

C Programming: Just the FAQs

C PROGRAMMING: JUST THE FAQs

Paul S. R. Chisholm
David Hanley
Michael Jones
Michael Lindner
Lloyd Work

SAMS
PUBLISHING

201 West 103rd Street
Indianapolis, Indiana 46290

FREE EBOOKS, NOTES , VIDEOS & PLACEMENT MATERIAL



For All Companies placement
Material

@placementclasses



For CAT Exam Preparation
Material

@cat_classes



For GATE Exam Preparation
Material

@gate_classes



For Engineering Books &
Material

@cs_ebooks



Codes of Following Programming Languages



C

@c_examples



C++

@cpp_examples



Java

@java_examples0



Python

@python_examples

Copyright © 1995 by Sams Publishing

FIRST EDITION

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. For information, address Sams Publishing, 201 W. 103rd St., Indianapolis, IN 46290.

International Standard Book Number: 0-672-30561-5

Library of Congress Catalog Card Number: 94-66635

98 97 96 95 4 3 2 1

Interpretation of the printing code: the rightmost double-digit number is the year of the book's printing; the rightmost single-digit, the number of the book's printing. For example, a printing code of 95-1 shows that the first printing of the book occurred in 1995.

Composed in AGaramond and MCPdigital by Macmillan Computer Publishing

Printed in the United States of America

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Sams Publishing 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.

Publisher

Richard K. Swadley

Acquisitions Manager

Greg Wiegand

Development Manager

Dean Miller

Managing Editor

Cindy Morrow

Acquisitions Editor

Chris Denny

Development Editor

Brad Jones

Production Editor

Cheri Clark

Editorial Coordinator

Bill Whitmer

Editorial Assistants

Carol Ackerman

Sharon Cox

Lynette Quinn

Technical Reviewer

Greg Guntle

Marketing Manager

Gregg Bushyeager

Assistant Marketing Manager

Michelle Milner

Cover Designer

Tim Amrhein

Book Designer

Alyssa Yesh

Vice President of
Manufacturing and Production

Jeff Valler

Manufacturing Coordinator

Paul Gilchrist

Imprint Manager

Kelly Dobbs

Team Supervisor

Katy Bodenmiller

Support Services Manager

Juli Cook

Support Services Supervisor

Mary Beth Wakefield

Production Analysts

Angela Bannan

Dennis Clay Hager

Bobbi Satterfield

Page Layout

Carol Bowers

Charlotte Clapp

Mary Ann Cosby

Aleata Howard

Louisa Klucznik

Casey Price

Jill Tompkins

Mark Walche

Dennis Wesner

Proofreading

Georgiana Briggs

Mona Brown

Michael Brumitt

Donna Harbin

Michael Henry

Kevin Laseau

Donna Martin

Indexer

Cheryl Dietsch

Greg Eldred

Overview

Introduction	xxix
I The C Language	1
II Variables and Data Storage	15
III Sorting and Searching Data	31
IV Data Files	63
V Working with the Preprocessor	87
VI Working with Strings	115
VII Pointers and Memory Allocation	131
VIII Functions	159
IX Arrays	175
X Bits and Bytes	189
XI Debugging	197
XII Standard Library Functions	215
XIII Times and Dates	243
XIV System Calls	255
XV Portability	275
XVI ANSI/ISO Standards	283
XVII User Interface—Screen and Keyboard	293
XVIII Writing and Compiling Your Programs	315
XIX Programming Style and Standards	331
XX Miscellaneous	349
XXI Windows	385
Index	415

Contents

Introduction	xxix
I The C Language	1
I.1: What is a local block?	1
Answer:	1
Cross Reference:	3
I.2: Should variables be stored in local blocks?	3
Answer:	3
Cross Reference:	3
I.3: When is a <i>switch</i> statement better than multiple <i>if</i> statements?	3
Answer:	3
Cross Reference:	4
I.4: Is a default case necessary in a <i>switch</i> statement?	4
Answer:	4
Cross Reference:	5
I.5: Can the last case of a <i>switch</i> statement skip including the <i>break</i> ?	5
Answer:	5
Cross Reference:	5
I.6: Other than in a <i>for</i> statement, when is the comma operator used?	6
Answer:	6
Cross Reference:	7
I.7: How can you tell whether a loop ended prematurely?	7
Answer:	7
Cross Reference:	8
I.8: What is the difference between <i>goto</i> and <i>longjmp()</i> and <i>setjmp()</i> ?	8
Answer:	8
Cross Reference:	10
I.9: What is an lvalue?	10
Answer:	10
Cross Reference:	11
I.10: Can an array be an lvalue?	11
Answer:	11
Cross Reference:	12
I.11: What is an rvalue?	12
Answer:	12
Cross Reference:	12
I.12: Is left-to-right or right-to-left order guaranteed for operator precedence?	12
Answer:	12
Cross Reference:	13

