

standard header (`<stddef.h>`). A similar change occurs in the type (`ptrdiff_t`) of the difference between pointers. See §A7.4.8 and §A7.7.

- The address-of operator `&` may not be applied to an object declared `register`, even if the implementation chooses not to keep the object in a register.
- The type of a shift expression is that of the left operand; the right operand can't promote the result. See §A7.8.
- The Standard legalizes the creation of a pointer just beyond the end of an array, and allows arithmetic and relations on it; see §A7.7.
- The Standard introduces (borrowing from C++) the notion of a function prototype declaration that incorporates the types of the parameters, and includes an explicit recognition of variadic functions together with an approved way of dealing with them. See §§A7.3.2, A8.6.3, B7. The older style is still accepted, with restrictions.
- Empty declarations, which have no declarators and don't declare at least a structure, union, or enumeration, are forbidden by the Standard. On the other hand, a declaration with just a structure or union tag redeclares that tag even if it was declared in an outer scope.
- External data declarations without any specifiers or qualifiers (just a naked declarator) are forbidden.
- Some implementations, when presented with an `extern` declaration in an inner block, would export the declaration to the rest of the file. The Standard makes it clear that the scope of such a declaration is just the block.
- The scope of parameters is injected into a function's compound statement, so that variable declarations at the top level of the function cannot hide the parameters.
- The name spaces of identifiers are somewhat different. The Standard puts all tags in a single name space, and also introduces a separate name space for labels; see §A11.1. Also, member names are associated with the structure or union of which they are a part. (This has been common practice from some time.)
- Unions may be initialized; the initializer refers to the first member.
- Automatic structures, unions, and arrays may be initialized, albeit in a restricted way.
- Character arrays with an explicit size may be initialized by a string literal with exactly that many characters (the `\0` is quietly squeezed out).
- The controlling expression, and the case labels, of a switch may have any integral type.



# Index

- 0... octal constant 37, 193
- 0x... hexadecimal constant 37, 193
- + addition operator 41, 205
- & address operator 93, 203
- = assignment operator 17, 42, 208
- += assignment operator 50
- \ backslash character 8, 38
- & bitwise AND operator 48, 207
- ^ bitwise exclusive OR operator 48, 207
- | bitwise inclusive OR operator 48, 207
- , comma operator 62, 209
- ?: conditional expression 51, 208
- ... declaration 155, 202
- decrement operator 18, 46, 106, 203
- / division operator 10, 41, 205
- == equality operator 19, 41, 207
- >= greater or equal operator 41, 206
- > greater than operator 41, 206
- ++ increment operator 18, 46, 106, 203
- \* indirection operator 94, 203
- != inequality operator 16, 41, 207
- << left shift operator 49, 206
- <= less or equal operator 41, 206
- < less than operator 41, 206
- && logical AND operator 21, 41, 49, 207
- ! logical negation operator 42, 203–204
- || logical OR operator 21, 41, 49, 208
- % modulus operator 41, 205
- \* multiplication operator 41, 205
- ~ one's complement operator 49, 203–204
- # preprocessor operator 90, 230
- ## preprocessor operator 90, 230
- ' quote character 19, 37–38, 193
- " quote character 8, 20, 38, 194
- >> right shift operator 49, 206
- . structure member operator 128, 201
- > structure pointer operator 131, 201
- subtraction operator 41, 205
- unary minus operator 203–204
- + unary plus operator 203–204
- \_ underscore character 35, 192, 241
- \0 null character 30, 38, 193
- 
- \a alert character 38, 193
- abort library function 252
- 
- abs library function 253
- abstract declarator 220
- access mode, file 160, 178, 242
- acos library function 251
- actual argument *see* argument
- addition operator, + 41, 205
- additive operators 205
- addpoint function 130
- address arithmetic *see* pointer arithmetic
- address of register 210
- address of variable 28, 94, 203
- address operator, & 93, 203
- addtree function 141
- afree function 102
- alert character, \a 38, 193
- alignment, bit-field 150, 213
- alignment by union 186
- alignment restriction 138, 142, 148, 167, 185, 199
- alloc function 101
- allocator, storage 142, 185–189
- ambiguity, if-else 56, 223, 234
- American National Standards Institute (ANSI) ix, 2, 191
- a.out 6, 70
- argc argument count 114
- argument, definition of 25, 201
- argument, function 25, 202
- argument list, variable length 155, 174, 202, 218, 225, 254
- argument list, void 33, 73, 218, 225
- argument, pointer 100
- argument promotion 45, 202
- argument, subarray 100
- arguments, command-line 114–118
- argv argument vector 114, 163
- arithmetic conversions, usual 42, 198
- arithmetic operators 41
- arithmetic, pointer 94, 98, 100–103, 117, 138, 205
- arithmetic types 196
- array, character 20, 28, 104
- array declaration 22, 111, 216
- array declarator 216
- array initialization 86, 113, 219
- array, initialization of two-dimensional 112, 220

