Index

*/, 7, 62
++, 47
-lm, 71
/*, 7, 62
//, 7, 62
#define, 14, 95, 100, 108, 235
#if, 237
#ifdef, 237
#include, 7, 70, 174
__FILE__, 236
__LINE__, 236
* operator, 19, 20, 91, 93, 236
+ operator, 19, 20, 236
++ operator, 24, 120, 230, 236
+= operator, 24, 236
, operator, 230, 236
- operator, 19, 20, 236
-- operator, 24, 230, 236
-> operator, 135, 137, 236
. operator, 129, 137, 236
/ operator, 19, 20, 236
< operator, 30, 236
<< operator, 230, 233, 236
<= operator, 30, 236
= operator, 29, 31, 58, 120, 236
== operator, 29, 30, 120, 236
> operator, 30, 236
>= operator, 30, 236
>> operator, 230, 234, 236
? operator, 230, 236
[] operator, 100, 137, 236
% operator, 19, 20, 210, 236
& operator, 21, 90, 93, 97, 131, 230, 233, 236
&& operator, 30, 236
| operator, 230, 234, 236
|| operator, 30, 236
~ operator, 230, 234
! operator, 30, 236
!= operator, 30, 236
^ operator, 234, 236
~ operator, 236
%c format control, 19, 21–23, 25, 116
%d format control, 16, 21–23
%e format control, 23
%f format control, 17, 21, 23
%lf format control, 21, 23
%o format control, 230
%p format control, 91, 118
%s format control, 23, 116, 118, 131
%x format control, 230

abs, 72
abstraction, 63, 83, 133, 141, 174, 181
acos, 72
Algol, 4, 212
algorithm, 103, 141, 183, 203–225
    analysis, 104, 203–205
aliasing, 92, 101
amicable numbers, 80
ampersand character, 21, 97
anagrams, 128
ANSI standard, 6, 7, 32, 34, 62, 70, 71, 100, 149
approximation, 146, 152–156
apps, 3
argc, 84, 124
argument, 64
argv, 84, 124, 169
array, 18, 99–125, 129, 141, 163, 197
    address, 105, 113, 180
    argument, 109, 131
    assignment, 131
    automatic, 113
    binary output, 195
    bounds checking, 100, 101, 108, 123
    dynamic, 163
    equality, 131
    global, 113
    initialization, 112, 116
    large, 112, 113, 167
    multi-dimensional, 113
    of characters, 116
    of files, 200
    of linked lists, 208
    of pointers, 123
    of strings, 123, 124
    of structures, 137
    parallel, 126, 139
    subscript, 100, 101, 111
    two-dimensional, 109, 116, 121, 123
ascending runs, 127, 228
ASCII, 23, 25, 58, 119, 195, 210, 211
    table, 60
asin, 72
assert, 169
assignment statement, 8, 15, 23, 25, 31, 46, 58, 87, 93
    for arrays, 131
    for strings, 118
    for structures, 131
asymptotic cost, 204
atan, 72
atof, 119, 120
atoi, 119, 120, 125, 128
auto, 230
average-case analysis, 105, 178, 215

backslash character, 19
backup, 10–12
    off-site, 11

base case, 74, 171, 213
Basic, 4
big-Oh notation, 204, 226
binary file, 195
binary numbers, 231
binary search, 103, 154, 179, 206, 226
    ternary, 226
binary search tree, 177, 205, 207, 221
    average depth, 178, 191
    balanced, 191
    deletion, 192
    for sorting, 191
    insertion, 177, 183
    iterative implementation, 192
    polymorphic implementation, 183
    searching, 183
    smallest item, 192
    stick, 178
    traversal, 187
binary tree, 176, 221
    height, 191
    size, 191
bisection method, 154, 160
bit, 17, 90, 230
bit manipulation, 230
boundary case, 79
`break`, 39, 55, 56, 68
`bsearch`, 226
BST, *see* binary search tree
bucket, 210
byte, 90, 114, 165, 195