I.13: What is the difference between <code>++var</code> and <code>var++</code> ?	13
Answer:	13
Cross Reference:	14
I.14: What does the modulus operator do?	14
Answer:	14
Cross Reference:	14
II Variables and Data Storage	15
II.1: Where in memory are my variables stored?	16
Answer:	16
Cross Reference:	16
II.2: Do variables need to be initialized?	16
Answer:	16
Cross Reference:	17
II.3: What is page thrashing?	17
Answer:	17
Cross Reference:	18
II.4: What is a <i>const</i> pointer?	18
Answer:	18
Cross Reference:	19
II.5: When should the register modifier be used? Does it really help?	19
Answer:	19
Cross Reference:	20
II.6: When should the <i>volatile</i> modifier be used?	20
Answer:	20
Cross Reference:	21
II.7: Can a variable be both <i>const</i> and <i>volatile</i> ?	21
Answer:	21
Cross Reference:	21
II.8: When should the <i>const</i> modifier be used?	21
Answer:	21
Cross Reference:	22
II.9: How reliable are floating-point comparisons?	22
Answer:	22
Cross Reference:	23
II.10: How can you determine the maximum value that a numeric variable can hold?	23
Answer:	23
Cross Reference:	24
II.11: Are there any problems with performing mathematical operations on different variable types?	24
Answer:	24
Cross Reference:	25
II.12: What is operator promotion?	25
Answer:	25
Cross Reference:	26

II.13: When should a type cast be used?	26
Answer:	26
Cross Reference:	26
II.14: When should a type cast not be used?	27
Answer:	27
Cross Reference:	27
II.15: Is it acceptable to declare/define a variable in a C header?	27
Answer:	27
Cross Reference:	27
II.16: What is the difference between declaring a variable and defining a variable?	28
Answer:	28
Cross Reference:	28
II.17: Can <i>static</i> variables be declared in a header file?	28
Answer:	28
Cross Reference:	28
II.18: What is the benefit of using <i>const</i> for declaring constants?	29
Answer:	29
Cross Reference:	29
III Sorting and Searching Data	31
Sorting	31
Searching	32
Performance of Sorting or Searching	33
Some Code to Get Started With	36
III.1: What is the easiest sorting method to use?	36
Answer:	36
Cross Reference:	37
III.2: What is the quickest sorting method to use?	37
Answer:	37
Cross Reference:	44
III.3: How can I sort things that are too large to bring into memory?	44
Answer:	44
Cross Reference:	48
III.4: What is the easiest searching method to use?	48
Answer:	48
Cross Reference:	50
III.5: What is the quickest searching method to use?	50
Answer:	50
Cross Reference:	55
III.6: What is hashing?	55
Answer:	55
Cross Reference:	57
III.7: How can I sort a linked list?	57
Answer:	57
Cross Reference:	57

III.8: How can I search for data in a linked list?	57
Answer:	57
Cross Reference:	57
Sample Code	57
IV Data Files	63
IV.1: If <i>errno</i> contains a nonzero number, is there an error?	63
Answer:	63
Cross Reference:	64
IV.2: What is a stream?	64
Answer:	64
Cross Reference:	64
IV.3: How do you redirect a standard stream?	65
Answer:	65
Cross Reference:	65
IV.4: How can you restore a redirected standard stream?	65
Answer:	65
Cross Reference:	66
IV.5: Can <i>stdout</i> be forced to print somewhere other than the screen?	66
Answer:	66
Cross Reference:	67
IV.6: What is the difference between text and binary modes?	67
Answer:	67
Cross Reference:	67
IV.7: How do you determine whether to use a stream function or a low-level function?	68
Answer:	68
Cross Reference:	68
IV.8: How do you list files in a directory?	68
Answer:	68
Cross Reference:	69
IV.9: How do you list a file's date and time?	70
Answer:	70
Cross Reference:	72
IV.10: How do you sort filenames in a directory?	73
Answer:	73
Cross Reference:	74
IV.11: How do you determine a file's attributes?	75
Answer:	75
Cross Reference:	76
IV.12: How do you view the <i>PATH</i> ?	76
Answer:	76
Cross Reference:	77
IV.13: How can I open a file so that other programs can update it at the same time?	77
Answer:	77
Cross Reference:	79