- array, multi-dimensional 110, 217
- array name argument 28, 100, 112
- array name, conversion of 99, 200
- array of pointers 107
- array reference 201
- array size, default 86, 113, 133
- array, storage order of 112, 217
- array subscripting 22, 97, 201, 217
- array, two-dimensional 110, 112, 220
- array vs. pointer 97, 99–100, 104, 113
- arrays of structures 132
- ASCII character set 19, 37, 43, 229, 249
- asctime library function 256
- asin library function 251
- asm keyword 192
- `<assert.h>` header 253
- assignment, conversion by 44, 208
- assignment expression 17, 21, 51, 208
- assignment, multiple 21
- assignment operator, = 17, 42, 208
- assignment operator, += 50
- assignment operators 42, 50, 208
- assignment statement, nested 17, 21, 51
- assignment suppression, `scanf` 157, 245
- associativity of operators 52, 200
- atan, atan2 library functions 251
- atexit library function 253
- atof function 71
- atof library function 251
- atoi function 43, 61, 73
- atoi library function 251
- atol library function 251
- auto storage class specifier 210
- automatic storage class 31, 195
- automatic variable 31, 74, 195
- automatics, initialization of 31, 40, 85, 219
- automatics, scope of 80, 228
- avoiding goto 66
- `\b` backspace character 8, 38, 193
- backslash character, `\\` 8, 38
- bell character *see* alert character
- binary stream 160, 241–242
- binary tree 139
- binsearch function 58, 134, 137
- bit manipulation idioms 49, 149
- bitcount function 50
- bit-field alignment 150, 213
- bit-field declaration 150, 212
- bitwise AND operator, & 48, 207
- bitwise exclusive OR operator, ^ 48, 207
- bitwise inclusive OR operator, | 48, 207
- bitwise operators 48, 207
- block *see* compound statement
- block, initialization in 84, 223
- block structure 55, 84, 223
- boundary condition 19, 65
- braces 7, 10, 55, 84
- braces, position of 10
- break statement 59, 64, 224
- bsearch library function 253
- buffered `getchar` 172
- buffered input 170
- buffering *see* `setbuf`, `setvbuf`
- BUFSIZ 243
- calculator program 72, 74, 76, 158
- call by reference 27
- call by value 27, 95, 202
- calloc library function 167, 252
- canonrect function 131
- carriage return character, `\r` 38, 193
- case label 58, 222
- cast, conversion by 45, 198–199, 205
- cast operator 45, 142, 167, 198, 205, 220
- cat program 160, 162–163
- cc command 6, 70
- ceil library function 251
- char type 10, 36, 195, 211
- character array 20, 28, 104
- character constant 19, 37, 193
- character constant, octal 37
- character constant, wide 193
- character count program 18
- character input/output 15, 151
- character set 229
- character set, ASCII 19, 37, 43, 229, 249
- character set, EBCDIC 43
- character set, ISO 229
- character, signed 44, 195
- character string *see* string constant
- character testing functions 166, 248
- character, unsigned 44, 195
- character-integer conversion 23, 42, 197
- characters, white space 157, 166, 245, 249
- clearerr library function 248
- CLOCKS\_PER\_SEC 255
- clock library function 255
- clock\_t type name 255
- close system call 174
- closedir function 184
- coercion *see* cast
- comma operator, , 62, 209
- command-line arguments 114–118
- comment 9, 191–192, 229
- comparison, pointer 102, 138, 187, 207
- compilation, separate 67, 80, 227
- compiling a C program 6, 25
- compiling multiple files 70
- compound statement 55, 84, 222, 225–226
- concatenation, string 38, 90, 194
- concatenation, token 90, 230
- conditional compilation 91, 231
- conditional expression, ?: 51, 208
- const qualifier 40, 196, 211
- constant expression 38, 58, 91, 209
- constant, manifest 230
- constant suffix 37, 193
- constant, type of 37, 193
- constants 37, 192
- continue statement 65, 224
- control character 249
- control line 88, 229–233
- conversion 197–199
- conversion by assignment 44, 208
- conversion by cast 45, 198–199, 205

