

# ÍNDICE REMISSIVO



- Procure um identificador pelo seu nome e não pela categoria à qual ele pertence. Por exemplo, não tente encontrar *função printf()*; em vez disso, procure diretamente *printf()*.
- Número de página em negrito significa que a respectiva informação procurada está em nota de rodapé.

## Símbolo

- | | (barra vertical dupla) disjunção, operador de 119
- & (e comercial)
  - endereço, operador de 250
- && (e comercial duplo) conjunção, operador de 119–120
- ' ' (apóstrofes) caractere constante, delimitadores de 113
- "" (aspas)
  - array com string, uso em iniciação de 438–439
  - string constante, delimitadoras de 87
  - string constante, no interior de 445
- \* (asterisco)
  - acesso, operador de 515–516
  - indireção, operador de 252–253
  - multiplicação, operador de 81
  - ponteiro, definidor de 250
- \*/ (asterisco barra) comentário, delimitador de 126
- /\* (barra asterisco) comentário, delimitador de 126
- // (barra barra) comentário, delimitador de 126
- / (barra) divisão, operador de 81, 117
- \ (barra inversa) caractere de escape 113–114
- # (cerquilha/tralha) diretiva, início de 133
- { } (chaves)
  - bloco, delimitador de 173
  - string, iniciador de 438
  - variável estruturada, uso em iniciação de 377, 511–512, 521
- ^ (circunflexo) especificador de formato, usado em 587
- [ ] (colchete)
  - array, usado em acesso a 374
  - array, usado em definição de 374
  - especificador de formato, usado em 587–588
  - indexação, operador de 515–516
  - string, usado em acesso a 441–442
- <> (colchete angular) cabeçalho, delimitador de 129
- ≠ (diferente) diferente (pseudolinguagem), operador 83
- : (dois pontos)
  - case**, usado com 191
  - default**, usado com 191
  - rótulo, usado como declarador de 198
- != (exclamação igualdade) diferente, operador 118
- ! (exclamação) negação, operador de 119
- = (igualdade)
  - atribuição, operador de 121
  - igualdade (pseudolinguagem), operador de 83
- == (igualdade dupla) igualdade, operador de 118
- ? : (interrogação dois pontos) condicional, operador 200
- >= (maior igual) maior ou igual, operador 118
- > (maior) maior do que, operador 83, 118
- ≥ (maior ou igual) maior ou igual (pseudolinguagem), operador 83
- ++ (mais duplo) incremento, operador de 125–126
- + (mais) soma, operador de 81, 117
- <= (menor igual) menor ou igual, operador 118
- < (menor) menor do que, operador 83, 118
- ≤ (menor ou igual) menor ou igual (pseudolinguagem), operador 83
- () (parênteses)
  - acesso, operador de 515
  - conversão explícita, usados em operador de 124–125
  - expressão, usados em 82, 86, 116
  - função, usados com `xl`, 255, 260, 269
  - vazios 257, 270
- (ponto)
  - acesso, operador de 512, 515
  - estrutura, usado em acesso a 512–515
- ; (ponto e vírgula)
  - declaração, terminador de 173
  - for**, usado com 179, 181
  - instrução, terminador de 173
  - instrução vazia, como 174

- % (porcentagem)
  - especificador de formato, usado em 130–131
  - resto de divisão, operador de 81–82, 117–118
- \_ (subtração) xli, 110
- (traço)
  - inversão de sinal, operador de 81, 117
  - subtração, operador de 81, 117
- (traço duplo) decremento, operador de 125–126
- > (traço seguido por maior)
  - acesso, operador de 515–516
  - estrutura, usado em acesso a 512–513
- ... (três pontos)
  - elipse, usados como xl
  - parâmetro, usados como 571
- , (vírgula)
  - separador de declarações, usada como 201
  - separador de parâmetros, usada como 201, 255, 260, 269
  - separador de variáveis em definições, usada como 120
  - vírgula, operador 201–202

## A

- "**a+b**", modo de abertura de arquivo 561, 599
- .**a**, extensão de arquivo 54, 151
- "**a**", modo de abertura de arquivo 560
- "**a+**", modo de abertura de arquivo 560, 599
- "**a+t**", modo de abertura de arquivo 561, 599
- "**ab**", modo de abertura de arquivo 561
- abertura de arquivo 557–561, 606–607
- aborto de programa 118
  - array, causado por acesso a 375, 393–394
  - divisão por zero, causado por 118
  - exit()**, por meio de 522–524, 527, 542, 562–563
  - memória, causado por corrupção de 393–394
  - memória, causado por violação de 394, 465, 656
  - ponteiro nulo, causado por 254
  - resto de divisão por zero, causado por 118
  - string constante, causado por alteração de 439, 441, 464–465
  - voluntário 407–408, 522–524, 527, 542, 562–563
- abrir janela de comando aqui (comando do sistema Windows) 55
- acentuação, uso de xxxix–xl
- acesso a conteúdo de variável 252
- acesso aleatório a arquivo 591
- acesso direto a arquivo 591, 596–599
- adição 81–82, 117–118
- adição de ponteiro com inteiro 382–383
- alargamento, conversão de 124
- algoritmo 74
  - acompanhamento de 66
  - análise de problema resolvido por meio de 93–95
  - arquivo, para leitura sequencial de 595–596
  - auxílio visual para acompanhar 66
  - caso de entrada de 74
  - conceito de 74
  - construção de 78, 93–99
  - correto 74

- dividir e conquistar para, abordagem de construção 75–77
- entrada de 74
- Euclides, de 366
- existência de 74
- fluxo de execução de 87
- funcionalmente equivalente a outro 74
- implementação de 97
- incorreto 74
- legibilidade de 91–93
- ordenação, de 409
- projeto de 93–99
- pseudocódigo e 78
- receita culinária e 74, 75–77
- refinamento de 95–96
- refinamentos sucessivos de 75–77
- saída de 74
- subalgoritmo e 77
- subproblema e 75
- teste de 96–97
- alinhamento de dados na tela 157–158, 604–605
- alocação de memória 650
  - definição de tipo e 510
  - dinâmica 650
  - estática 121, 650
  - subdimensionamento em 650
  - superdimensionamento em 650
  - typedef** e, uso de 510
- alocação dinâmica de memória 650
  - funções de 650–654
  - heap e 656–657
  - ponteiro genérico usado em 654–655
  - teste de 657–658
- alusão de função 269–270
- ambiente de trabalho 52–55
- ambiente integrado de desenvolvimento (IDE) 51–52
- anagrama 506
- análise de problema 93–95
- aninho de bloco 173, 282–283, 328
- aninho de comentário 126
- aninho de estrutura 513–515, 525
- aninho de estrutura de controle 88, 182, 188–190, 200
- aninho de **if-else** 188–190, 331
- aninho de laço de repetição 182, 199
- aninho de união 522–524
- aninho em registro variante 522–524
- ano bissexto, checagem de 210–212
- ANSI C, padrão 110
- apóstrofes (símbolo) 113
- argc**, parâmetro 460
- argumento de linha de comando 460–461
- argv**, parâmetro 460
- aridade 80
- aritmética de ponteiros 382–385
- arquivo 556
  - abertura de 557–561, 606–607
  - acesso direto a 591, 596–599
  - armazenado 556

- arquivo (*continuação*)
  - atualização de registro de 617–621
  - buffer associado a 557
  - cabeçalho, de 128, 687–688
  - comparação de 615–617
  - cópia de 609–611
  - criado via teclado 603–604
  - erro em, indicador de 557
  - erro em processamento de 564–567
  - escrita em 592–596
  - escrita para leitura, passagem de 599
  - fechamento de 561–564
  - filtro de 603
  - final de, indicador de 557
  - fonte 48, 58–61, 135, 138
  - formato de 556
  - inclusão de 129, 688
  - indexação de 591
  - indicador de posição de 557, 596–599, 611–615
  - leitura em 592–596
  - leitura para escrita de, passagem de 599
  - makefile 696
  - número de linhas de 607–609
  - objeto 48
  - posicionamento em 596–599, 600, 611–615
  - programa, de 687
  - stream e xli
  - tamanho de 611–615
  - texto para binário, conversão de 621–624
- arquivo binário 556
  - criação de 621–624
  - escrita em 592–595
  - escrita na tela de registro de 624–627
  - leitura em 592–596, 624–627
  - número de registros de 627–629
- arquivo de texto 556
  - arquivo binário, conversão em 621–624
  - atualização de 617–621
  - conversão para binário de 621–624
  - criação via teclado de 603–604
  - escrita em 592
  - leitura em 588–590, 592, 658–665, 667–670
  - linha de, salto de 667–671
  - linha em, leitura de 658–665, 667–671
  - linhas de, número de 607–609
  - registro de 607
  - registro em, atualização de 617–621
- array 374
  - acesso com índice de 374
  - acesso com ponteiro de 385–387
  - acesso sequencial de 375–376
  - busca em 536–539, 539–544
  - bytes, de 591
  - caracteres, de 438–439, 446–449, 586–590
  - constante simbólica e 376
  - const**, qualificado com 389, 390, 397–398
  - corrupção de memória em acesso a 393–394
  - definição de 374
  - dimensão de 398–399
  - duração automática, de 377–379
  - duração fixa, de 377
  - elemento de 374, 398
  - endereço de 386
  - enumeração, indexação com 528
  - escrita na tela de 375, 378, 391, 395, 399
  - estruturas, de 536–539, 539–544
  - indexação de 374–376
  - indexação por meio de enumeração de 528
  - índice de 374
  - índice de, acesso com 374
  - iniciação de 376–379
  - iniciador designado de 525
  - inversão de 411–413
  - maior elemento de 400–401
  - média de elementos de 400–401
  - menor elemento de 400–401
  - multidimensional 398–400
  - número mágico e 376
  - ocorrência de valor em 401–404
  - ordenação de 409–411
  - parâmetro de função, como 389–393
  - permutação de 413–417
  - ponteiro, acesso com 385–387
  - ponteiro e 385–387
  - ponteiros, de 460–461
  - retorno de função, como 394–397
  - sizeof** e 380–382
  - size\_t** e 381–382
  - static**, qualificado com 377
  - strings, de 460–461
  - subdimensionamento de 650
  - superdimensionamento de 650
  - troca de posição de elementos de 415–416
  - unidimensional 398
  - zumbi e 394–397
- arredondamento
  - conversão de tipo, em 123
  - conversão de tipo sem 123, 260
  - printf()**, efetuado por 352–354
- ar (Unix), comando utilitário 695, 696
- ASCII, código 111
- aspas (símbolo)
  - array com string, uso em iniciação de 438–439
  - string constante, como delimitadoras de 87
  - string constante, no interior de 445
- assembler 47, 377
- assembly, linguagem 47
- associatividade 81, 116, 683
- asterisco (símbolo)
  - acesso, operador de 515–516
  - comentário, usado em 126
  - indireção, operador de 252–253
  - multiplicação, operador de 81
  - ponteiro, definidor de 250