IV.14: How can I make sure that my program is the only one accessing a file?	79
Answer:	79
Cross Reference:	79
IV.15: How can I prevent another program from modifying part of a file that I am modifying?	79
Answer:	79
Cross Reference:	80
IV.16: How can I have more than 20 files open at once?	81
Answer:	81
Cross Reference:	81
IV.17: How can I avoid the <i>Abort, Retry, Fail</i> messages?	81
Answer:	81
Cross Reference:	83
IV.18: How can I read and write comma-delimited text?	83
Answer:	83
Cross Reference:	85
V Working with the Preprocessor	87
V.1: What is a macro, and how do you use it?	88
Answer:	88
Cross Reference:	89
V.2: What will the preprocessor do for a program?	90
Answer:	90
Cross Reference:	92
V.3: How can you avoid including a header more than once?	92
Answer:	92
Cross Reference:	92
V.4: Can a file other than a .h file be included with <i>#include</i> ?	93
Answer:	93
Cross Reference:	93
V.5: What is the benefit of using <i>#define</i> to declare a constant?	93
Answer:	93
Cross Reference:	94
V.6: What is the benefit of using <i>enum</i> to declare a constant?	94
Answer:	94
Cross Reference:	94
V.7: What is the benefit of using an <i>enum</i> rather than a <i>#define</i> constant?	95
Answer:	95
Cross Reference:	96
V.8: How are portions of a program disabled in demo versions?	97
Answer:	97
Cross Reference:	97
V.9: When should you use a macro rather than a function?	97
Answer:	97
Cross Reference:	97

V.10: Is it better to use a macro or a function?.....	98
Answer:	98
Cross Reference:	98
V.11: What is the best way to comment out a section of code that contains comments?	98
Answer:	98
Cross Reference:	99
V.12: What is the difference between <code>#include <file2></code> and <code>#include "file"</code> ? ...	99
Answer:	99
Cross Reference:	99
V.13: Can you define which header file to include at compile time?	100
Answer:	100
Cross Reference:	100
V.14: Can include files be nested?	100
Answer:	100
Cross Reference:	100
V.15: How many levels deep can include files be nested?	101
Answer:	101
Cross Reference:	101
V.16: What is the concatenation operator?	101
Answer:	101
Cross Reference:	102
V.17: How can type-insensitive macros be created?	102
Answer:	102
Cross Reference:	103
V.18: What are the standard predefined macros?	103
Answer:	103
Cross Reference:	103
V.19: How can a program be made to print the line number where an error occurs?	104
Answer:	104
Cross Reference:	104
V.20: How can a program be made to print the name of a source file where an error occurs?	105
Answer:	105
Cross Reference:	105
V.21: How can you tell whether a program was compiled using C versus C++?	106
Answer:	106
Cross Reference:	106
V.22: What is a pragma?	106
Answer:	106
Cross Reference:	107
V.23: What is <code>#line</code> used for?	107
Answer:	107
Cross Reference:	108

V.24: What is the <code>_FILE_</code> preprocessor command?	108
Answer:	108
Cross Reference:	108
V.25: How can I print the name of the source file in a program?	108
Answer:	108
Cross Reference:	108
V.26: What is the <code>_LINE_</code> preprocessor command?	108
Answer:	108
Cross Reference:	109
V.27: How can I print the current line number of the source file in a program?	109
Answer:	109
Cross Reference:	109
V.28: What are the <code>_DATE_</code> and <code>_TIME_</code> preprocessor commands?	109
Answer:	109
Cross Reference:	110
V.29: How can I print the compile date and time in a program?	110
Answer:	110
Cross Reference:	110
V.30: How can you be sure that a program follows the ANSI C standard? ..	110
Answer:	110
Cross Reference:	111
V.31: How do you override a defined macro?	111
Answer:	111
Cross Reference:	112
V.32: How can you check to see whether a symbol is defined?	112
Answer:	112
Cross Reference:	112
V.33: What common macros are available?	112
Answer:	112
Cross Reference:	113
VI Working with Strings	115
VI.1: What is the difference between a string copy (<i>strcpy</i>) and a memory copy (<i>memcpy</i>)? When should each be used?	115
Answer:	115
Cross Reference:	117
VI.2: How can I remove the trailing spaces from a string?	117
Answer:	117
Cross Reference:	118
VI.3: How can I remove the leading spaces from a string?	118
Answer:	118
Cross Reference:	120
VI.4: How can I right-justify a string?	120
Answer:	120
Cross Reference:	122

VI.5: How can I pad a string to a known length?	122
Answer:	122
Cross Reference:	123
VI.6: How can I copy just a portion of a string?	123
Answer:	123
Cross Reference:	124
VI.7: How can I convert a number to a string?	124
Answer:	124
Cross Reference:	126
VI.8: How can I convert a string to a number?	126
Answer:	126
Cross Reference:	128
VI.9: How do you print only part of a string?	128
Answer:	128
Cross Reference:	129
VI.10: How do you remove spaces from the end of a string?	129
Answer:	129
Cross Reference:	129
VI.11: How can you tell whether two strings are the same?	129
Answer:	129
Cross Reference:	130
VII Pointers and Memory Allocation	131
VII.1: What is indirection?	133
Answer:	133
Cross Reference:	133
VII.2: How many levels of pointers can you have?	134
Answer:	134
Cross Reference:	135
VII.3: What is a null pointer?	135
Answer:	135
Cross Reference:	135
VII.4: When is a null pointer used?	136
Answer:	136
Using a Null Pointer to Stop Indirection or Recursion	136
Using a Null Pointer As an Error Value	137
Using a Null Pointer As a Sentinel Value	137
Cross Reference:	138
VII.5: What is a <i>void</i> pointer?	138
Answer:	138
Cross Reference:	138
VII.6: When is a <i>void</i> pointer used?	138
Answer:	138
Cross Reference:	139
VII.7: Can you subtract pointers from each other?	
Why would you?	139
Answer:	139
Cross Reference:	141