- conversion by `return` 73, 225
- conversion, character-integer 23, 42, 197
- conversion, double-float 45, 198
- conversion, float-double 44, 198
- conversion, floating-integer 45, 197
- conversion, integer-character 45
- conversion, integer-floating 12, 197
- conversion, integer-pointer 199, 205
- conversion of array name 99, 200
- conversion of function 200
- conversion operator, explicit *see* `cast`
- conversion, pointer 142, 198, 205
- conversion, pointer-integer 198–199, 205
- conversions, usual arithmetic 42, 198
- `copy` function 29, 33
- `cos` library function 251
- `cosh` library function 251
- `creat` system call 172
- `CRLF` 151, 241
- `ctime` library function 256
- `<ctype.h>` header 43, 248
  
- date conversion 111
- `day_of_year` function 111
- `dcl` function 123
- `dcl` program 125
- declaration 9, 40, 210–218
- declaration, array 22, 111, 216
- declaration, bit-field 150, 212
- declaration, external 225–226
- declaration of external variable 31, 225
- declaration of function 217–218
- declaration of function, implicit 27, 72, 201
- declaration of pointer 94, 100, 216
- declaration, storage class 210
- declaration, structure 128, 212
- declaration, type 216
- declaration, `typedef` 146, 210, 221
- declaration, union 147, 212
- declaration vs. definition 33, 80, 210
- declarator 215–218
- declarator, abstract 220
- declarator, array 216
- declarator, function 217
- decrement operator, `--` 18, 46, 106, 203
- default array size 86, 113, 133
- default function type 30, 201
- default initialization 86, 219
- default label 58, 222
- defensive programming 57, 59
- `#define` 14, 89, 229
- `#define`, multi-line 89
- `#define` vs. `enum` 39, 149
- `#define` with arguments 89
- defined preprocessor operator 91, 232
- definition, function 25, 69, 225
- definition, macro 229
- definition of argument 25, 201
- definition of external variable 33, 227
- definition of parameter 25, 201
- definition of storage 210
- definition, removal of *see* `#undef`
- definition, tentative 227
  
- dereference *see* indirection
- derived types 1, 10, 196
- descriptor, file 170
- designator, function 201
- `difftime` library function 256
- `DIR` structure 180
- `dirdcl` function 124
- directory list program 179
- `Dirent` structure 180
- `dir.h` include file 183
- `dirwalk` function 182
- `div` library function 253
- division, integer 10, 41
- division operator, `/` 10, 41, 205
- `div_t`, `ldiv_t` type names 253
- `do` statement 63, 224
- do-nothing function 70
- double constant 37, 194
- double type 10, 18, 36, 196, 211
- double-float conversion 45, 198
  
- E notation 37, 194
- EBCDIC character set 43
- echo program 115–116
- EDOM 250
- efficiency 51, 83, 88, 142, 187
- `else` *see* `if-else` statement
- `#else`, `#elif` 91, 232
- `else-if` 23, 57
- empty function 70
- empty statement *see* `null` statement
- empty string 38
- end of file *see* EOF
- `#endif` 91
- `enum` specifier 39, 215
- `enum` vs. `#define` 39, 149
- enumeration constant 39, 91, 193–194, 215
- enumeration tag 215
- enumeration type 196
- enumerator 194, 215
- EOF 16, 151, 242
- equality operator, `==` 19, 41, 207
- equality operators 41, 207
- equivalence, type 221
- ERANGE 250
- `errno` 248, 250
- `<errno.h>` header 248
- `#error` 233
- error function 174
- errors, input/output 164, 248
- escape sequence 8, 19, 37–38, 193, 229
- escape sequence, `\x` hexadecimal 37, 193
- escape sequences, table of 38, 193
- evaluation, order of 21, 49, 53, 63, 77, 90, 95, 200
- exceptions 200, 255
- `exit` library function 163, 252
- `EXIT_FAILURE`, `EXIT_SUCCESS` 252
- `exp` library function 251
- expansion, macro 230
- explicit conversion operator *see* `cast`
- exponentiation 24, 251
- expression 200–209