"**at**", modo de abertura de arquivo 561  
**atoi()**, função 466  
 atribuição 79  
   C, em 121–122  
   conversão de 123–124  
   dependência de implementação causada por 122  
   efeito colateral em 122  
   entre estruturas 512  
   expressão de 121  
   instrução de 79, 121  
   linguagem algorítmica, em 79–80  
   operador de 121–122  
**auto**, qualificador de tipo 277  
 avaliação de expressão 80  
   aritmética 81–82  
   curto-circuito, com 120  
   lógica 84–85, 119–120  
   relacional 83, 118–119  
 avaliação de operandos, ordem de 116, 120, 122, 201–202, 464–465  
 avaliação de parâmetros, ordem de 464–465

## B

banco de dados da dinastia Tudor 601  
 barra inversa (símbolo) 113–114  
 barra (símbolo)  
   comentário, usada em delimitador de 126  
   divisão, operador de 81, 117  
 barra vertical (símbolo) 119  
 base e expoente 217–248  
 base numérica, representação em 113, 123, 350, 352, 572, 576  
 biblioteca 54, 128–130  
   cabeçalho de 128  
   componente de 128  
   função de xxxii, xliii, 128  
   módulo de 128  
   padrão 53, 128–129  
 biblioteca **LEITURAFACIL** 54, 129, 132–133  
   cabeçalho da 688–690  
   compilação da 694–696  
   função global da 690–694  
   função local da 688–690  
   implementação da 687–694  
   integração da 697–698  
   justificativa para a 685–687  
 bit de sinal 379  
 bits, sequência de 46, 123  
 bloco de comentário 127–128  
 bloco de instruções 172–173  
 bloco de memória 593–595  
 bloco, escopo de 282–283  
 bolha, ordenação pelo método da 409–411  
 booleano (pseudolinguagem), tipo 80, 82–84  
 Borland C++, compilador/IDE xxxviii  
**break**, instrução 194–196, 199  
 bubble sort, método de ordenação 409–411

buffer 567, 578–580  
 buffering 567  
 bug 50  
 busca  
   array, em 536–539, 539–544  
   string, em 454–455, 455–458  
 busca binária  
   depuração, usada em 349–350  
   raiz quadrada com, cálculo de 299–302  
 busca sequencial  
   array, em 406, 536–539, 539–544

## C

**%c**, especificador de formato 130, 572  
**.c**, extensão de arquivo 142  
**-c**, opção de compilação 150  
**.cpp**, extensão de arquivo 142  
 C11, padrão 110  
 C89, padrão 110  
 C90, padrão 110  
 C99, padrão 110  
 cabeçalho 128  
   **ctype.h** 461–464  
   inclusão de 130  
   **leitura.h** 132, 687–688  
   **math.h** 129  
   **stdio.h** 130–133  
   **stdlib.h** 202, 254, 466, 523, 650  
   **string.h** 442  
   **time.h** 202  
 cabeçalho de função 255–257  
 calculadora dirigida por menu 304–306  
**calloc()**, função 651–652  
 camelo, notação de xl  
 campo 510, 601  
   acesso a 512–515  
   alternativo 522–524  
   estrutura, de 510, 511, 522–524  
   fixo 522–524  
   indicador 523  
   iniciação de 511–512, 524–525  
   registro, de 510, 601  
   registro variante, de 522–524  
   variante 522–524  
 caractere 111–112, 113  
   classificação de 461–463  
   codificação de 111–112  
   constante 113–114  
   escape, de 113–114  
   gráfico 111  
   identificador, permitido em 110  
   letra maiúscula/minúscula de, conversão em 463–464  
   nulo 438  
   representação de 111  
   representação gráfica, com 113  
   representação gráfica, sem 113

- caractere (*continuação*)
  - sequência de escape, representado como 113–114
  - terminal 438
  - transformação de 463–464
- caracteres, array de 438–439, 446–448, 448–449, 586–590
- caracteres, código de 111–112
- caracteres, conjunto básico de 111–112
- caracteres, conjunto de 111–112
- caracteres, sequência de 87
- carregador (programa) 51
- casas decimais escritas por **printf()** 352
- case**, parte de instrução **switch** 190–194
- caso de entrada 74
- casting 124–125
- Celsius para Fahrenheit, conversão de 205–207
- cerquilha (símbolo) 133
- chamada de função 117, 130, 172, 260–269
  - alusão e 269–270
  - atribuição, em 260
  - expressão, usada em 260
  - instrução isolada, como 260
  - main()** 137, 460–461
  - passagem de parâmetro em 261–269, 390–394
  - printf()** 130, 197–198, 349–350
  - retorno de valor em 260
- char** \* (ponteiro para **char**), tipo 383, 439, 439–442, 444
- char**, tipo 112
- chave criptográfica 475
- chaves (símbolo)
  - array, usadas em iniciação de 377, 438
  - bloco, usadas em delimitação de 173
- circunflexo (símbolo) 587
- Clang, compilador xxxvii, 397, 696
- clearerr()**, função 567
- C, linguagem xxxi, 110
  - ANSI da, padrão 110
  - biblioteca padrão da 128–129
  - C++, comparação com 142
  - compatibilidade histórica da 110
  - compilador da xxxviii, 53
  - conjunto básico de caracteres da 111
  - extensão **void main()** da 461
  - história da 110
  - ISO da, padrão 110
  - língua franca em programação, como xxxi
  - padrão da 110
  - portabilidade na 110
  - pré-processor da 129
  - programa simples em, estrutura de 137
- C++, linguagem 58, 142
- CodeBlocks, IDE 53–54, 55–64
  - abreviação (abbreviation) no 345–347
  - auto** indent do, opção de configuração 139
  - backspace unindents do, opção de configuração 139
  - brace completion do, opção de configuração 139
  - build do, comando 61–64
  - build log do, painel 61–68
  - compile current file do, comando 63–64
  - compiler de, opção de configuração 56–58
  - configuração do 56–58, 138–139
  - default code do, opção de configuração 140–142
  - editor de programa do 60–61
  - endentação no 139
  - indent options do, opção de configuração 139–140
  - linker usado pelo, configuração do 57–58
  - molde de programa no 140–142
  - painéis do 55–56
  - programa executável no, construção de 61–64
  - programa-fonte no, criação/edição de 58–61
  - settings do, opção de configuração 56, 139
  - show indentation guides do, opção de configuração 139
  - show spaces do, opção de configuração 139
  - smart indent do, opção de configuração 139
  - TAB indents do, opção de configuração 139
  - TAB options do, opção de configuração 139
  - TAB size in spaces do, opção de configuração 139
  - tabulação no, configuração de 138–139
  - toolchain executables do, opção de configuração 56
  - use TAB caractere do, opção de configuração 139
- codificação 135–137
- código de caracteres 111–112
- código de substituição direta 475
- código espaguete 332
- código-fonte xxxviii, 48, 58–61, 135, 138
- código morto 199–202
- código-objeto 48
- colação de caracteres 450
  - LC\_COLLATE** e, constante 453
  - localidade e 452–453
  - ordem de 450
  - setlocale()** e, função 453
  - strcmp()** e, função 450–452
  - strcoll()** e, função 452–454
- colchete angular (símbolo) 129
- colchete (símbolo) xl
  - array, usado em acesso a 374
  - array, usado em definição de 374
  - especificador de formato, usado em 587–588
  - indexação, usado como operador de 515–516
  - string, usado em acesso a 441–442
- colisão de identificadores 284
- Collatz, conjectura de 306–308, 367–368
- coloração de sintaxe 52
- comentário 68, 126–128
  - algoritmo, em 91
  - aninhado 126
  - bloco, de 127–128, 333–334
  - caixa de 336
  - conteúdo de 333–334
  - delimitador de 126
  - depuração, usado em 350
  - didático xlii, 127
  - informação comum em 127–128
  - irrelevante 335