VII.8: When you add a value to a pointer, what is really added?	141
Answer:	141
Cross Reference:	142
VII.9: Is <i>NULL</i> always defined as 0?	142
Answer:	142
Cross Reference:	142
VII.10: Is <i>NULL</i> always equal to 0?	142
Answer:	142
Cross Reference:	142
VII.11: What does it mean when a pointer is used in an <i>if</i> statement?	143
Answer:	143
Cross Reference:	143
VII.12: Can you add pointers together? Why would you?	143
Answer:	143
Cross Reference	144
VII.13: How do you use a pointer to a function?	144
Answer:	144
Cross Reference:	145
VII.14: When would you use a pointer to a function?	145
Answer:	145
Cross Reference:	147
VII.15: Can the size of an array be declared at runtime?	147
Answer:	147
Cross Reference:	148
VII.16: Is it better to use <i>malloc()</i> or <i>calloc()</i> ?	149
Answer:	149
Cross Reference:	149
VII.17: How do you declare an array that will hold more than 64KB of data?	150
Answer:	150
Cross Reference:	150
VII.18: What is the difference between <i>far</i> and <i>near</i> ?	150
Answer:	150
Cross Reference:	151
VII.19: When should a <i>far</i> pointer be used?	151
Answer:	151
Cross Reference:	151
VII.20: What is the stack?	151
Answer:	151
Cross Reference:	152
VII.21: What is the heap?	152
Answer:	152
Cross Reference:	153
VII.22: What happens if you free a pointer twice?	153
Answer:	153
Cross Reference:	154

VII.23: What is the difference between <i>NULL</i> and <i>NUL</i> ?	155
Answer:	155
Cross Reference:	155
VII.24: What is a “null pointer assignment” error? What are bus errors, memory faults, and core dumps?	155
Answer:	155
Cross Reference:	156
VII.25: How can you determine the size of an allocated portion of memory?	156
Answer:	156
Cross Reference:	156
VII.26: How does <i>free()</i> know how much memory to release?	156
Answer:	156
Cross Reference:	157
VII.27: Can math operations be performed on a <i>void</i> pointer?	157
Answer:	157
Cross Reference:	157
VII.28: How do you print an address?	157
Answer:	157
Cross Reference:	158
VIII Functions.....	159
VIII.1: When should I declare a function?	159
Answer:	159
Cross Reference:	162
VIII.2: Why should I prototype a function?	162
Answer:	162
Cross Reference:	163
VIII.3: How many parameters should a function have?	163
Answer:	163
Cross Reference:	165
VIII.4: What is a static function?	165
Answer:	165
Cross Reference:	166
VIII.5: Should a function contain a <i>return</i> statement if it does not return a value?	166
Answer:	166
Cross Reference:	166
VIII.6: How can you pass an array to a function by value?	167
Answer:	167
Cross Reference:	169
VIII.7: Is it possible to execute code even after the program exits the <i>main()</i> function?	169
Answer:	169
Cross Reference:	170
VIII.8: What does a function declared as <i>PASCAL</i> do differently?	170
Answer:	170
Cross Reference:	171

VIII.9: Is using <i>exit()</i> the same as using <i>return</i> ?	171
Answer:	171
Cross Reference:	173
IX Arrays	175
IX.1: Do array subscripts always start with zero?	176
Answer:	176
Cross Reference:	177
IX.2: Is it valid to address one element beyond the end of an array?	177
Answer:	177
Cross Reference:	178
IX.3: Why worry about the addresses of the elements beyond the end of an array?	178
Answer:	178
Cross Reference:	179
IX.4: Can the <i>sizeof</i> operator be used to tell the size of an array passed to a function?	179
Answer:	179
Cross Reference:	180
IX.5: Is it better to use a pointer to navigate an array of values, or is it better to use a subscripted array name?	181
Answer:	181
Cross Reference:	183
IX.6: Can you assign a different address to an array tag?	183
Answer:	183
Cross Reference:	184
IX.7: What is the difference between <i>array_name</i> and <i>&array_name</i> ?	184
Answer:	184
Cross Reference:	184
IX.8: Why can't constant values be used to define an array's initial size?	185
Answer:	185
Cross Reference:	185
IX.9: What is the difference between a string and an array?	186
Answer:	186
Cross Reference:	187
X Bits and Bytes	189
X.1: What is the most efficient way to store flag values?	190
Answer:	190
Cross Reference:	191
X.2: What is meant by "bit masking"?	191
Answer:	191
Cross Reference:	194
X.3: Are bit fields portable?	194
Answer:	194
Cross Reference:	194