C
    advantages of, 5
    history, 5
    preprocessor, 235, 239
C++, 4
calculation, 63, 141
calculator, 5
`calloc`, 164, 166
`case`, 39
cast, 19, 24, 29, 165, 180
Celsius, 28, 43
central processor unit (CPU), 2
change calculation, 42, 98
`char`, 19, 23, 58, 90, 112, 116, 197, 234
`char**`, 123, 124, 169
Christie, I.W., 178
coding problem, 225
coin toss, 149
combinations, 80
command-line argument, 124, 195
comment, 7, 25, 62, 174, 237
comparison function, 180, 182, 183, 190, 191, 216
compiler, 7, 10, 90, 91, 108, 112
    flags, 69, 71, 83
    preprocessor, 235
    warning messages, 35, 36, 88
complex numbers, 138
compound interest, 49, 63, 72, 81
compound statement, 32, 46, 50
computer
    early, 5
    hardware, 2, 10
    memory, 90, 99, 231
    software, 3
    speed, 2, 145, 225
computer science, 1, 103, 146, 203, 225
`const`, 183, 230
constant, 14, 25, 38, 100, 236
    character, 19
    double, 17, 18
    integer, 18
    pointer, 106, 113
    type, 18
`continue`, 230
control structure, 8, 99
control-C, 57
control-D, 57, 101
control-Z, 57
`core`, 21
core dump, 21
`cos`, 72
`ctype.h`, 71, 121
cube root, 76
curve length, 152, 160

dangling else, 35
data abstraction, 133, 138, 174, 220
data structure, 99, 171
date manipulation, 42, 47, 113
day number in year, 42, 47
De Morgan's laws, 31
debugger, 238, 239
declaration, 8, 25
    array, 99, 101
    function, 64
    pointer, 91
    structure, 129
`default`, 39
dentifier, 164
dice roll, 149
dictionary data structure, 205–212
differential equation, 157
Dijkstra, E.W., 212
distinct words, 121, 128, 139, 163, 167, 183
divide and conquer, 142–147, 212
divide by zero, 20
`do`, 45, 56, 60, 229
`double`, 17, 18, 23, 99, 153, 164, 197, 234
Dutch national flag, 212

editor, 7, 9
`else`, 32, 33
empty statement, 32, 48, 49, 53, 120
end of file, 57, 58, 101
English, 3
`enum`, 230
`EOF`, 58, 121
equality operator, 29, 34
Erlang, 4
errata page, ix
escape character, 19
Euclidean distance, 138
Euler forward difference, 158
exclusive or, 234
execution time, 146, 167
exhaustive enumeration, 145
`exit`, 33, 84, 169
`EXIT_FAILURE`, 33, 83

EXIT_SUCCESS, 33, 83
exp, 72
exponent, 17, 18, 234, 239
exponential growth, 146, 176
expression, 15, 20
    evaluation order, 31
    logical, 29
    pointer, 91
extern, 230

fabs, 72
Fahrenheit, 28, 43
false, 29, 32, 46
fclose, 194, 200
Ferrari, 103, 205
fflush, 231
fgets, 200
Fibonacci numbers, 60
field width, 23
    negative, 23
FIFO queue, 172
file operations, 193–200
    merging, 199
    random access, 199
FILE*, 194, 195
flag, 55, 67, 95, 123
float, 18, 21, 23, 112, 153, 164, 234
floating exception, 21
floating point precision, 17, 18, 77, 80, 153, 155, 234, 239
fopen, 194, 195, 200
for, 45, 52, 99, 100, 171
format control string, 21
Fortran, ix, 4, 6, 58, 84
fprintf, 193, 194
fread, 194, 195, 197
free, 164, 166, 170, 188, 200
Free Software Foundation, 7
freopen, 194
fscanf, 194
fseek, 194, 199, 202
ftell, 231
full house, 160
function, 63–79, 83–90, 236
    argument variable, 73, 85, 88, 93, 106
    array argument, 105, 109
    call, 65
    choice of arguments, 72
    compilation, 66
    declaration, 64, 84
    evaluation, 65
    library, 70, 83
    main, 83
    pointer, 179, 181
    pointer argument, 93, 95
    prototype, 66, 70, 183
    recursive, 74, 90, 143, 145, 171, 186
    return value, 64, 84
    scope, 85, 88, 96
    static, 186
    structure arguments, 135
    without arguments, 84
functional language, 4, 74, 86
fwrite, 194, 197, 239