- comentário (*continuação*)
  - linha, de 335–336
  - modelo de 336–337
  - momento de inclusão de 91
  - propósito de 127
  - redundante 335
  - remoção de trecho de programa, usado em 350
- comparação de arquivos 615–617
- comparação de números reais 297–299, 355
- comparação de strings 449–454
- comparação entre inteiro com sinal e inteiro sem sinal 471–472
- compatibilidade em conversão de tipo 122–124, 252
- compatibilidade entre compiladores 110
- compatibilidade entre ponteiros 252, 262–264
- compatibilidade entre tipos primitivos 123
- compatibilidade histórica 110
- compilação de programa xli, 142
  - c de, opção 150
  - erro de 63, 66, 143–146, 347
  - construção de programa executável e xli
  - ligação e 150–151
  - o de, opção 150
  - pedantic de, opção 57
  - std=c99 de, opção 57
- compilador 48, 51
  - Clang xxxvii, 397, 696
  - mensagem de advertência emitida por 146–148
  - mensagem de erro emitida por 144–146, 148–149
- compile, programa 142
- componente de biblioteca 128
- computador 46–47
- concatenação de strings 114, 448–449
- condição de erro 564–567
- condição de exceção 50
  - chamada de função, em 259
  - processamento de arquivo, em 564–567, 600
- condição de parada 183–184
  - atingida 183
  - nunca atingida 185
  - satisfeita 183
  - zero, resultante sempre em 184
- condicional, expressão 87, 89–90, 183–184, 331
- condicional, operador 200–201, 452
- conectivo lógico 80
- confiabilidade 50
- conflito de identificadores 284
- conjunção lógica 83–85, 119–120
- conjunto básico de caracteres 111
- conjunto de instruções 46
- console xxxix, 51
- constante 113–115
  - caractere 113–114
  - enumeração, de 526–530
  - inteira 113
  - lógica 83
  - ponteiro 388–389
  - real 113
  - string 114–115, 439
  - variável 387–389
- constante simbólica 133
  - const** e 388
  - #define** e 133
  - definição de 133–134
  - notação para escrita de 329
  - número mágico e 330–333
- const**, qualificador 387
  - constante simbólica e 388
  - conteúdo apontado, aplicado a 388–389
  - elemento de array, aplicado a 389
  - parâmetro, aplicado a 390, 391, 397–398, 448, 465, 517, 531
  - ponteiro, aplicado a 388–389
  - string constante, aplicado a 439
  - variável, aplicado a 387–389
- construtor de tipo 515
- contador (variável) 181
- contagem, laço de 181–182
- contagem, variável de 176
- conteúdo de memória 252
  - buffer, de 561, 567, 574–580
  - constante 387–389, 398, 439
  - indeterminado 280, 377
  - parâmetro, de 264–269
  - ponteiro, referenciado por 252–253
  - reinterpretação de 263–264
  - variável, de 46, 79, 121
- continuação de instrução 93
- continuação de string constante 114
- continue**, instrução 196–200
- conversão de Celsius em Fahrenheit 205–207
- conversão de string em número 466–468
- conversão de tipo 122–125
  - aritmética usual 124
  - alargamento, de 124
  - atribuição, de 123–124
  - automática 122–124
  - casting 124–128
  - explícita 124–128
  - hierarquia de 124
  - implícita 330–333
  - passagem de parâmetro, em 262–264
  - regra de 124
  - retorno de função, em 259–260
- conversão entre letra maiúscula e minúscula 463
- cópia de arquivo 609–611
- cópia de array 392–393
- cópia de parâmetro 264–269
- cópia de string 446–448
- corpo de estrutura de controle 175
- corpo de função 257–258
- corpo de laço 175
- corpo vazio 176–178, 258
- corrupção de memória 393–394
  - array, em acesso a 374–376

corrupção de memória (*continuação*)

**fgets()**, usando 589

**gets()**, usando 588–590

**scanf()**, usando 586

**strcat()**, usando 449

**strcpy()**, usando 447

criação de programa-fonte 58–61

criptografia 475–478

[CTRL]+[C], combinação de teclas 65, 177

[CTRL]+[D], combinação de teclas 570

[CTRL]+[J], combinação de teclas 345

[CTRL]+[Z], combinação de teclas 570

**ctype.h**, cabeçalho 129, 461–464

curto-circuito de operador 116, 120

## D

**%d**, especificador de formato 130, 572

dados de aluno, processamento de 530–533

data, leitura e validação de 363–366, 483–486

**DBL\_DIG**, constante 353

debugger 51

declaração de função 269–270

declaração de variável 80

decremento, operador de xlii, 125–126

default code em CodeBlocks 141–142

**default**, parte de instrução **switch** 191, 331

**#define**, diretiva 133–134

definição de array 374

definição de constante simbólica 133–134

definição de estrutura 510–511

definição de função 255–260

definição de ponteiro 250–252

definição de tipo 510

definição de união 520–521

definição de variável 120–121

**const**, com uso de 387

iniciação, com 121

objetivo de 120

**static**, com uso de 278–279, 281–282

definidor de tipo 251, 515–516

array, de 374

enumeração, de 526

estrutura, de 510–511

função, de 255

ponteiro, de 251

união, de 520–521

De Morgan, leis de 184

dependência de implementação 110

arquivos abertos simultaneamente, do número de 559

código de caracteres, de 114

dependência de compilador versus 110

**EOF**, de 564

expressão com 122

expressão sem 201

**FILE**, da estrutura 557

**int**, da largura do tipo 112

**NULL**, de 254

número mágico, causada por 330

dependente de compilador 110

depuração de programas 347–350

depurador 51

despedida de programa 196, 338, 488

desvio condicional 87

C, em 186–194

**if-else** 186–190

linguagem algorítmica, em 87–89

se-então-senão 87–88

selecione-caso 88–89

**switch-case** 190–194

desvio incondicional 87

**break** 194–196, 199

C, em 194–200

código morto e 199–200

**continue** 196–200

**goto** 198–200, 332–333

instrução inacessível e 199–200

linguagem algorítmica, em 90–91

pare 90–91

desvio múltiplo 190–194

diferente (símbolo) 83

dígito, classificação de caractere como 461–463

dígito, conversão em número inteiro de 472–474

dígito em código de caracteres 450

dígito, uso em identificador de 110–111

dígito verificador 473

diretiva 129

disjunção lógica 83–85, 119–120

dispositivo periférico 556

arquivo e 556

buffer associado a 561, 567, 578–580

entrada, de 51, 556, 568

impressora 558

padrão 567

saída, de 51, 556

stream e 556

dívida, calculo de 207–208

dividir e conquistar, abordagem 75–77

divisão 81–82, 117–118

inteira 117–118

por zero 118

real 82, 117–118

resto de 81, 117–118

dois pontos (símbolo)

**case**, usados com 191

**default**, usados com 191

operador condicional, usados em 200

rótulo, usados como declarador de 198

DOS, sistema operacional 54–55, 64–65

**(double)**, operador de conversão 124, 351

**double\*** (ponteiro para **double**), tipo 252

**double**, tipo 112

**do-while**, instrução 178–179, 194, 196, 199

download xxxvii–xxxviii, 52, 54, 66, 128

duração de variável 277–280  
 automática 277, 279–280  
 fixa 278–280  
 iniciação de acordo com 279–280

## E

**%e**, especificador de formato 130  
**%E**, especificador de formato 130  
 e (pseudolinguagem), operador de conjunção lógica 83–85  
 EBCDIC, código de caracteres 111  
 e comercial (símbolo)  
   conjunção, usado em operador de 119–120  
   endereço, usado como operador de 250  
 economia de memória 264, 520–521, 663  
 edição de programa-fonte 58–61  
 editor de programa 51, 60–61  
 efeito colateral 116, 122, 125–126  
 eficiência de programa 50  
   aprendizagem de programação e 50  
   assembly e 47  
   equivalência funcional e 96  
   **goto** e, uso de 332–333  
   indexação de arrays e 387  
 elemento de array 374  
   acesso de 374–376, 386–387  
   iniciação de 377–379, 399, 525  
**else**, parte de instrução **if** 186–190  
 encerramento de execução de programa  
   dirigido por menu 194–195  
   normal 64–65, 137–138  
 encerramento de laço de repetição 183  
   acidental 331  
   **break**, por meio de 194–196, 199  
   **do-while** 178  
   enquanto-faça 89  
   faça-enquanto 90  
   **for** 179, 181–182  
   **goto**, por meio de 332–333  
   infinito 184–186  
   instrução **pare**, por meio de 90–91  
   **while** 175  
 endentação 91  
   automática 52  
   coerência em uso de 92  
   compilador e 176  
   corpo de função, em 257  
   espaços recomendados em 331  
   estrutura de controle, em 330  
   ilusória 187  
   instrução **if-else** aninhada, em 189–190  
   instrução rotulada, em 198  
   linguagem algorítmica, em 91  
   recomendação de estilo para 92, 93, 189–190, 198, 257, 330, 331  
   tabulação e 138–140