- expression, assignment 17, 21, 51, 208
- expression, constant 38, 58, 91, 209
- expression order of evaluation 52, 200
- expression, parenthesized 201
- expression, primary 200
- expression statement 55, 57, 222
- extern storage class specifier 31, 33, 80, 210
- external declaration 225–226
- external linkage 73, 192, 195, 211, 228
- external names, length of 35, 192
- external static variables 83
- external variable 31, 73, 195
- external variable, declaration of 31, 225
- external variable, definition of 33, 227
- externals, initialization of 40, 81, 85, 219
- externals, scope of 80, 228
- 
- \f formfeed character 38, 193
- fabs library function 251
- fclose library function 162, 242
- fcntl.h include file 172
- feof library function 164, 248
- feof macro 176
- ferror library function 164, 248
- ferror macro 176
- fflush library function 242
- fgetc library function 246
- fgetpos library function 248
- fgets function 165
- fgets library function 164, 247
- field *see* bit-field
- file access 160, 169, 178, 242
- file access mode 160, 178, 242
- file appending 160, 175, 242
- file concatenation program 160
- file copy program 16–17, 171, 173
- file creation 161, 169
- file descriptor 170
- file inclusion 88, 231
- file opening 160, 169, 172
- file permissions 173
- file pointer 160, 175, 242
- \_\_FILE\_\_ preprocessor name 254
- FILE type name 160
- filecopy function 162
- filename suffix, .h 33
- FILENAME\_MAX 242
- \_fillbuf function 178
- float constant 37, 194
- float type 9, 36, 196, 211
- float-double conversion 44, 198
- <float.h> header 36, 257
- floating constant 12, 37, 194
- floating point, truncation of 45, 197
- floating types 196
- floating-integer conversion 45, 197
- floor library function 251
- fmod library function 251
- fopen function 177
- fopen library function 160, 242
- FOPEN\_MAX 242
- for(;;) infinite loop 60, 89
- for statement 13, 18, 60, 224
- 
- for vs. while 14, 60
- formal parameter *see* parameter
- formatted input *see* scanf
- formatted output *see* printf
- formfeed character, \f 38, 193
- fortran keyword 192
- fpos\_t type name 248
- fprintf library function 161, 243
- fputc library function 247
- fputs function 165
- fputs library function 164, 247
- fread library function 247
- free function 188
- free library function 167, 252
- freopen library function 162, 242
- frexp library function 251
- fscanf library function 161, 245
- fseek library function 248
- fsetpos library function 248
- fsize function 182
- fsize program 181
- fstat system call 183
- ftell library function 248
- function argument 25, 202
- function argument conversion *see* argument promotion
- function call semantics 201
- function call syntax 201
- function, conversion of 200
- function, declaration of 217–218
- function declaration, static 83
- function declarator 217
- function definition 25, 69, 225
- function designator 201
- function, implicit declaration of 27, 72, 201
- function names, length of 35, 192
- function, new-style 202
- function, old-style 26, 33, 72, 202
- function, pointer to 118, 147, 201
- function prototype 26, 30, 45, 72, 120, 202
- function type, default 30, 201
- functions, character testing 166, 248
- fundamental types 9, 36, 195
- fwrite library function 247
- 
- generic pointer *see* void \* pointer
- getbits function 49
- getc library function 161, 247
- getc macro 176
- getch function 79
- getchar, buffered 172
- getchar library function 15, 151, 161, 247
- getchar, unbuffered 171
- getenv library function 253
- getint function 97
- getline function 29, 32, 69, 165
- getop function 78
- gets library function 164, 247
- gettoken function 125
- getword function 136
- gmtime library function 256
- goto statement 65, 224
- greater or equal operator, >= 41, 206

greater than operator, > 41, 206

.h filename suffix 33  
 hash function 144  
 hash table 144  
 header file 33, 82  
 headers, table of standard 241  
 hexadecimal constant, 0x... 37, 193  
 hexadecimal escape sequence, \x 37, 193  
 Hoare, C. A. R. 87  
 HUGE\_VAL 250

identifier 192  
 #if 91, 135, 231  
 #ifdef 91, 232  
 if-else ambiguity 56, 223, 234  
 if-else statement 19, 21, 55, 223  
 #ifndef 91, 232  
 illegal pointer arithmetic 102–103, 138, 205  
 implicit declaration of function 27, 72, 201  
 #include 33, 88, 152, 231  
 incomplete type 212  
 inconsistent type declaration 72  
 increment operator, ++ 18, 46, 106, 203  
 indentation 10, 19, 23, 56  
 indirection operator, \* 94, 203  
 inequality operator, != 16, 41, 207  
 infinite loop, for(;;) 60, 89  
 information hiding 67–68, 75, 77  
 initialization 40, 85, 218  
 initialization, array 86, 113, 219  
 initialization by string constant 86, 219  
 initialization, default 86, 219  
 initialization in block 84, 223  
 initialization of automatics 31, 40, 85, 219  
 initialization of externals 40, 81, 85, 219  
 initialization of statics 40, 85, 219  
 initialization of structure arrays 133  
 initialization of two-dimensional array 112, 220  
 initialization, pointer 102, 138  
 initialization, structure 128, 219  
 initialization, union 219  
 initializer 227  
 initializer, form of 85, 209  
 inode 179  
 input, buffered 170  
 input, formatted *see* scanf  
 input, keyboard 15, 151, 170  
 input pushback 78  
 input, unbuffered 170  
 input/output, character 15, 151  
 input/output errors 164, 248  
 input/output redirection 152, 161, 170  
 install function 145  
 int type 9, 36, 211  
 integer constant 12, 37, 193  
 integer-character conversion 45  
 integer-floating conversion 12, 197  
 integer-pointer conversion 199, 205  
 integral promotion 44, 197  
 integral types 196