gambling games, 147
garbage collection, 117
gcc, 7, 34, 36, 69, 71, 97, 237
gdb, 239
generate and test, 141–142, 144, 146
geometric sequence, 167
getc, 194, 195
getchar, 58, 71
gets, 200, 229
getword, 121, 168, 189
goto, 230
greedy heuristic, 147
guard, 32, 46, 49, 53, 237

handle to structure, 170, 171, 174, 183, 195, 209
hashing, 207–212
    collision resolution, 208
Haskell, 4
header file, 71, 183
heap, 221
    construction, 223
heap sort, 220–224, 228
helloworld.c, 6, 96
hexadecimal number, 18, 91, 230, 239
Hoare, C.A.R., 212
Hooke's law, 156

identifier, 13, 99, 109, 134
if, 32, 39, 53, 99
imperative language, 4
in-order traversal, 187, 190, 207, 221
inf, 20, 239
infinite loop, 49, 57
informatics, 1
information, 1, 203
input buffer, 22
insertion sort, 103, 104, 126, 144, 147, 183, 190, 191, 204, 205
int, 16, 18, 21, 23, 90, 99, 112, 164, 195
int_swap, 94, 98, 104
integer
    arithmetic, 16, 18, 20, 232
    division, 19
    hash value, 208
    negative, 232
    overflow, 16, 48, 210, 230
    subtraction, 232
    unsigned, 164, 232
inversions, 127
isalpha, 71, 121
isascii, 71
isdigit, 71
islower, 71
isspace, 71
isupper, 71
iteration, 45, 63, 74, 141

Java, 4

K&R, 5
Kernighan, B., 5
KISS, 10, 32
knapsack problems, 145

lazy evaluation, 32
library, 70, 118

LIFO queue, 174
limits.h, 27
linear search, 103, 122, 123, 170
linked list, 141, 171, 178, 207, 208, 211, 221
    insertion, 171, 177, 190
    searching, 190
Lisp, 4
list_t, 171, 190
log, 72
logic-based language, 4, 74
logical
    and, 29, 32
    expression, 29
    or, 29, 32
long, 230, 234
long long, 234
loop, 45–59, 104
    body, 46, 56
    for reading, 56
    guard, 54, 56, 58, 118, 123
    infinite, 49, 57
    nested, 47, 50, 56, 63, 109
    termination, 54, 74
lowercase character, 60, 72
ls, 7
lucky, 16, 92, 101, 105, 117, 178, 214

M_E, 72
M_PI, 71, 72
M_SQRT2, 72
mac:, 7
MacOS, 3, 7, 57, 71
macro, 236, 239
magic number, 15
main, 7, 66, 69, 83, 124
    return value, 83, 96
make, 69, 189, 238
malloc, 164, 165, 167, 200, 218
man, 71, 118
mantissa, 17, 234, 239
mask operation, 234
math.h, 71
mathematics library, 70
median, 216, 227
memcmp, 231
memcpy, 231
memory, 2, 114, 123
    allocation, 163
    consumption, 112, 203, 224
    leak, 117, 167, 172
    management, 167
    read-only, 117
    word, 90, 99, 164, 231
merge sort, 218–220, 224, 227, 228
merging, 199, 218
ML, 4
modulus, 20
Monte Carlo estimation, 150
moons, 129
multi-way tree, 192
multiplicative operators, 19

nan, 20, 239
newline character, 19, 23, 50, 53, 58, 200
Newton Raphson method, 76, 160
non-linear equation, 154
NULL, 123, 166
null byte, 116, 118, 123, 124, 169, 200
number representation, 230, 239
numerical integration, 154