endereço 46–47  
   array, de 386  
   operador de 250  
   retorno de 394–397  
   string constante, de 439  
   unidade de memória, de 46  
   variável, de 79, 250  
**#endif**, diretiva 688  
 enquanto-faça (pseudolinguagem), instrução 89–90  
**[ENTER]**, tecla 61  
   buffering e 567  
   leitura de dados com, encerramento de 568, 574–578  
   '**\n**', representada como 574–578  
   quebra de linha, como representação de 574–578  
   remoção de buffer de caractere correspondente a 578–580  
   remoção de string de caractere correspondente a 590  
   string vazio e 442  
 entrada, caso de 74  
 entrada de algoritmo 74, 93–95  
 entrada, meio de 86, 568, 573–574, 589  
**enum**, definidor de tipo 526  
 enumeração 526–530  
**EOF**, constante 564–567, 570  
 EP (exercício de programação) xxxvi, 66–67  
 época de referência em programação 202  
 equação de segundo grau, programa de 94–99  
 Eratóstenes, crivo de 671–682  
 erro de programação 50  
   arquivo, em processamento de 564–567  
   classificação de 347  
   compilação, de 63–64, 143–144  
   corriqueiro 143–144  
   execução, de 347, 349  
   ligação, de 149–151  
   lógica, de 153, 347, 394, 394–397, 464–466  
   overflow, devido a 123  
   prática de 151–153  
   precisão, de 123, 297–299  
   sintaxe, de 63–64, 143–144  
   truncamento, de 352–353  
   zumbi, causado por 394–397  
 escoamento de memória 657, 664  
 escopo 280–284  
   arquivo, de 281–282  
   bloco, de 282–283  
   conflito de identificador e 284  
   função, de 282, 283  
   programa, de 280–281  
   sobreposição de 284  
 escreva (pseudolinguagem), instrução 86  
 escrita de dados em arquivo binário 592–595, 621–624  
 escrita de dados em arquivo de texto 592  
 escrita de dados na tela 130–132  
   array, de 375, 378, 391, 395, 399  
   caractere, de 132, 592  
   estrutura, de 531  
   linguagem algorítmica, em 86–87



- escrita de dados na tela (*continuação*)
  - string, de 130–132, 592–593
  - tabela, em forma de 537, 624–627
- espaço em branco, classificação de 461
- espaço em branco em string 481–483
- espaço em branco horizontal em programa 92
- espaço em branco vertical em programa 92
- espaguete, código 332
- especificador de formato 130–132, 571–573
  - %5.2f** (família printf) 131
  - %.16f** (família printf) 352
  - %c** (família printf) 130
  - %c** (família scanf) 572
  - %d** (família printf) 130
  - %d** (família scanf) 572
  - %E** (família printf) 130
  - %f** (família printf) 130
  - %g** (família printf) 130
  - %G** (família printf) 130
  - %i** (família printf) 130
  - %i** (família scanf) 572
  - %lf** (família scanf) 572
  - %n[^]** (família scanf) 587
  - %n[]** (família scanf) 587–588
  - %ns** (família scanf) 586
  - %s** (família printf) 130
  - %s** (família scanf) 572, 586
  - %u** (família printf) 382
- estilo de programação 50, 67–68
  - chaves, para uso de 331–332
  - comentário, para escrita de 333–337
  - constante simbólica, para escrita de 329–333
  - desvio incondicional, para uso de 331–332
  - documentação de função, para 333–337
  - endentação, para 330, 331
  - espaço em branco, para uso de 121, 337
  - estrutura de controle, para escrita de 330–332
  - expressão, para escrita de 328–333
  - goto**, para uso de 332–333
  - identificador, para escrita de xl, 329–333
  - if-else**, para uso de 331
  - instrução, para escrita de 328–333
  - interação com usuário, para 337–338
  - jargão e, para uso de 328–333
  - número mágico e 330–333
  - operador de igualdade, para uso de 331
  - string constante, para escrita de 328–333
  - switch-case**, para uso de 331
- estrutura de controle 87, 172
  - bloco de instrução e 91
  - break** 194–196, 199
  - código espaguete e 332–333
  - código morto e 199–200
  - continue** 196–200
  - desvio condicional 87, 174, 186–194
  - desvio incondicional 87, 174, 194–200
  - desvios múltiplos, de 190
  - do-while** 178–179
  - endentação e 91, 330
  - enquanto-faça 89–90
  - estilo para escrita de 330–332
  - expressão condicional e 183–184
  - faça-enquanto 90
  - fluxo de execução e 172, 328
  - for** 179–182
  - goto** 198, 332–333
  - if-else** 186–190
    - instrução inacessível e 199–200
    - pare 90
    - repetição, laço aninhado de 199
    - repetição, laço de 87, 174–186
    - se-então-senão 87–88
    - seleção, de 190
    - selecione-caso 88–89
    - switch-case** 190–194
    - while** 174–178
  - estrutura (variável) 510
    - aninhada 513–516
    - arquivo, escrita em 621–624
    - atribuição de 512
    - campo de, acesso a 512–513
    - definição de 510–511
    - iniciação de 511–512
    - parâmetro de função, como 516–517
    - retorno de função, como 517–519
    - teclado, leitura via 531
    - tela, escrita na 531, 537, 624–627
  - Euclides, algoritmo de 366
  - exclamação (símbolo)
    - diferença, usada em operador de 118
    - negação, usada como operador de 119
  - .exe**, extensão de arquivo 62
  - execução de programa 54–55, 64–65
  - exemplo de programação xxxv–xxxvi, 65
  - exercício de programação (EP) xxxvi, 66, 66–67
  - exercício de revisão, resolução de xxxvi, 66–69
  - exit()**, função 522–524, 527, 542, 562–563
  - expoente 100, 113
  - exponenciação 217–219
  - expressão 79, 116
    - aritmética 80, 81–82
    - booleana 83
    - compilador, resolvida pelo 380
    - complexa 79
    - condicional 87, 89–90, 183–184, 331
    - constante 191
    - controle, de 183–184
    - função em, uso de 85, 260
    - lógica 80, 83–85
    - relacional 80, 82
    - simples 79
    - \*str++ 446–508
  - expressão telefônica, tradução de 539–544
  - extensão da linguagem C **void main()** 461

extensão de arquivo

- .a 54, 151
- .c 142–143
- .cpp 142–143
- .exe 62
- .h 128
- .o 62

extensão do código ASCII 112

**extern**, qualificador 269–270

## F

**%f**, especificador de formato 130

**fabs()**, função 298

faça-enquanto (pseudolinguagem), instrução 90

Fahrenheit, conversão de Celsius em 205–207

falso (pseudolinguagem), constante lógica 82

fatoração de número inteiro positivo 221–223

fator de escala 382

fatorial, cálculo de 258, 259, 272, 295

**fclose()**, função 561

fechamento de arquivo 561–564

**feof()**, função 564–567

ferramenta de desenvolvimento 53

- Borland C++ xxxviii
- Clang xxxvii, 397, 696
- GCC 53
- GDB 53
- MinGW 53

**ferror()**, função 564–567

**fflush()**, função 567

**fgetc()**, função 592

**fgetc()**, função de biblioteca 591

**fgets()**, função 588–590

**fgets()**, função de biblioteca 591

Fibonacci, sequência de 101–103, 362–363, 406–409

**FILENAME\_MAX**, constante 559, 606–607

**FILE**, tipo 557

filtro de arquivo 603

final de arquivo 557, 564–567

final de linha 114

final de string 438

**float.h**, cabeçalho 353

fluxo de execução de programa 87, 172, 328

**fopen()**, função 557–559, 561

**FOPEN\_MAX**, constante 559

**for**, instrução 179–182

- aninhada 182–183
- array, usada em acesso sequencial de 375–376
- break**, usada com 194, 199
- continue**, usada com 196–200
- expressão da, omissão de 181
- iniciação de variável em 182
- instrução vazia em, uso de 182
- jargão usado com 185
- laço de contagem, usada como 181–182
- laço infinito, usada como 185

operador vírgula, uso com 201–202

**while**, relação com 180

formatação em memória 604–605

formato de argumento passado para programa 460

formato de arquivo 556

formato de comentário 127

formato de endentação 92

formato de inclusão de cabeçalho 129

formato de instrução em linguagem algorítmica 86–91

**fprintf()**, função 564–565

**fprintf()**, função de biblioteca 592

**fputc()**, função 592

**fputc()**, função de biblioteca 591

**fputs()**, função 592–593

**fputs()**, função de biblioteca 591

fragmentação de heap 656

**fread()**, função 593–595

**fread()**, função de biblioteca 591

**free()**, função 652–653

**fscanf()**, função de biblioteca 592

**fseek()**, função 567, 596–599

**fseek()**, função de biblioteca 592, 596–600

**ftell()**, função 597–599

**ftell()**, função de biblioteca 592, 596

função 254–255

alusão de 269

apresentação de 333–334

benefício obtido com uso de 255

biblioteca, de xxxii, xliv, 128

cabeçalho de 255–257

cabeçalho de, formato de 255

chamada de 260–269

corpo de 257–258

corpo vazio, com 258

declaração de 269

definição de 255–260

denominação de 257

documentação de 333–337

escopo de 282

escopo de arquivo, com 281–282, 688–690

forma de 255

global, com escopo 280, 690–694

local, com escopo 688–690

nome de 257

parâmetro de 256

parâmetro de, declaração de 256

parâmetros de, lista de 256

parâmetro variante, com 571

posicionamento em arquivo, de 596–599

projeto de 260

protótipo de 270

retorno de 255–256, 394–397

retorno de, tipo de 255–256

**static**, definida com 281

zumbi e retorno de 394–397

**fwrite()**, função 593–595

**fwrite()**, função de biblioteca 591

## G

**%g**, especificador de formato 130  
**%G**, especificador de formato 130  
 GCC, compilador xxxvii, 53  
   compilação do, opção de 61, 150  
   **gcc.exe** e, programa executável 61  
   ligação do, opção de 61  
   linker associado ao 61  
 GDB, depurador 53  
 gerador de número aleatório 202  
**getchar()**, função 64, 568–571, 573–578  
**gets()**, função 588–590  
**goto**, instrução 198, 332–333  
   código espaguete, como origem de 332  
   código espaguete, sem 285, 418, 484, 585, 690, 691  
   eficiência da 332–333  
   laço de repetição aninhado, usado em 199  
   paranoia causada por 332–333  
   rótulo usado com 198  
 grupo de precedência de operador 81