internal linkage 195, 228  
 internal names, length of 35, 192  
 internal static variables 83  
 \_IOFBF, \_IOLBF, \_IONBF 243  
 isalnum library function 136, 249  
 isalpha library function 136, 166, 249  
 iscntrl library function 249  
 isdigit library function 166, 249  
 isgraph library function 249  
 islower library function 166, 249  
 ISO character set 229  
 isprint library function 249  
 ispunct library function 249  
 isspace library function 136, 166, 249  
 isupper library function 166, 249  
 isxdigit library function 249  
 iteration statements 224  
 itoa function 64

jump statements 224

keyboard input 15, 151, 170  
 keyword count program 133  
 keywords, list of 192

label 65, 222  
 label, case 58, 222  
 label, default 58, 222  
 label, scope of 66, 222, 228  
 labeled statement 65, 222  
 labs library function 253  
 %ld conversion 18  
 ldexp library function 251  
 lddiv library function 253  
 leap year computation 41, 111  
 left shift operator, << 49, 206  
 length of names 35, 192  
 length of string 30, 38, 104  
 length of variable names 192  
 less or equal operator, <= 41, 206  
 less than operator, < 41, 206  
 lexical conventions 191  
 lexical scope 227  
 lexicographic sorting 118  
 library function 7, 67, 80  
 <limits.h> header 36, 257  
 #line 233  
 line count program 19  
 \_\_LINE\_\_ preprocessor name 254  
 line splicing 229  
 linkage 195, 227–228  
 linkage, external 73, 192, 195, 211, 228  
 linkage, internal 195, 228  
 list directory program 179  
 list of keywords 192  
 locale issues 241  
 <locale.h> header 241  
 localtime library function 256  
 log, log10 library functions 251  
 logical AND operator, && 21, 41, 49, 207  
 logical expression, numeric value of 44

- logical negation operator, ! 42, 203–204
- logical OR operator, || 21, 41, 49, 208
- long constant 37, 193
- long double constant 37, 194
- long double type 36, 196
- long type 10, 18, 36, 196, 211
- longest-line program 29, 32
- longjmp library function 254
- LONG\_MAX, LONG\_MIN 252
- lookup function 145
- loop *see* while, for, do
- lower case conversion program 153
- lower function 43
- ls command 179
- lseek system call 174
- lvalue 197
  
- macro preprocessor 88, 228–233
- macros with arguments 89
- magic numbers 14
- main function 6
- main, return from 26, 164
- makepoint function 130
- malloc function 187
- malloc library function 143, 167, 252
- manifest constant 230
- <math.h> header 44, 250
- member name, structure 128, 213
- memchr library function 250
- memcmp library function 250
- memcpy library function 250
- memmove library function 250
- memset library function 250
- missing storage class specifier 211
- missing type specifier 211
- mktime library function 256
- modf library function 251
- modularization 24, 28, 34, 67, 74–75, 108
- modulus operator, % 41, 205
- month\_day function 111
- month\_name function 113
- morecore function 188
- multi-dimensional array 110, 217
- multiple assignment 21
- multiple files, compiling 70
- multiplication operator, \* 41, 205
- multiplicative operators 205
- multi-way decision 23, 57
- mutually recursive structures 140, 213
  
- \n newline character 7, 15, 20, 37–38, 193, 241
- name 192
- name hiding 84
- name space 227
- names, length of 35, 192
- negative subscripts 100
- nested assignment statement 17, 21, 51
- nested structure 129
- newline 191, 229
- newline character, \n 7, 15, 20, 37–38, 193, 241
- new-style function 202
- NULL 102
- null character, \0 30, 38, 193
- null pointer 102, 198
- null statement 18, 222
- null string 38
- numbers, size of 9, 18, 36, 257
- numcmp function 121
- numeric sorting 118
- numeric value of logical expression 44
- numeric value of relational expression 42, 44
  
- object 195, 197
- octal character constant 37
- octal constant, 0... 37, 193
- old-style function 26, 33, 72, 202
- one's complement operator, ~ 49, 203–204
- open system call 172
- opendir function 183
- operations on unions 148
- operations permitted on pointers 103
- operators, additive 205
- operators, arithmetic 41
- operators, assignment 42, 50, 208
- operators, associativity of 52, 200
- operators, bitwise 48, 207
- operators, equality 41, 207
- operators, multiplicative 205
- operators, precedence of 17, 52, 95, 131–132, 200
- operators, relational 16, 41, 206
- operators, shift 48, 206
- operators, table of 53
- order of evaluation 21, 49, 53, 63, 77, 90, 95, 200
- order of translation 228
- O\_RDONLY, O\_RDWR, O\_WRONLY 172
- output, formatted *see* printf
- output redirection 152
- output, screen 15, 152, 163, 170
- overflow 41, 200, 250, 255
  