FREE EBOOKS, NOTES , VIDEOS & PLACEMENT MATERIAL



For All Companies placement
Material

@placementclasses



For CAT Exam Preparation
Material

@cat_classes



For GATE Exam Preparation
Material

@gate_classes



For Engineering Books &
Material

@cs_ebooks



Codes of Following Programming Languages



C

@c_examples



C++

@cpp_examples



Java

@java_examples0



Python

@python_examples

X.4: Is it better to bitshift a value than to multiply by 2?	194
Answer:	194
Cross Reference:	195
X.5: What is meant by high-order and low-order bytes?	195
Answer:	195
Cross Reference:	195
X.6: How are 16- and 32-bit numbers stored?	196
Answer:	196
Cross Reference:	196
XI Debugging	197
XI.1: My program hangs when I run it. What should I do?	197
Answer:	197
Infinite Loops	199
Taking Longer Than Expected to Execute	200
Waiting for Correct Input	202
Cross Reference:	203
XI.2: How can I detect memory leaks?	203
Answer:	203
Cross Reference:	204
XI.3: What is the best way to debug my program?	204
Answer:	204
What Tools Should Be Used to Debug a Program?	205
What Methods Can Be Used to Find Bugs in a Program?	207
How Can Bugs Be Avoided in the First Place?	208
Cross Reference:	211
XI.4: How can I debug a TSR program?	211
Answer:	211
Cross Reference:	212
XI.5: How do you get a program to tell you when (and where) a condition fails?	212
Answer:	212
Cross Reference:	213
XII Standard Library Functions	215
XII.1: Why should I use standard library functions instead of writing my own?	216
Answer:	216
Cross Reference:	216
XII.2: What header files do I need in order to define the standard library functions I use?	216
Answer:	216
Cross Reference:	223
XII.3: How can I write functions that take a variable number of arguments?	223
Answer:	223
Cross Reference:	225

XII.4: What is the difference between a free-standing and a hosted environment?	225
Answer:	225
Cross Reference:	225
XII.5: What standard functions are available to manipulate strings?	226
Answer:	226
Cross Reference:	229
XII.6: What standard functions are available to manipulate memory?	229
Answer:	229
Cross Reference:	231
XII.7: How do I determine whether a character is numeric, alphabetic, and so on?	232
Answer:	232
Cross Reference:	233
XII.8: What is a "locale"?	233
Answer:	233
Cross Reference:	233
XII.9: Is there a way to jump out of a function or functions?	233
Answer:	233
Cross Reference:	235
XII.10: What's a signal? What do I use signals for?	235
Answer:	235
Cross Reference:	236
XII.11: Why shouldn't I start variable names with underscores?	236
Answer:	236
Cross Reference:	236
XII.12: Why does my compiler provide two versions of <i>malloc()</i> ?	236
Answer:	236
Cross Reference:	239
XII.13: What math functions are available for integers? For floating point?	239
Answer:	239
Cross Reference:	240
XII.14: What are multibyte characters?	240
Answer:	240
Cross Reference:	241
XII.15: How can I manipulate strings of multibyte characters?	241
Answer:	241
Cross Reference:	241
XIII Times and Dates	243
XIII.1: How can I store a date in a single number? Are there any standards for this?	243
Answer:	243
Cross Reference:	247
XIII.2: How can I store time as a single integer? Are there any standards for this?	248
Answer:	248
Cross Reference:	251

XIII.3: Why are so many different time standards defined?	251
Answer:	251
Cross Reference:	251
XIII.4: What is the best way to store the date?	252
Answer:	252
Cross Reference:	252
XIII.5: What is the best way to store the time?	252
Answer:	252
Cross Reference:	253
XIV System Calls	255
XIV.1: How can environment variable values be retrieved?	256
Answer:	256
Cross Reference:	257
XIV.2: How can I call DOS functions from my program?	257
Answer:	257
Cross Reference:	258
XIV.3: How can I call BIOS functions from my program?	258
Answer:	258
Cross Reference:	260
XIV.4: How can I access important DOS memory locations from my program?	260
Answer:	260
Cross Reference:	262
XIV.5: What is BIOS?	262
Answer:	262
Cross Reference:	262
XIV.6: What are interrupts?	263
Answer:	263
Cross Reference:	264
XIV.7: Which method is better, ANSI functions or BIOS functions?	264
Answer:	264
Cross Reference:	265
XIV.8: Can you change to a VGA graphics mode using the BIOS?.....	265
Answer:	265
Cross Reference:	269
XIV.9: Does operator precedence always work (left to right, right to left)?...	269
Answer:	269
Cross Reference:	271
XIV.10: Should a variable's type be declared within the header of a function or immediately following? Why?	271
Answer:	271
Cross Reference:	271
XIV.11: Should programs always include a prototype for <i>main()</i> ?	271
Answer:	271
Cross Reference:	271