$O()$ notation, 204, 226
obfuscated C, 50, 120
object file, 69
object-oriented language, 4
octal number, 18, 230
od, 239
operating system, 3, 83, 84, 95
operator
    arithmetic, 19
    logical, 29
    mask, 234
    overloading, 233
    precedence table, 236
    relational, 29
    shift, 233
    unary, 91
optimal algorithm, 220
overloading, 233

palindromes, 128
parentheses, 19, 31, 137
partitioning, 212, 218, 226
    two way, 217, 227
Pascal, 4
perfect numbers, 80
pig manure, 128
pivot value, 226
PL/1, 4
planets, 129, 137, 177
pointer, 90–95, 106
    anonymous, 165, 181
    argument, 93, 95, 135
    arithmetic, 114
    assignment, 114
    comparison, 115
    constant, 113, 117, 123, 180
    difference, 115
    in an array, 123
    initialization, 92
    null, 123
    operations, 90
    to a binary search tree, 183
    to a function, 179
    to a linked list, 171
    to a pointer, 167
    to an array, 105
    variable, 91, 117, 164, 170
poker hand, 160
polygon, 138
polymorphism, 176, 182, 183, 186, 189, 226, 227
portability, 5, 164
post-order traversal, 187
postincrement operator, 24, 47, 120, 236
pow, 72
pre-reading, 57
precedence, 19, 31, 36, 93, 109, 137
    table, 236
predecrement operator, 230
preincrement operator, 24, 230, 236

preprocessor, 235, 239
pretty printer, 238
prime numbers, 55, 61, 66, 141, 209
printf, 7, 8, 16, 17, 22, 25, 118, 193
printing numbers, 22
priority queue, 220
problem size, 204
problem solving, 141–159
    techniques, *see* approximation, divide and conquer, generate and test, simulation
procedural language, 4, 86
procedure, 84
profiler, 238
program
    arguments, 124
    development, 9, 78, 83, 175
    layout, 15, 50
    return value, 84, 96
    termination, 84
programming language, 3
Prolog, 4
prompt, 7, 25, 49
pump priming, 57
putc, 194
putchar, 58

qsort, 182, 191, 197, 217, 227
quadratic roots, 41
queue, 172, 190
quick sort, viii, 212–218, 224, 226
    pivot value, 212, 214
    ternary, 227

rand, 147, 149, 216
RAND_MAX, 149
random, 149
random access file, 199
randomization, 209, 216
reading numbers, 21, 56
realloc, 164, 166, 167, 211
recurrence relation, 74, 206, 214, 218, 227
recursion, 74, 81, 90, 127, 141, 143, 145, 186, 206, 212
    base case, 74, 171
    mutual, 82
register, 230
repeat-until, 56
reserved word, 13, 84, 164
return, 7, 64, 67, 73, 75, 83, 84
Ritchie, D., 5
root finding, 76, 154, 160
rounding error, 17, 77, 80, 153, 156, 235
Runge-Kutta method, 159