- parameter 84, 99, 202
- parameter, definition of 25, 201
- parenthesized expression 201
- parse tree 123
- parser, recursive-descent 123
- pattern finding program 67, 69, 116–117
- permissions, file 173
- perror library function 248
- phases, translation 191, 228
- pipe 152, 170
- pointer argument 100
- pointer arithmetic 94, 98, 100–103, 117, 138, 205
- pointer arithmetic, illegal 102–103, 138, 205
- pointer arithmetic, scaling in 103, 198
- pointer comparison 102, 138, 187, 207
- pointer conversion 142, 198, 205
- pointer, declaration of 94, 100, 216
- pointer, file 160, 175, 242
- pointer generation 200



- pointer initialization 102, 138
- pointer, null 102, 198
- pointer subtraction 103, 138, 198
- pointer to function 118, 147, 201
- pointer to structure 136
- pointer, void \* 93, 103, 120, 199
- pointer vs. array 97, 99–100, 104, 113
- pointer-integer conversion 198–199, 205
- pointers and subscripts 97, 99, 217
- pointers, array of 107
- pointers, operations permitted on 103
- Polish notation 74
- pop function 77
- portability 3, 37, 43, 49, 147, 151, 153, 185
- position of braces 10
- postfix ++ and -- 46, 105
- pow library function 24, 251
- power function 25, 27
- #pragma 233
- precedence of operators 17, 52, 95, 131–132, 200
- prefix ++ and -- 46, 106
- preprocessor, macro 88, 228–233
- preprocessor name, \_\_FILE\_\_ 254
- preprocessor name, \_\_LINE\_\_ 254
- preprocessor names, predefined 233
- preprocessor operator, # 90, 230
- preprocessor operator, ## 90, 230
- preprocessor operator, defined 91, 232
- primary expression 200
- printf function 87
- printf conversions, table of 154, 244
- printf examples, table of 13, 154
- printf library function 7, 11, 18, 153, 244
- printing character 249
- program arguments *see* command-line arguments
- program, calculator 72, 74, 76, 158
- program, cat 160, 162–163
- program, character count 18
- program, dc1 125
- program, echo 115–116
- program, file concatenation 160
- program, file copy 16–17, 171, 173
- program format 10, 19, 23, 40, 138, 191
- program, fsize 181
- program, keyword count 133
- program, line count 19
- program, list directory 179
- program, longest-line 29, 32
- program, lower case conversion 153
- program, pattern finding 67, 69, 116–117
- program readability 10, 51, 64, 86, 147
- program, sorting 108, 119
- program, table lookup 143
- program, temperature conversion 8–9, 12–13, 15
- program, undc1 126
- program, white space count 22, 59
- program, word count 20, 139
- promotion, argument 45, 202
- promotion, integral 44, 197
- prototype, function 26, 30, 45, 72, 120, 202
- ptinrect function 130
- ptrdiff\_t type name 103, 147, 206
- push function 77
- pushback, input 78
- putc library function 161, 247
- putc macro 176
- putchar library function 15, 152, 161, 247
- puts library function 164, 247
- qsort function 87, 110, 120
- qsort library function 253
- qualifier, type 208, 211
- quicksort 87, 110
- quote character, ' 19, 37–38, 193
- quote character, " 8, 20, 38, 194
- \r carriage return character 38, 193
- raise library function 255
- rand function 46
- rand library function 252
- RAND\_MAX 252
- read system call 170
- readdir function 184
- readlines function 109
- realloc library function 252
- recursion 86, 139, 141, 182, 202, 269
- recursive-descent parser 123
- redirection *see* input/output redirection
- register, address of 210
- register storage class specifier 83, 210
- relational expression, numeric value of 42, 44
- relational operators 16, 41, 206
- removal of definition *see* #undef
- remove library function 242
- rename library function 242
- reservation of storage 210
- reserved words 36, 192
- return from main 26, 164
- return statement 25, 30, 70, 73, 225
- return, type conversion by 73, 225
- reverse function 62
- reverse Polish notation 74
- rewind library function 248
- Richards, M. 1
- right shift operator, >> 49, 206
- Ritchie, D. M. xi
- sbrk system call 187
- scaling in pointer arithmetic 103, 198
- scanf assignment suppression 157, 245
- scanf conversions, table of 158, 246
- scanf library function 96, 157, 246
- scientific notation 37, 73
- scope 195, 227–228
- scope, lexical 227
- scope of automatics 80, 228
- scope of externals 80, 228
- scope of label 66, 222, 228
- scope rules 80, 227
- screen output 15, 152, 163, 170
- SEEK\_CUR, SEEK\_END, SEEK\_SET 248
- selection statement 223