XIV.12: Should <i>main()</i> always return a value?	272
Answer:	272
Cross Reference:	272
XIV.13: Can I control the mouse using the BIOS?	272
Answer:	272
Cross Reference:	273
XV Portability	275
XV.1: Should C++ additions to a compiler be used in a C program?.....	277
Answer:	277
Cross Reference:	277
XV.2: What is the difference between C++ and C?	277
Answer:	277
Cross Reference:	279
XV.3: Is it valid to use <i>//</i> for comments in a C program?.....	279
Answer:	279
Cross Reference:	279
XV.4: How big is a <i>char</i> ? A <i>short</i> ? An <i>int</i> ? A <i>long</i> ?	279
Answer:	279
Cross Reference:	280
XV.5: What's the difference between big-endian and little-endian machines?	280
Answer:	280
Cross Reference:	281
XVI ANSI/ISO Standards	283
XVI.1: Does operator precedence always work?	284
Answer:	284
Cross Reference:	287
XVI.2: Should function arguments' types be declared in the argument list of a function or immediately following?.....	288
Answer:	288
Cross Reference:	289
XVI.3: Should programs include a prototype for <i>main()</i> ?.....	289
Answer:	289
Cross Reference:	290
XVI.4: Should <i>main()</i> always return a value?	290
Answer:	290
Cross Reference:	291
XVII User Interface—Screen and Keyboard.....	293
XVII.1: Why don't I see my screen output until the program ends?	293
Answer:	293
Cross Reference:	294
XVII.2: How do I position the cursor on the screen?	294
Answer:	294
Cross Reference:	295

XVII.3: What is the easiest way to write data to the screen?	295
Answer:	295
Cross Reference:	296
XVII.4: What is the fastest way to write text to the screen?	296
Answer:	296
Choosing Print Functions with a Lower Overhead	297
Using a Package or Library with Faster Print Features	297
Bypassing the Operating System and Writing Directly to the Screen	297
Cross Reference:	300
XVII.5: How can I prevent the user from breaking my program with <i>Ctrl-Break</i> ?	300
Answer:	300
Cross Reference:	301
XVII.6: How can you get data of only a certain type, for example, only characters?	302
Answer:	302
Cross Reference:	302
XVII.7: Why shouldn't <i>scanf</i> be used to accept data?	302
Answer:	302
Cross Reference:	303
XVII.8: How do I use function keys and arrow keys in my programs?	304
Answer:	304
Cross Reference:	305
XVII.9: How do I prevent the user from typing too many characters in a field?	305
Answer:	305
Cross Reference:	307
XVII.10: How do you zero-pad a number?	307
Answer:	307
Cross Reference:	307
XVII.11: How do you print a dollars-and-cents value?	307
Answer:	307
Cross Reference:	309
XVII.12: How do I print a number in scientific notation?	310
Answer:	310
Cross Reference:	310
XVII.13: What is the ANSI driver?	310
Answer:	310
Cross Reference:	311
XVII.14: How do you clear the screen with the ANSI driver?	311
Answer:	311
Cross Reference:	311
XVII.15: How do you save the cursor's position with the ANSI driver?	311
Answer:	311
Cross Reference:	312

XVII.16: How do you restore the cursor's position with the ANSI driver?...	312
Answer:	312
Cross Reference:	312
XVII.17: How do you change the screen color with the ANSI driver?	312
Answer:	312
Cross Reference:	312
XVII.18: How do you write text in color with the ANSI driver?.....	312
Answer:	312
Cross Reference:	313
XVII.19: How do I move the cursor with the ANSI driver?	313
Answer:	313
Cross Reference:	314
XVIII Writing and Compiling Your Programs.....	315
XVIII.1: Should my program be written in one source file or several source files?	316
Answer:	316
Cross Reference:	316
XVIII.2: What are the differences between the memory models?	317
Answer:	317
Cross Reference:	317
XVIII.3: What are the most commonly used memory models?	317
Answer:	317
Cross Reference:	318
XVIII.4: Which memory model should be used?	318
Answer:	318
Cross Reference:	319
XVIII.5: How do you create a .COM file?	319
Answer:	319
Cross Reference:	319
XVIII.6: What is the benefit of a .COM file over an .EXE file?	319
Answer:	319
Cross Reference:	319
XVIII.7: Are all the functions in a library added to an .EXE file when the library is linked to the objects?	320
Answer:	320
Cross Reference:	321
XVIII.8: Can multiple library functions be included in the same source file?	321
Answer:	321
Cross Reference:	321
XVIII.9: Why should I create a library?	321
Answer:	321
Cross Reference:	322
XVIII.10: My program has several files in it. How do I keep them all straight?.....	322
Answer:	322
Cross Reference:	323