## H

**.h**, extensão de arquivo 62  
 hardware xxxvii  
 HCI (Human-Computer Interaction) 337  
 heap 656–657  
 história da linguagem C 110  
 história de linguagens de programação 46–47  
 hospedeiro, sistema 137

## I

**%i**, especificador de formato 130  
 IDE (ambiente integrado de desenvolvimento) 51–52  
 identificador 110  
   caractere permitido em 110  
   conflito de 284  
   maiúscula e minúscula em, diferenciação de 111  
   notação para escrita de xl, 121, 257, 329–333, 510  
   ocultação de 283  
   regra de formação de 110–111  
   representativo 92  
   reservado 111  
   restrição de denominação de 110  
   significativo 92  
   tamanho de 111  
 IEEE 754, padrão 350  
**if-else**, instrução 186–190  
**#ifndef**, diretiva 688  
 igualdade e atribuição, confusão entre 188, 331  
 igualdade em C 118  
 igualdade em linguagem algorítmica 83  
 igualdade (símbolo)  
   atribuição, usada como operador de 121  
   diferença, usada em operador de 118

igualdade, usada como operador de 83  
 igualdade, usada em operador de 118  
   maior ou igual, usada em operador 118  
   menor ou igual, usada em operador 118  
 implementação da linguagem C 110  
 implementação de número real 350–354  
 implementação, dependência de 110  
 implementação de programa 97–107, 134–137  
**#include**, diretiva 129  
 inclusão de cabeçalho 129  
 incompatibilidade de tipos 252  
 incremento, operador de xlii, 125–126  
 indexação 374  
   arquivo, de 591, 596  
   array, de 374, 387, 399  
   byte em arquivo, de 597, 615–617  
   linha em arquivo, de 599  
   operador de 515  
   registro em arquivo, de 591, 596  
 indireção xli  
 indireção, operador de 252–253  
 iniciação 121  
   array com string, de 438–439  
   array, de 376–379  
   duração de variável e 279–280  
   estrutura aninhada, de 514  
   estrutura, de 511–512  
   explícita 280  
   implícita 280  
   iniciador designado, por meio de 524–525  
   ponteiro com string, de 439  
   ponteiro, de 251  
   união, de 521  
   variável constante, de 388–389  
   variável, de 121  
   variável de duração automática, de 279  
   variável de duração fixa, de 279  
 iniciador 377  
   array, de 377  
   designado 524–525  
   estrutura, de 511  
   ponteiro, de 251  
   string, de 438  
   união, de 521  
 inserção de string em string 488–491  
 inspeção de programa 136  
 instalação da biblioteca **LEITURAFACIL** xxxviii, 54, 68–69, 697–698  
 instalação de ambiente de trabalho 52–55  
 instalação de CodeBlocks 53–54  
 instalação de MinGW 53  
 instalação de open command window here 55  
 instrução 46–47, 172  
   ilegal 144  
   inacessível 199–202  
   muito extensa, divisão de 93  
   rotulada 198

instrução (*continuação*)  
 rótulo de 198  
 seleção, de 190–194  
 sem efeito 172  
 tipo de 172  
 vazia 174

instrução vazia 174  
 acidental 176–178, 331  
 intencional 182, 189, 201

instruções, bloco de 172–173

instruções, conjunto de 46

instruções, sequência de 172–173

(**int**), operador de conversão 124–125, 351, 471

interação com usuário 51  
 laço infinito, com 194–196, 275–277  
 menu, dirigida por 194–196, 273  
 programa com argumento de linha de comando, de 488  
 recomendação para 337–338, 488

interação dirigida por menu 194–196, 273–277  
 calculadora com 304–306  
 confirmação sim/não em 274–275  
 leitura de opção em 273–274  
 menu em, apresentação de 273

interação humano-computador 337

interpretador 48–49

interrogação (símbolo)  
 operador condicional, usada em 200

**int**, maior valor do tipo 212–214

**INT\_MAX**, constante 212–214

**int\*** (ponteiro para **int**), tipo 251

**int**, tipo 112

inversão de sinal 117

**isalnum()**, função 461

**isalpha()**, função 461

**isblank()**, função 461

**isdigit()**, função 461

**islower()**, função 461

ISO 8859, padrão 112

ISO da linguagem C, padrão 110

**ispunct()**, função 461

**isspace()**, função 461, 568–571

**isupper()**, função 461

## J

jargão 328–333

jogo de palavras 665–671

juízo, cálculo de 207–208

## K

Knuth, Donald 298

## L

laço de contagem 181–182

laço de repetição 87  
 aninhado 182–183

C, em 174–186

condição de parada de 183–184

**do-while** 178–179

enquanto-faça 89–90

faça-enquanto 90

**for** 179–182

infinito 89, 184–186

linguagem algorítmica, em 89–90

**while** 174–178

largura de tipo 112

**LC\_COLLATE**, constante 453–454

**LC\_CTYPE**, constante 463

ld, linker GCC 150–170

**LeCaractere()**, função 132, 690–691

legibilidade de algoritmo 91–93

legibilidade de programa 49  
 comentário em, importância de 333–337  
 constante simbólica e, uso de 330–333  
 documentação em, importância de 333–337  
 função em, influência do uso de 255

**goto** e, uso de 332–333

notação para identificadores em, importância de 121

número mágico e 330

leia (pseudolinguagem), instrução 86

**LeInteiro()**, função 132, 584, 691

leis de De Morgan 183–184

leitura árabe 388–389

leitura de dados 86  
 linguagem algorítmica, em 86  
 sequencial em arquivo 592–648  
 string, de 658–665

leitura de dados em arquivo binário  
 acesso direto, com 592–596  
 sequencial 624–629

leitura de dados em arquivo de texto 588–590, 592, 658–665, 667–670

leitura de dados via teclado xlii, 132–133, 568  
 aluno, de 530–532  
 buffering, com 573–580  
 caractere, de 132, 568–571  
 data, de 363–366, 483–486  
 estrutura, de 530–533

**getchar()**, usando 568–571

laço de repetição em, uso de 580–585

linha de texto, de 568, 658–665

nome, de 468–470

nota, de 354–355

número de identificação, de 470–472

número inteiro, de 132, 418, 685–687

número natural, de 284–286

número real, de 132, 571–573

opção, de 273–274

**scanf()**, usando 571–573

string, de xlii, 442–443, 586–590, 658–665

**leitura.h**, cabeçalho 132

**LeOpcao()**, função 273–274, 693–694

**LeReal()**, função 132, 691–692

**LeString()**, função 442–443, 692–693  
 letra maiúscula/minúscula, classificação de 461  
 letra maiúscula/minúscula como opção de menu 193  
 letra maiúscula/minúscula, comparando strings sem diferenciar 486–487  
 letra maiúscula/minúscula, conversão entre 463–464  
 letra maiúscula/minúscula, distinção entre 111  
 letra maiúscula/minúscula em código de caracteres 450  
**%lf**, especificador de formato 572  
 liberação de memória  
   dinâmica 561, 652–653, 656–657  
   estática 277, 394, 396–397, 656  
**libleitura.a**, arquivo 54, 694–696  
 ligação 64  
   erro de 149–151  
   -**lleitura** de, opção 57  
   -**lm** de, opção 58  
   parâmetro, de 261–264  
**limits.h**, cabeçalho 212  
**LimpaBuffer()**, função 578–580, 688–689  
 linguagem 46–47, 78  
   algorítmica 78  
   alto nível, de 47  
   assembly 47  
   baixo nível, de 47  
   C++ 58, 142  
   estruturada 332  
   máquina, de 46  
   nativa de computador 46  
   Pascal xxxi, 111, 258, 264  
 linguagem algorítmica 77–78  
   bloco de instruções em 91  
   constante simbólica em 134  
   declaração de variável em 80  
   endentação em 91  
   entrada de dados em 86–87  
   estrutura de controle em 87–91  
   função em, chamada de 85  
   saída de dados em 86  
   subprograma em 270–272  
   tipo de dado usado em 80  
 linha de comando, argumento de 337, 338, 460–461, 487–488  
 linha de comando, invocação de compilador via 150  
 linha de comando, invocação de linker via 142  
 linha, leitura de 568–569, 658–682  
 linhas de arquivo, número de 607–609  
 linker 48, 51, 58, 142  
 Linux, sistema operacional xli, 64  
 lista  
   escrita na tela de 375, 378, 391, 395, 399  
 lista de verificação 136  
 -**lleitura**, opção de ligação 57  
 -**lm**, opção de ligação 58  
 loader 51  
**locale.h**, cabeçalho 453  
 localidade 452–454