- self-referential structure 140, 213
- semicolon 10, 15, 18, 55, 57
- separate compilation 67, 80, 227
- sequencing of statements 222
- setbuf library function 243
- setjmp library function 254
- <setjmp.h> header 254
- setvbuf library function 243
- Shell, D. L. 61
- shellsort function 62
- shift operators 48, 206
- short type 10, 36, 196, 211
- side effects 53, 90, 200, 202
- SIG\_DFL, SIG\_ERR, SIG\_IGN 255
- sign extension 44–45, 177, 193
- signal library function 255
- <signal.h> header 255
- signed character 44, 195
- signed type 36, 211
- sin library function 251
- sinh library function 251
- size of numbers 9, 18, 36, 257
- size of structure 138, 204
- sizeof operator 91, 103, 135, 203–204, 247
- size\_t type name 103, 135, 147, 204, 242
- sorting, lexicographic 118
- sorting, numeric 118
- sorting program 108, 119
- sorting text lines 107, 119
- specifier, auto storage class 210
- specifier, enum 39, 215
- specifier, extern storage class 31, 33, 80, 210
- specifier, missing storage class 211
- specifier, register storage class 83, 210
- specifier, static storage class 83, 210
- specifier, storage class 210
- specifier, struct 212
- specifier, type 211
- specifier, union 212
- splicing, line 229
- sprintf library function 155, 245
- sqrt library function 251
- squeeze function 47
- srand function 46
- srand library function 252
- sscanf library function 246
- standard error 161, 170
- standard headers, table of 241
- standard input 151, 161, 170
- standard output 152, 161, 170
- stat structure 180
- stat system call 180
- statement terminator 10, 55
- statements 222–225
- statements, sequencing of 222
- stat.h include file 180–181
- static function declaration 83
- static storage class 31, 83, 195
- static storage class specifier 83, 210
- static variables, external 83
- static variables, internal 83
- statics, initialization of 40, 85, 219
- <stdarg.h> header 155, 174, 254
- <stddef.h> header 103, 135, 241
- stderr 161, 163, 242
- stdin 161, 242
- <stdio.h> contents 176
- <stdio.h> header 6, 16, 89–90, 102, 151–152, 241
- <stdlib.h> header 71, 142, 251
- stdout 161, 242
- storage allocator 142, 185–189
- storage class 195
- storage class, automatic 31, 195
- storage class declaration 210
- storage class specifier 210
- storage class specifier, auto 210
- storage class specifier, extern 31, 33, 80, 210
- storage class specifier, missing 211
- storage class specifier, register 83, 210
- storage class specifier, static 83, 210
- storage class, static 31, 83, 195
- storage, definition of 210
- storage order of array 112, 217
- storage, reservation of 210
- strcat function 48
- strcat library function 249
- strchr library function 249
- strcmp function 106
- strcmp library function 249
- strcpy function 105–106
- strcpy library function 249
- strcspn library function 250
- stream, binary 160, 241–242
- stream, text 15, 151, 241
- strerror library function 250
- strftime library function 256
- strindex function 69
- string concatenation 38, 90, 194
- string constant 7, 20, 30, 38, 99, 104, 194
- string constant, initialization by 86, 219
- string constant, wide 194
- string, length of 30, 38, 104
- string literal *see* string constant
- string, type of 200
- <string.h> header 39, 106, 249
- strlen function 39, 99, 103
- strlen library function 250
- strncat library function 249
- strncmp library function 249
- strncpy library function 249
- strpbrk library function 250
- strrchr library function 249
- strspn library function 250
- strstr library function 250
- strtod library function 251
- strtok library function 250
- strtol, strtoul library functions 252
- struct specifier 212
- structure arrays, initialization of 133
- structure declaration 128, 212
- structure initialization 128, 219
- structure member name 128, 213
- structure member operator, 128, 201
- structure, nested 129
- structure pointer operator, -> 131, 201