XVIII.11: I get the message <i>DGROUP: group exceeds 64K</i> during my link. What's wrong?	323
Answer:	323
Cross Reference:	323
XVIII.12: How can I keep my program from running out of memory?	324
Answer:	324
Cross Reference:	324
XVIII.13: My program is too big to run under DOS. How can I make it fit?	324
Answer:	324
Cross Reference:	325
XVIII.14: How can I get more than 640KB of memory available to my DOS program?	325
Answer:	325
Cross Reference:	326
XVIII.15: What is the difference between near and far?	327
Answer:	327
Cross Reference:	329
XIX Programming Style and Standards	331
XIX.1: Should the underscore be used in variable names?	332
Answer:	332
Cross Reference:	332
XIX.2: Can a variable's name be used to indicate its data type?	332
Answer:	332
Cross Reference:	333
XIX.3: Does the use of comments affect program speed, executable size, or efficiency?	333
Answer:	333
Cross Reference:	334
XIX.4: Does the use of white space affect program speed, executable size, or efficiency?	334
Answer:	334
Cross Reference:	336
XIX.5: What is camel notation?	336
Answer:	336
Cross Reference:	336
XIX.6: Do longer variable names affect the speed, executable size, or efficiency of a program?	336
Answer:	336
Cross Reference:	337
XIX.7: What is the correct way to name a function?	337
Answer:	337
Cross Reference:	338
XIX.8: What is the correct way to use braces?	338
Answer:	338
Cross Reference:	339

XIX.9: How many letters long should variable names be? What is the ANSI standard for significance?	339
Answer:	339
Cross Reference:	340
XIX.10: What is Hungarian notation, and should I use it?	340
Answer:	340
Cross Reference:	340
XIX.11: What is iterative processing?.....	341
Answer:	341
Cross Reference:	342
XIX.12: What is recursion, and how do you use it?	342
Answer:	342
Cross Reference:	344
XIX.13: What is the best way to represent true and false in C?	344
Answer:	344
Cross Reference:	345
XIX.14: What is the difference between a null loop and an infinite loop?	345
Answer:	345
Cross Reference:	346
XIX.15: What is the difference between <i>continue</i> and <i>break</i> ?.....	346
Answer:	346
Cross Reference:	347
XX Miscellaneous	349
XX.1: How are command-line parameters obtained?	349
Answer:	349
Cross Reference:	351
XX.2: Should programs always assume that command-line parameters can be used?	351
Answer:	351
Cross Reference:	352
XX.3: What is the difference between “exception handling” and “structured exception handling”?.....	352
Answer:	352
Cross Reference:	352
XX.4: How do you create a delay timer in a DOS program?	353
Answer:	353
Cross Reference:	353
XX.5: Who are Kernighan and Ritchie?	353
Answer:	353
Cross Reference:	354
XX.6: How do you create random numbers?	354
Answer:	354
Cross Reference:	356
XX.7: When should a 32-bit compiler be used?	356
Answer:	356
Cross Reference:	357

XX.8: How do you interrupt a Windows program?	357
Answer:	357
Cross Reference:	360
XX.9: Why should I use static variables?	360
Answer:	360
Cross Reference:	360
XX.10: How can I run another program after mine?	361
Answer:	361
Cross Reference:	362
XX.11: How can I run another program during my program's execution? ...	362
Answer:	362
Cross Reference:	363
XX.12: How can I pass data from one program to another?	363
Answer:	363
Cross Reference:	368
XX.13: How can I determine which directory my program is running from?	368
Answer:	368
Cross Reference:	369
XX.14: How can I locate my program's important files (databases, configuration files, and such)?	369
Answer:	369
Cross Reference:	370
XX.15: Some of your examples are not very efficient. Why did you write them so badly?	370
Answer:	370
Cross Reference:	370
XX.16: How do you disable Ctrl-Break?	370
Answer:	370
Cross Reference:	372
XX.17: Can you disable warm boots (Ctrl-Alt-Delete)?	372
Answer:	372
Cross Reference:	374
XX.18: How do you tell whether a character is a letter of the alphabet?	374
Answer:	374
Cross Reference:	375
XX.19: How do you tell whether a character is a number?	375
Answer:	375
Cross Reference:	376
XX.20: How do you assign a hexadecimal value to a variable?	376
Answer:	376
Cross Reference:	376
XX.21: How do you assign an octal value to a number?	377
Answer:	377
Cross Reference:	377