scaffolding, 78, 95
scalar variable, 99
scanf, 21–23, 25, 33, 49, 57, 85, 120
Scheme, 4
scope, 85, 88, 96
search tree, *see* binary search tree
searching, 205
    algorithm, *see* binary search, linear search
seed, 149
SEEK_CUR, 199
SEEK_END, 199
SEEK_SET, 199
segmentation fault, 92, 95, 101
selection, 32–39, 63, 141
selection sort, 127, 144, 147
semantics, 3
semi-colon, 14, 15, 49, 66
sentinel, 113, 116, 118, 123, 124
separate chaining, 211
separate compilation, 69, 81, 88, 174, 183, 186
sex at noon taxes, 128
shell expansion, 124, 195
shift operators, 233
short, 230, 234
shortest path problem, 225
side effect, 31, 58, 87, 88, 120
sign-magnitude representation, 232
simulation, 147–152
    of time, 156–159
sin, 72
size_t, 164, 181, 190, 194
sizeof, 164, 165, 181, 236
slide rule, 5
Smalltalk, 4
software engineering, 1, 79, 238
sorting, 103, 127, 147, 159, 183, 205
    algorithm, *see* heap sort, insertion sort, merge sort, quick sort, selection sort
    choice of algorithm, 224
    exchanges required, 228
source code control, 238
spiral, 45, 54, 74, 104
spring equation, 156
sprintf, 231
sqrt, 72, 85
square root, 160
srand, 147, 149
srandom, 149
sscanf, 231
stack, 75, 83–85, 90, 91, 174, 190, 192
static, 89, 186, 230
stdarg, 231
stderr, 193, 202
stdin, 193
stdio.h, 194
stdlib.h, 33, 83, 119, 147, 182, 217, 226
stdout, 193, 202
stick, 178
storage class, 89, 182, 186, 230
strcasecmp, 119
strcat, 119, 128, 229
strchr, 231
strcmp, 119, 123
strcpy, 119, 120, 229
strcspn, 231
stream, 193
string, 7, 14, 116–125, 195
    allocation, 169, 189
    comparison, 120
    hash function, 209
    initialization, 117
    library, 118
    numeric, 120
    pointer, 118, 120
string.h, 118, 119, 123
strings.h, 118
strlen, 119, 128, 164, 169

`strncat`, 119
`strncmp`, 119
`strncpy`, 119, 123, 131
`strpbrk`, 231
`strrchr`, 231
`strspn`, 231
`strstr`, 231
`strtok`, 231
`struct`, *see* structure
structure, 129–137
    argument, 135
    array of, 137
    assignment, 131
    equality, 131
    initialization, 130
    linked, 170
    nested, 132
    pointer, 135
    printing, 131
    reading, 131
    recursive, 171
    sorting, 159
    tag, 130
    type naming, 134
stub, 78
subroutine, 84
subset sum problem, 144, 160, 161
`switch`, 38, 45, 56, 229
syntax, 10, 141
`system`, 231
system software, 3

tab character, 19
`tan`, 72
tax rates, 36
`tee`, 202
terminal, 23, 193
testing, 78
text file, 193
three-n problem, 53, 61
timetabling problem, 225
`tolower`, 71
`toupper`, 71
towers of Hanoi, 142, 145, 146, 161
tractor, 103, 205
trapezoidal rule, 154, 160
treadmill, 45, 49, 54
tree, *see* binary tree
tree sort, 191
`tree_t`, 183, 186
triangle numbers, 74, 81, 105
true, 29, 32, 46, 169
two-dimensional
    array, 109
    coordinates, 138
    structure, 176
    table, 50, 53, 63
twos-complement representation, 232, 239
type
    anonymous, 181
    conversion, 19, 24
    declaration, 15, 64
    for constant, 18
    return, 64
    user defined, 108
    variable, 16
`typedef`, 108, 111, 123, 130, 171

`ungetc`, 231
`union`, 230
Unix, 3, 7, 57, 71, 84, 96, 101, 202, 226, 239
    file redirection, 22, 193, 202
unlucky, 16, 92, 101, 105, 178, 214
`unsigned`, 210, 230, 234
uppercase character, 60, 72

validation, 78
variable, 15
    address, 21, 90, 99
    argument, 64, 68, 73, 93
    automatic, 90
    global, 87, 88, 229
    in an array, 99
    initialization, 16, 92, 112, 130
    local, 64, 68, 73, 85, 88
    pointer, 90, 91, 93, 114, 170
    shadowing, 88
    static, 89, 91, 112
    structure, 129
`void`, 84, 165
`void*`, 165, 180, 181, 183, 186, 190
`volatile`, 230

web site, ix
`while`, 8, 52, 101, 171
Williams, J.W.J., 220
Windows, 3, 7, 57
word count, 61
word frequency counting, 189
worst-case analysis, 105, 214, 218, 224

zero, 29, 116, 169, 239
zodiac, 191