## M

Mac OS X, sistema operacional xxxvii, 696  
**main()**, função 137, 459–461  
   **argc** da, parâmetro 460  
   argumento de linha de comando e 460  
   **argv** da, parâmetro 460  
   hospedeiro, relação com 459  
   parâmetro, com 460–461  
   portável da, uso 461  
   programa com argumento e 460  
   programa hospedado, uso em 137, 459  
   protótipos da 459–460  
   retorno da 137–138, 461  
   **return 0** pela, uso de 137–138  
   **void** como tipo de retorno da 461  
 maior do que, operador 83, 118  
 maior ou igual, operador 83, 118  
 maior ou igual (símbolo) 83  
 maior (símbolo)  
   acesso, usado em operador de 515–516  
   estrutura, usado em acesso a 512–513  
   maior do que, usado como operador 83, 118  
   maior ou igual, usado em operador 118  
 maior valor do tipo **int** 212–214  
 mais (símbolo)  
   incremento, usado em operador de 125–126  
   soma, usado como operador de 81, 117  
 make, comando utilitário 695  
 makefile, arquivo 695, 696  
**malloc()**, função 651  
 manutenibilidade 49  
**math.h**, cabeçalho 129  
 máximo divisor comum (MDC), cálculo de 289–291, 366  
 MDC (máximo divisor comum), cálculo de 289–291, 366  
 média de elementos de array 400–401  
 mega-sena 417–420  
 meio de entrada 86, 568, 573–580, 685–687  
 meio de entrada padrão 568, 573–580, 685–687  
 meio de saída 86, 130, 568  
 meio de saída padrão 130, 568  
 membro de estrutura 510  
 memória 46  
   acesso inválido de 254  
   alocação de 121, 277  
   array em, representação de 374  
   bloco de 591  
   buffer e 567  
   compartilhamento de 520  
   conteúdo constante de 387–389, 439  
   conteúdo de, acesso com ponteiro a 250–254, 382–388  
   conteúdo de, interpretação de 120  
   conteúdo de, reinterpretação de 263  
   contiguidade em 593  
   corrupção de 393–394  
   definição de tipo e alocação de 510

memória (*continuação*)  
 dispositivo periférico e, troca de dados entre 556, 593  
 endereço em 250  
 escoamento de 657, 664  
 estrutura em, representação de 517, 520–521  
 formatação em 605  
 fragmentação de 656  
 função de alocação de 650–654  
 heap e 656  
 instruções e dados em, armazenamento de 656  
 liberação de 277  
 parâmetro em, representação de 256, 265–269  
 partição de 655–657  
 pilha e 656  
 stack e 656  
 subdimensionamento de 650  
 superdimensionamento de 650  
 união em, representação de 520  
 unidade de 46  
 variável de duração fixa em, armazenamento de 656  
 variável em, duração de armazenamento de 277–280, 377–379  
 variável em, representação de 79  
 violação de 465  
 zumbi e 394–397

menor do que, operador 83, 118  
 menor ou igual, operador 83, 118  
 menor ou igual (símbolo) 83  
 menor (símbolo) 83, 118  
 menos unário 117

mensagem de advertência 146–150, 348, 397, 465–466  
 mensagem de despedida 196, 273  
 mensagem de erro 144–146, 148–153  
 mensagem de informação para usuário 213, 331, 337–338, 606  
 mensagem de prompt 133, 337  
 mensagem sobre propósito de programa 337  
 menu, interação dirigida por 273–277, 304–306  
 mingw32-make, comando utilitário (MinGW/DOS) 695  
 MinGW, ferramentas de desenvolvimento 53  
 mínimo múltiplo comum (MMC) 291–293  
 MMC (mínimo múltiplo comum) 291–293  
 mnemônico 47

modo de abertura de arquivo 560–561  
 "a" 560  
 "a+" 560, 599  
 "a+b" 561, 599  
 "ab" 561  
 "a+t" 561, 599  
 "at" 561  
 "r" 560, 598, 614, 618, 623, 664  
 "r+" 560, 599  
 "r+b" 561, 599  
 "rb" 561  
 "r+t" 561, 599  
 "rt" 561  
 "w" 560, 563, 618  
 "w+" 560, 564, 599

"w+b" 561, 599  
 "wb" 561, 594, 611, 623  
 "w+t" 561, 599  
 "wt" 561  
 binário 560  
 texto, de 560–561

módulo de biblioteca 128, 281  
 ctype 463  
 LEITURAFACIL 54, 687–694  
 stdio 130, 556, 600, 687  
 stdlib 202, 466  
 string 438

Molde.c, arquivo 141  
 molde de programa em CodeBlocks 140–142  
 monitor de vídeo 51  
 montador 47  
 multiplicação 81–82, 117  
 multiplicação, tabuada de 361–362  
 Murphy, lei de 600

## N

'\n', sequência de escape 114  
 não (pseudolinguagem), operador de negação 83–85  
 negação lógica 83–85, 119–120, 183–184  
 nome de arquivo 558, 559, 561  
 nome de arquivo da biblioteca LEITURAFACIL 54  
 nome de arquivo de biblioteca GCC 151  
 nome de array xl, 386  
 nome de campo de estrutura 510–511  
 nome de constante de enumeração 526–530  
 nome de constante simbólica 329  
 nome de função xl, 257  
 nome de identificador xl, 110, 121, 329  
 nome de localidade 453  
 nome de parâmetro 257  
 nome de parâmetro em alusão 269  
 nome de programa-fonte recebido por programa 460  
 nome de rótulo 198  
 nome de tipo definido pelo programador 510  
 nome de variável 79, 121  
 nome de variável de contagem 328  
 nome do linker GCC 150  
 nome, leitura de 468–470  
 nome mnemônico 48  
 nome, ocultação de 283–284  
 nome representativo 92, 94, 271, 329  
 nomes, colisão de 284  
 nomes, conflito de 284  
 nome significativo 92, 94, 271, 329  
 notação adotada neste livro xl, 121  
 notação de número real 113, 130  
 nota, leitura de 354–355  
 %ns, especificador de formato 586  
 NULL, constante 254  
 uso com time() 202

numeração de linha de arquivo de texto 603–604

número aleatório 202, 214–215, 302–304, 417–420, 667–669  
 número de identificação 470–472  
 número de ponto fixo 350  
 número de ponto flutuante xli, 350  
 número inteiro  
   **char** e, tipo 112  
   Collatz para, conjectura de 306–308, 367–368  
   escrita de 130–131  
   fatoração de 221–223  
   fatorial de 258, 259, 272, 295  
   Fibonacci, de 101–103, 362–363, 406–409  
   ímpar 204–205  
   **int** e, tipo 112  
   leitura de 132, 284–286, 583–585, 691  
   maior valor do tipo **int** 212–214  
   ocorrência de dígito em 404–406  
   par 204–205  
   primo 286–289, 355–358, 356–358, 671–682  
   separação em centena, dezena e unidade de 156–157  
   sinal, sem 379  
 número mágico xlii, 330, 376  
 número natural 284–286, 291, 355  
 número primo 286–289, 355–358, 671–675  
 número real xli  
   comparação de 297–299  
   **double** e, tipo 112  
   escrita de 130–132  
   expoente de 113  
   imprecisão de 350–354  
   leitura de 132, 571–573  
   mantissa de 113  
   notação científica de 113, 130  
   notação convencional de 113, 130  
   ponto fixo, de 350  
   ponto flutuante, de 350  
   truncamento de 123, 352–354  
 número sem sinal 379

## O

.o, extensão de arquivo 128  
 -o, opção de compilação 150–151  
 ocorrência de caractere em string 455–458  
 ocorrência de constante simbólica em programa-fonte 133  
 ocorrência de dígito em número inteiro 404–406  
 ocorrência de letra em palavra 478–481  
 ocorrência de string em string 454–455  
 ocorrência de valor em array 401–404  
 octal, base 113  
 ocultação de identificador 283  
 opção de compilação 61, 150  
 opção de ligação 61  
 open command window here (extensão do sistema MS Windows) 55  
 operação ilegal 118  
 operador 79, 115  
   acesso, de 515–516

aridade de 80, 115  
 aritmético 81–82, 117–118  
 associatividade de 81, 116, 683  
 binário 80, 115  
 chamada de função, de 515  
 condicional 200–201, 452  
 conversão explícita, de 124–128  
 curto-circuito de 116, 120  
 decremento, de 125–126  
 efeito colateral de 116, 122, 125–126  
 incremento, de 125–126  
 indexação, de 515  
 indireção, de 252–253  
 ordem de avaliação e 116  
 ponto 512–513  
 precedência de 85, 115, 683  
 prefixo 125–126  
 propriedade de 80–81, 115–116  
 relacional 82–83, 118–119  
 resultado de 80, 115  
 seta 512–513  
**sizeof** 142–143, 280, 379–382  
 sufixo 125–126  
 ternário 115, 200, 452  
 unário 80, 115  
 vírgula 201–202  
 operador lógico 83, 119  
   C, em 119–122  
   conjunção 83–85, 119–120  
   curto-circuito de 120  
   disjunção 83–85, 119–120  
   linguagem algorítmica, em 83–85  
   negação 83–85, 119  
   operando de 83–85, 119  
   precedência de 119  
 operando 79, 115  
   aridade e 115  
   aritmético, de operador 117  
   atribuição, do operador de 122  
   com sinal e sem sinal juntos 381–382  
   condicional, do operador 200  
   conversão implícita e 123–124  
   curto-circuito e 116, 120  
   decremento, do operador de 125  
   efeito colateral e 116, 122  
   endereço, do operador de 253  
   incremento, do operador de 125  
   lógico, de operador 83–85, 119  
   negativo em divisão inteira 117  
   negativo em resto de divisão inteira 117  
   ordem de avaliação de 116  
   relacional, de operador 82, 118  
   retorno de função como 256  
   **sizeof**, do operador 379–380  
   vírgula, do operador 201  
 ordem alfabética 450  
 ordem de avaliação de operando 116

ordem de colação 450  
 ordenação de array 409–411  
 ordenação de inteiros 358–359  
 ordenação de lista 409–411  
 ordenação pelo método da bolha 409–411  
 organizador prévio xliii  
 ou (pseudolinguagem), operador lógico 83–85  
 overflow 123, 212–214, 407–409

