit, isodigit, isxdigit, iswdigit, iswxdigit	338
12.4	isgraph, isprint, ispunct, iswgraph, iswprint, iswpunct	339
12.5	islower, isupper, iswlower, iswupper	340
12.6	isblank, isspace, iswhite, iswspace	341
12.7	toascii	341
12.8	toint	342
12.9	tolower, toupper, tolower, toupper	342
12.10	wctype_t, wctype, iswctype	343
12.11	wctrans_t, wctrans	344
13	String Processing	347
13.1	strcat, strncat, wscat, wcsncat	348
13.2	strcmp, strncmp, wscmp, wcsncmp	349
13.3	strcpy, strncpy, wscpy, wcsncpy	350
13.4	strlen, wcslen	351
13.5	strchr, strchr, wcschr, wcschr	351
13.6	strspn, strcspn, strpbrk, strpbrk, wcsspn, wscspn, wcpbrk	352
13.7	strstr, strtok, wcsstr, wcstok	354
13.8	strtod, strtod, strtold, strtol, strtoll, strtoul, strtoull	355
13.9	atof, atoi, atol, atoll	356
13.10	strcoll, strxfrm, wscoll, wcsxfrm	356
14	Memory Functions	359
14.1	memchr, wmemchr	359
14.2	memcmp, wmemcmp	360
14.3	memcpy, memcpy, memmove, wmemcpy, wmemmove	361
14.4	memset, wmemset	362
15	Input/Output Facilities	363
15.1	FILE, EOF, wchar_t, wint_t, WEOF	365
15.2	fopen, fclose, fflush, freopen, fwide	366
15.3	setbuf, setvbuf	370
15.4	stdin, stdout, stderr	371
15.5	fseek, ftell, rewind, fgetpos, fsetpos	372
15.6	fgetc, fgetwc,getc, getwc, getchar, getwchar, ungetc, ungetwc	374
15.7	fgets, fgetws, gets	376
15.8	fscanf, fwscanf, scanf, wscanf, sscanf, swscanf	377
15.9	fputc, fputwc,putc, putwc, putchar, putwchar	385
15.10	fputs, fputws, puts	386
15.11	fprintf, printf, sprintf, snprintf, fwprintf, wprintf, swprintf	387

- 15.12 `v[x]printf, v[x]scanf` 401
- 15.13 `fread, fwrite` 402
- 15.14 `feof, ferror, clearerr` 404
- 15.15 `remove, rename` 404
- 15.16 `tmpfile, tmpnam, mktemp` 405

16 General Utilities**407**

- 16.1 `malloc, calloc, mlalloc, clalloc, free, cfree` 407
- 16.2 `rand, srand, RAND_MAX` 410
- 16.3 `atof, atoi, atol, atoll` 411
- 16.4 `strtod, strtod, strtold, strtol, strtoll, strtoul, strtoull` 412
- 16.5 `abort, atexit, exit, _Exit, EXIT_FAILURE, EXIT_SUCCESS` 414
- 16.6 `getenv` 415
- 16.7 `system` 416
- 16.8 `bsearch, qsort` 417
- 16.9 `abs, labs, llabs, div, ldiv, lldiv` 419
- 16.10 `mblen, mbtowc, wctomb` 420
- 16.11 `mbstowcs, wcstombs` 422

17 Mathematical Functions**425**

- 17.1 `abs, labs, llabs, div, ldiv, lldiv` 426
- 17.2 `fabs` 426
- 17.3 `ceil, floor, lrint, llrint, lround, llround, nearbyint, round, rint, trunc` 427
- 17.4 `fmod, remainder, remquo` 428
- 17.5 `frexp, ldexp, modf, scalbn` 429
- 17.6 `exp, exp2, expm1, ilogb, log, log10, log1p, log2, logb` 430
- 17.7 `cbrt, fma, hypot, pow, sqrt` 432
- 17.8 `rand, srand, RAND_MAX` 432
- 17.9 `cos, sin, tan, cosh, sinh, tanh` 433
- 17.10 `acos, asin, atan, atan2, acosh, asinh, atanh` 434
- 17.11 `fdim, fmax, fmin` 435
- 17.12 `Type-Generic Macros` 435
- 17.13 `erf, erfc, lgamma, tgamma` 439
- 17.14 `fpclassify, isfinite, isinf, isnan, isnormal, signbit` 440
- 17.15 `copysign, nan, nextafter, nexttoward` 441
- 17.16 `isgreater, isgreaterequal, isless, islessequal, islessgreater, isunordered` 442

18 Time and Date Functions**443**

- 18.1 `clock, clock_t, CLOCKS_PER_SEC, times` 443
- 18.2 `time, time_t` 445
- 18.3 `asctime, ctime` 445
- 18.4 `gmtime, localtime, mktime` 446
- 18.5 `difftime` 447

18.6	strftime, wcsftime	448
19	Control Functions	453
19.1	assert, NDEBUG	453
19.2	system, exec	454
19.3	exit, abort	454
19.4	setjmp, longjmp, jmp_buf	454
19.5	atexit	456
19.6	signal, raise, gsignal, ssignal, psignal	456
19.7	sleep, alarm	458
20	Locale	461
20.1	setlocale	461
20.2	localeconv	463
21	Extended Integer Types	467
21.1	General Rules	467
21.2	Exact-Size Integer Types	470
21.3	Least-Size Types of a Minimum Width	471
21.4	Fast Types of a Minimum Width	472
21.5	Pointer-Size and Maximum-Size Integer Types	473
21.6	Ranges of ptrdiff_t, size_t, wchar_t, wint_t, and sig_atomic_t	474
21.7	imaxabs, imaxdiv, imaxdiv_t	474
21.8	strtoimax, strtouimax	475
21.9	wcstoimax, wcstouimax	475
22	Floating-Point Environment	477
22.1	Overview	477
22.2	Floating-Point Environment	478
22.3	Floating-Point Exceptions	479
22.4	Floating-Point Rounding Modes	481
22.5	Floating-Point Expression Contraction	481
23	Complex Arithmetic	483
23.1	Complex Library Conventions	483
23.2	complex, _Complex_I, imaginary, _Imaginary_I, I	484
23.3	CX_LIMITED_RANGE	484
23.4	cacos, casin, catan, ccosh, csinh, ctanh	485
23.5	cacosh, casinh, catanh, ccosh, csinh, ctanh	486
23.6	cexp, clog, cabs, cpow, csqrt	487
23.7	carg, cimag, creal, conj, cproj	488
24	Wide and Multibyte Facilities	489
24.1	Basic Types and Macros	489
24.2	Conversions Between Wide and Multibyte Characters	490
24.3	Conversions Between Wide and Multibyte Strings	491
24.4	Conversions to Arithmetic Types	493
24.5	Input and Output Functions	493

Contents	7
24.6 String Functions	493
24.7 Date and Time Conversions	494
24.8 Wide-Character Classification and Mapping Functions	494
A The ASCII Character Set	497
B Syntax	499
C Answers to the Exercises	513
Index	521