- structure, pointer to 136
- structure reference semantics 202
- structure reference syntax 202
- structure, self-referential 140, 213
- structure, size of 138, 204
- structure tag 128, 212
- structures, arrays of 132
- structures, mutually recursive 140, 213
- subarray argument 100
- subscripting, array 22, 97, 201, 217
- subscripts and pointers 97, 99, 217
- subscripts, negative 100
- subtraction operator, - 41, 205
- subtraction, pointer 103, 138, 198
- suffix, constant 193
- swap function 88, 96, 110, 121
- switch statement 58, 75, 223
- symbolic constants, length of 35
- syntax notation 194
- syntax of variable names 35, 192
- syscalls.h include file 171
- system calls 169
- system library function 167, 253
- 
- \t tab character 8, 11, 38, 193
- table lookup program 143
- table of escape sequences 38, 193
- table of operators 53
- table of printf conversions 154, 244
- table of printf examples 13, 154
- table of scanf conversions 158, 246
- table of standard headers 241
- tag, enumeration 215
- tag, structure 128, 212
- tag, union 212
- malloc function 142
- tan library function 251
- tanh library function 251
- temperature conversion program 8-9, 12-13, 15
- tentative definition 227
- terminal input and output 15
- termination, program 162, 164
- text lines, sorting 107, 119
- text stream 15, 151, 241
- Thompson, K. L. 1
- time library function 256
- <time.h> header 255
- time\_t type name 255
- tmpfile library function 243
- TMP\_MAX 243
- tmpnam library function 243
- token 191, 229
- token concatenation 90, 230
- token replacement 229
- tolower library function 153, 166, 249
- toupper library function 166, 249
- translation, order of 228
- translation phases 191, 228
- translation unit 191, 225, 227
- tree, binary 139
- tree, parse 123
- treeprint function 142
- 
- trigraph sequence 229
- trim function 65
- truncation by division 10, 41, 205
- truncation of floating point 45, 197
- two-dimensional array 110, 112, 220
- two-dimensional array, initialization of 112, 220
- type conversion by return 73, 225
- type conversion operator *see* cast
- type conversion rules 42, 44, 198
- type declaration 216
- type declaration, inconsistent 72
- type equivalence 221
- type, incomplete 212
- type names 220
- type of constant 37, 193
- type of string 200
- type qualifier 208, 211
- type specifier 211
- type specifier, missing 211
- typedef declaration 146, 210, 221
- types, arithmetic 196
- types, derived 1, 10, 196
- types, floating 196
- types, fundamental 9, 36, 195
- types, integral 196
- types.h include file 181, 183
- 
- ULONG\_MAX 252
- unary minus operator, - 203-204
- unary plus operator, + 203-204
- unbuffered getchar 171
- unbuffered input 170
- undcl program 126
- #undef 90, 172, 230
- underflow 41, 250, 255
- underscore character, \_ 35, 192, 241
- ungetc library function 166, 247
- ungetch function 79
- union, alignment by 186
- union declaration 147, 212
- union initialization 219
- union specifier 212
- union tag 212
- unions, operations on 148
- UNIX file system 169, 179
- unlink system call 174
- unsigned char type 36, 171
- unsigned character 44, 195
- unsigned constant 37, 193
- unsigned long constant 37, 193
- unsigned type 36, 50, 196, 211
- usual arithmetic conversions 42, 198
- 
- \v vertical tab character 38, 193
- va\_list, va\_start, va\_arg, va\_end 155, 174, 245, 254
- variable 195
- variable, address of 28, 94, 203
- variable, automatic 31, 74, 195
- variable, external 31, 73, 195
- variable length argument list 155, 174, 202, 218, 225, 254

- variable names, length of 192
- variable names, syntax of 35, 192
- vertical tab character, `\v` 38, 193
- `void * pointer` 93, 103, 120, 199
- void argument list 33, 73, 218, 225
- void type 30, 196, 199, 211
- volatile qualifier 196, 211
- `vprintf`, `vfprintf`, `vsprintf` library functions 174, 245

- `wchar_t` type name 193
- `while` statement 10, 60, 224
- `while` vs. `for` 14, 60
- white space 191
- white space characters 157, 166, 245, 249
- white space count program 22, 59
- wide character constant 193
- wide string constant 194
- word count program 20, 139
- `write` system call 170
- `writelines` function 109

- `\x` hexadecimal escape sequence 37, 193

- zero, omitted test against 56, 105



# **The C Programming Language**

**Second Edition**

**Brian W. Kernighan/Dennis M. Ritchie**

## **From the Preface**

We have tried to retain the brevity of the first edition. C is not a big language, and it is not well served by a big book. We have improved the exposition of critical features, such as pointers, that are central to C programming. We have refined the original examples, and have added new examples in several chapters. For instance, the treatment of complicated declarations is augmented by programs that convert declarations into words and vice versa. As before, all examples have been tested directly from the text, which is in machine-readable form.

As we said in the preface to the first edition, C “wears well as one’s experience with it grows.” With a decade more experience, we still feel that way. We hope that this book will help you to learn C and use it well.

PRENTICE HALL, Englewood Cliffs, N.J. 07632

ISBN 0-13-110362-8