## P

padrão IEEE 754 353  
 padrão ISO 8859 112  
 padrão ISO da linguagem C 110  
 palavra-chave 111  
 palavra-chave da linguagem C 111

**auto** 277  
**break** 194–196  
**case** 190–194  
**char** 112  
**const** 387–389  
**continue** 196–200  
**default** 191  
**do** 178–179  
**double** 112  
**else** 186–190  
**enum** 526  
**extern** 269–270  
**for** 179–182  
**goto** 198  
**if** 186–190  
**int** 112  
**return** 258–260  
**sizeof** 379–382  
**static** 278–279  
**struct** 511  
**switch** 190–194  
**typedef** 510  
**union** 520  
**void** 256  
**while** 174–178

palavra mnemônica 47  
 palavra reservada 111  
 parâmetro 256, 261  
   argumento e xli  
   array usado como xlii  
   avaliação de, ordem de 464–466  
   casamento de 261–264  
   **const**, qualificado com xli  
   conversão de 262  
   declaração de 256–257  
   entrada, de 261  
   entrada e saída, de 261  
   formal 261  
   ligação de 261–264  
   linha de comando, de 460–461  
   modo de 261

passagem de 261–269  
 real 261  
 referência de, passagem por 264  
 saída, de 261  
 valor de, passagem por 264  
 variante 571

parênteses (símbolo)  
 acesso, usados como operador de 515  
 conversão explícita, usados em operador de 124–125  
 expressão, usados em 82, 86, 92, 116  
 função, usados em xl, 255, 260, 269  
 redundantes 86, 92  
**sizeof**, usados com 379, 380  
 vazios 257, 270

pare (pseudolinguagem), instrução 90–91  
 partição de memória reservada para programa 656–657  
 Pascal, linguagem xxxi, 111, 258, 264  
 passagem de parâmetro 264  
   array multidimensional em, uso de 399–400  
   array unidimensional em, uso de 390–393  
   endereço de variável em, uso de 267–269, 518–519, 522–524, 655  
   enumeração em, uso de 528–529  
   estrutura em, uso de 516–517, 522–524  
   ordem de avaliação em 464–465  
   ponteiro em, uso de 267–269, 518–519, 522–524, 655  
   referência, por 264, 267–269  
   valor, por 264

pause, comando (DOS) 64  
**-pedantic**, opção de compilação 57  
 pilha de execução 656  
 PIS/PASEP, validação de 470–472  
 plataforma 49  
 ponteiro 250  
   aritmética de 382–385  
   array com, relação de 385–387  
   compatibilidade de 252  
   constante 388–389  
   constante, para 388–389  
   decremento de 383  
   definição de 250  
   genérico 654–655  
   incremento de 383  
   indireção de 252–253  
   inteiro com, soma de 382–383  
   inteiro com, subtração de 383  
   inválido 254  
   **NULL**, com conteúdo 254  
   nulo 254  
   parâmetro, usado como 264–269  
   subtração de 383

ponto decimal 113  
 ponto e vírgula (símbolo)  
 declaração, usado como terminador de 173  
**for**, usado com instrução 179, 181  
 indevido 176–178, 331



- ponto e vírgula (*continuação*)
  - instrução, usado como terminador de 173
  - instrução vazia, usado como 174
- ponto fixo, número de 350
- ponto flutuante, número de xl, 350
- ponto, operador 512–513
- ponto (símbolo)
  - acesso, usado como operador de 515
  - estrutura, usado em acesso a 512–515
- porcentagem (símbolo)
  - especificador de formato, usado em 130–131
  - resto de divisão, usado como operador de 82, 117–118
- portabilidade 49
- posicionamento em arquivo 596–599
- potências, tabela de 100
- pow()**, função 85, 207
- PP (projeto de programação) xxxvii
- preâmbulo de exemplo/exercício de programação xxxvi
- precedência de operador 81, 683
  - aritmético 117–118
  - aritmético, relacional e lógico 120
  - decremento sufixo, de xlii
  - grupo de 81
  - incremento sufixo, de xlii
  - lógico 119
  - relacional 118–119
  - tabela geral de 683
- precisão de número real 123, 298, 299, 352–354
- precisão, erro de 123, 298
- precisão, perda de 123
- pré-processador de C 129
- printf()**, função 130–132
  - arredondamento efetuado pela 352
  - aspas usando, escrita de 445
  - cabeçalho necessário para 260
  - casa decimal pela, escrita de 131, 352
  - depuração, usada em 349–350
  - operador condicional, usada com 452
  - prompt, usada em 133
  - protótipo da 571
  - quebra de linha provocada pela 114, 444
  - sequência de escape, usada com 114
  - string de formatação da 130
  - string, usada em escrita de 443–444
  - valor retornado pela 130, 197
- problema de programação, análise de 93–95
- procedimento 250
- processador 46–47
- processamento de arquivo 556
  - acesso direto, por 591, 596–599
  - acesso sequencial, por 591, 592–596
  - algoritmo para leitura sequencial em 595–596
  - bloco, por 591, 593–595
  - byte, por 591, 592
  - caractere, por 591, 592
  - formatação em memória, com 604–605
  - formatado 591
  - linha, por 591, 592–593
- programa 46–47, 93
  - argumento, com 460
  - console, baseado em 51, 64–65
  - construção de, etapa de 93–99, 134–137
  - depuração de 126, 146, 347–350
  - DOS de, execução no sistema 64–65
  - execução de 54–55, 64–65
  - execução de, espaço de 655–657
  - execução de, fluxo de 172, 328
  - executável xli, 48, 61–64, 135–136, 142, 149–150
  - fonte xxxviii, 48, 58–61, 135, 138
  - hospedado 137, 459
  - inspeção de 136
  - interativo 51
  - melindroso 154–156
  - minimalista xxxv, 65
  - monoarquivo 137–142
  - objeto 48
  - projeto de 93–99
  - qualidade desejável em 49–50
  - robusto 153–154
  - simples 137
  - standalone 48
  - terminal, baseado em 51
  - Unix de, execução no sistema 64
  - Windows de, execução no sistema 64–65
- programa, abordagem de construção de 75
  - construir e consertar 67, 348
  - dividir e conquistar 75–77, 93–99
  - refinamentos sucessivos, de 75–77, 93–99
  - tentativa e erro, por 348
- programação
  - alto nível, de 64–72
  - análise de problema de 93–95
  - aprendizagem de 65–68
  - auxílio visual para 66
  - baixo nível, de 47
  - conhecimento necessário em xxxi
  - estruturada 332
  - prática de 65–68
  - tentativa e erro em, uso de 348
- programa com argumento 460–461
  - cálculo de tamanho de arquivo, para 611–615
  - comparação de arquivos, para 615–617
  - contagem de linhas de arquivo, para 607–609
  - cópia de arquivo, para 609–611
  - soma de valores inteiros, para 487–488
- projeto de programação (PP) xxxvii
- prompt 133, 337
- propriedade de operador 80–81, 115–116
  - aridade 80, 115
  - associatividade 81, 116, 683
  - curto-circuito 116, 120
  - efeito colateral 116, 122, 125–126
  - ordem de avaliação de operandos 116

propriedade de operador (*continuação*)  
 precedência 85, 115, 683  
 resultado 80, 115  
 protótipo de função 270  
 pseudocódigo 78  
**putchar()**, função 132  
**puts()**, função 444

## Q

quadrado, desenho de 359–361  
 quebra de linha 556  
 arquivo de texto de, contagem em 607–609  
 arquivo de texto, em 556  
 buffer de entrada padrão de, remoção em 578  
 buffer de entrada padrão, em 574  
 buffering de linha e 567  
 espaço em branco, interpretada como 111, 114–115, 461  
 linha sem incluir, lendo 658–665  
 '\n', representada como 114  
**printf()**, provocada por 444  
**puts()**, provocada por 444  
 sequência de escape, representada como 114  
 string de, remoção em 590