XX.22: What is binary?.....	377
Answer:	377
Cross Reference:	379
XX.23: What is octal?	379
Answer:	379
Cross Reference:	379
XX.24: What is hexadecimal?	380
Answer:	380
Cross Reference:	381
XX.25: What are escape characters?	382
Answer:	382
Cross Reference:	383
XXI Windows	385
XXI.1: Can <i>printf()</i> be used in a Windows program?	386
Answer:	386
Cross Reference:	386
XXI.2: How do you create a delay timer in a Windows program?	387
Answer:	387
Cross Reference:	387
XXI.3: What is a handle?	387
Answer:	387
Cross Reference:	388
XXI.4: How do you interrupt a Windows program?	388
Answer:	388
Cross Reference:	389
XXI.5: What is the GDI and how is it accessed?	389
Answer:	389
Cross Reference:	390
XXI.6: Why is windows.h important?	390
Answer:	390
Cross Reference:	391
XXI.7: What is the Windows SDK?	391
Answer:	391
Cross Reference:	391
XXI.8: Do you need Microsoft's Windows SDK to write Windows programs?.....	392
Answer:	392
Cross Reference:	392
XXI.9: What is the difference between Windows functions and standard DOS functions?	392
Answer:	392
Cross Reference:	393
XXI.10: What is dynamic linking?	393
Answer:	393
Cross Reference:	393

XXI.11: What are the differences among <i>HANDLE</i> , <i>HWND</i> , and <i>HDC</i> ? ...	394
Answer:	394
Cross Reference:	394
XXI.12: Are Windows programs compatible from one compiler to the next?	394
Answer:	394
Cross Reference:	395
XXI.13: Will Windows always save and refresh your program's windows? ...	395
Answer:	395
Cross Reference:	396
XXI.14: How do you determine a Windows program's client area size?	396
Answer:	396
Cross Reference:	397
XXI.15: What are OEM key codes?	397
Answer:	397
Cross Reference:	397
XXI.16: Should a Windows program care about the OEM key codes?	397
Answer:	397
Cross Reference:	398
XXI.17: What are virtual key codes?	398
Answer:	398
Cross Reference:	400
XXI.18: What is a dead key?	400
Answer:	400
Cross Reference:	401
XXI.19: What is the difference between the caret and the cursor?	401
Answer:	401
Cross Reference:	402
XXI.20: Can a mouse click be captured in an area outside your program's client area?	402
Answer:	402
Cross Reference:	403
XXI.21: How do you create an animated bitmap?	403
Answer:	403
Cross Reference:	403
XXI.22: How do you get the date and time in a Windows program?	404
Answer:	404
Cross Reference:	404
XXI.23: How do you update the title bar in a Windows program?	405
Answer:	405
Cross Reference:	405
XXI.24: How do you access the system colors in a Windows program?	405
Answer:	405
Cross Reference:	406

XXI.25: What are the system color constants?	406
Answer:	406
Cross Reference:	407
XXI.26: How do you create your own buttons or controls?	407
Answer:	407
Cross Reference:	408
XXI.27: What is a static child window?	408
Answer:	408
Cross Reference:	408
XXI.28: What is window subclassing?	409
Answer:	409
Cross Reference:	409
XXI.29: What is the edit class?	410
Answer:	410
Cross Reference:	411
XXI.30: What is the listbox class?	411
Answer:	411
Cross Reference:	412
XXI.31: How is memory organized in Windows?	412
Answer:	412
Cross Reference:	413
XXI.32: How is memory allocated in a Windows program?	413
Answer:	413
Cross Reference:	414
XXI.33: What is the difference between modal and modeless dialog boxes?	414
Answer:	414
Cross Reference:	414
Index	415

Introduction

What is a *FAQ* ? It's a Frequently Asked Question. You can see FAQs just about everywhere in the online community. They originated in the USENET groups on the Internet as a way to answer users' most common questions regarding the groups. The FAQs files were efficient. Instead of answering the same questions over and over, one file was written that contained all the frequently asked questions and answers to those questions.

This book is a comprehensive list of FAQs, assembled by a group of professional C programmers. This book contains the FAQs most often posed by our readers over our many years of publishing programming books. You won't find a FAQ list that goes into as much detail as this book does. (Have you ever seen a 400-page FAQ list?)

Our team of expert programmers has tackled the toughest topics, including variables and data storage, sorting data, pointers and memory allocation, tables and arrays, debugging, portability, ANSI standards, and Windows concerns. If something in C is stumping you, odds are you'll find an answer in this book. In addition, the extensive cross referencing in this book will help you find the answer you need—even if you start out looking in the wrong spot.