## R

"**r+b**", modo de abertura de arquivo 561, 599  
 "**r**", modo de abertura de arquivo 560, 664, 667  
 "**r+**", modo de abertura de arquivo 560, 599  
 raiz quadrada, cálculo de 85, 187, 299–302  
**rand()**, função 85, 202, 214–215, 302  
 "**rb**", modo de abertura de arquivo 561  
**realloc()**, função 653–658  
 recomendação sobre  
 advertência, trato com mensagem de 146–148  
 algoritmo, acompanhamento de 66  
 algoritmo, implementação de 97–99, 134–137  
 algoritmo, refinamento de 95–96  
 algoritmo, teste de 96–97  
 alocação dinâmica, teste de 657–658  
 ambiente de desenvolvimento 53–54  
 ambiente integrado de desenvolvimento (IDE) 53–54  
 arquivo aberto para atualização, processamento de 599  
 arquivo, fechamento de 564  
 arquivo, preparo para processamento de 558  
 arquivo, teste de erro de processamento de 566  
 array como parâmetro formal, escrita de 390  
 array como parâmetro real, uso de 394  
**break**, uso de 196  
 buffer, esvaziamento de 578–580  
 chaves, uso de 176–178  
 compilador errado, prevenção de uso de 142–143  
 condição de parada, escrita de 183–184  
 constante de enumeração, escrita de 526  
 constante simbólica, uso de 329  
**const**, uso de 389, 391, 397–398

**continue**, uso de 198  
**default** de **switch**, uso de parte 331  
 endereço, retorno de 395, 397  
 enumeração, uso de 530  
**EOF**, uso de 566  
 erro de ligação, trato com 149–151  
 erro de sintaxe, trato com 143–146, 148–149  
 escoamento de memória, prevenção de 657  
 espaço em branco, uso de 121, 337  
 estilo de programação 67–68, 121  
 estrutura de controle, escrita de 330–333  
 exemplo de programação, exploração de 66  
 exercício de programação, resolução de 66–68  
 exercício de revisão, resolução de 66  
 expressão condicional, escrita de 183–184  
**feof()**, uso de 566  
**feof()**, uso de 566  
**free()**, uso de 652, 657  
 função, chamada de 464–465  
 função de alocação dinâmica, uso de 657–658  
 função de processamento de arquivo, uso de 600  
 função, omissão de tipo de retorno de 256  
**goto**, uso de 332–333  
 IDE (ambiente integrado de desenvolvimento) 53–54  
 identificador, escrita de xl, 329–333  
**if-else**, aninho de 189–190  
 instrução vazia, uso de 174  
 interação com usuário 337–342, 580–585  
 laço de repetição infinito, escrita de 185–186  
 leitura além de final de arquivo, teste de 566  
 leitura com laço de repetição 581–590  
 leitura de string 586, 588, 589, 658–665  
 memória, prevenção de escoamento de 657  
 mistura de tipos 122, 382  
 operador vírgula, uso de 201–202  
 ponteiro genérico, uso de 655  
 ponteiro, iniciação de 254  
 problema, análise de 93–95  
 programa, acompanhamento de 66  
 programação, aprendizagem de xxxvii, 65–68  
 programação, estudo de xxxvii  
 programa executável, construção de 135–136  
 programa minimalista, estudo de 65  
 programa, teste de 136  
**realloc()**, uso de 654  
 retorno de endereço 394, 397  
**rewind()**, uso de 599  
**scanf()**, leitura com 572  
 sequência de estudo deste livro xxxvii  
 string como parâmetro formal, escrita de 444  
 string como parâmetro real, uso de 465  
 tipo de retorno de função, omissão de 256  
 usuário, interação com 337–338, 580–585  
 redirecionamento de entrada/saída 603–604  
 refinamentos sucessivos 75–77  
 registro, campo de 601

- registro de arquivo 596, 601
  - atualização de 617–621
  - contagem de 627–629
  - escrita na tela de 624–627
  - texto para binário de, conversão de 621–624
- registro variante 522–524
- reinterpretação de conteúdo de memória 263–264
- reinvenção da roda 68, 344, 530
- remoção de arquivo 562–564
- remoção de caractere no buffer de entrada padrão 578–580
- remoção de espaço em branco em string 481–483
- remoção de indicação de erro em arquivo 566–567
- remoção de quebra de linha em string 590
- representação de caractere 111, 113–114
- representação de número real 113, 352
- representação em memória
  - array, de 374
  - estrutura, de 517, 520–521
  - parâmetro, de 256, 265–269
  - união, de 520
  - variável, de 79
- representação gráfica de caractere 113
- resto de divisão inteira 81, 117
- resultado (propriedade de operador) 80, 82, 115
- retorno de função 258–260
  - array como 535
  - comentário, descrito em 271
  - endereço, de 531–532
  - enumeração, de constante de 533
  - estrutura, de 517–519
  - expressão, usado em 260
  - return** e, instrução 258–260
  - scanf()**, de 571, 581
  - tipo de 255–256
  - valor. com 259–260
  - valor. sem 258–259
  - void**, do tipo 258–259
  - zero 137–138
  - zumbi 518–519
- return 0**, instrução 138
- return**, instrução 258–260
- reusabilidade 50
- reúso de código xlii, 68, 77, 344–346
- rewind()**, função 566, 599
- rewind()**, função de biblioteca 596
- Ritchie, Dennis 110
- robustez 50, 132–133, 153–154, 580–585, 600, 686
- rotina 250
- rótulo 198
  - case** 190
  - default** 190
  - instrução, de 198
- "**rt**", modo de abertura de arquivo 561
- "**r+t**", modo de abertura de arquivo 561
- %s**, especificador de formato 130, 572, 586
- saída de algoritmo 93–95
- saída de dados 130–132
- saída, meio de 86, 130, 568
- scanf()**, função 571–573, 573–578, 586–589
- SEEK\_CUR**, constante de biblioteca 597
- SEEK\_END**, constante de biblioteca 597
- SEEK\_SET**, constante de biblioteca 597
- se-então-senão (pseudolinguagem), instrução 87–88
- seleção, instrução de 190–194
- selecione-caso (pseudolinguagem), instrução 88–89
- semente de gerador de número aleatório 202
- seno, cálculo de 293–297
- sequência de bits 46, 123
- sequência de escape 113–114
  - '\'' 114
  - '\"' 114
  - '\\' 114
  - '\0' 114, 438
  - '\a' 114
  - '\b' 114
  - '\n' 114
  - '\r' 114
  - '\t' 114
- sequência de instruções 91, 173
- série de Taylor para cálculo de seno 293–297
- seta, operador 512–513
- setlocale()**, função 453
- [**SHIFT**], tecla 55
- simplificações adotadas neste livro xl–xliii
- sin()**, função 297
- sintaxe 143
  - C, da linguagem xxxi
  - coloração de 52
  - erro de 63, 66, 143–146, 347
  - notação usada neste livro para xxxix
- sistema de execução 137
- sistema operacional
  - comando de, criação de 487–488
  - Linux xli, 52, 64
  - Mac OS X 52, 698
  - Posix de, padrão 52, 460
  - Unix xli
- site (deste livro) [www.ulysseso.com/ip](http://www.ulysseso.com/ip) xxxviii
- sizeof**, operador 142–143, 280, 379–382
- size\_t**, tipo 379–382, 444
- soma 81–82, 117
- soma de ponteiro com inteiro 382–383
- sorteio de bônus 214–215
- sqrt()**, função 85, 301
- srand()**, função 85, 202, 214–215, 302
- stack 656
- stack frame 656
- static**, qualificador 278–279, 281–282
- std=c99**, opção de compilação 57
- stderr**, variável (stream) 568
- stdin**, variável (stream) 568

**stdio.h**, cabeçalho 130, 444, 556  
 stdio, módulo de biblioteca 130, 444, 556  
**stdlib.h**, cabeçalho 129, 202  
**stdout**, variável (stream) 568  
**strcat()**, função 448–449  
**strchr()**, função 455–456  
**strcmp()**, função 450–454, 486  
**strcoll()**, função 452–454  
**strcpy()**, função 446–448  
 stream 556–557  
   arquivo e xli  
   binário 560  
   buffer associado a 557  
   entrada, de 561  
   erro em, indicador de 557  
   final de, indicador de 557  
   implementação de 557  
   padrão 567–568  
   ponteiro para 557  
   posição em, indicador de 557  
   saída, de 561  
**stderr** 568  
**stdin** 568  
**stdout** 568  
   texto, de 559  
 string 87, 438  
   alinhamento de 534–536  
   aparo de 481–483  
   arquivo de, escrita em 563  
   arquivo, representando nome de 559  
   array, armazenado em 438–439  
   array de 460–461  
   busca de caractere em 455–458  
   busca de string em 454–455  
   caractere e 439–442  
   caractere em, busca de 455–458  
   casamento de 454–455  
   centralização na tela de 474–475  
   comparação de 449–454, 486–487  
   comprimento de 444–446  
   concatenação de 114–115, 448–449  
   conceito de 87  
   constante 114–115, 439  
   cópia de 446–448  
   embaralho de 665–666  
   endereço, representado como 439  
   escrita em arquivo de 563  
   escrita na tela de 443–444  
**fopen()**, usado por 558  
   formatação, de 130  
   importância em programação de 438  
   inserção de string em 488–508  
   leitura via teclado de xlii, 586–590  
   numérico 470  
   número de, conversão em 466–468  
   ocorrência de caractere em 455–458  
   ordenação de 449–454

partes de, separação em 458–459  
 poda de 481–483  
 ponteiro e 439–442  
 remoção de caractere de 481–483  
 string em, busca de 454–455  
 string em, inserção de 488–508  
 tamanho de 444–446  
 teclado de, leitura via xlii, 586–590  
 tela de, escrita na 443–444  
 tokens de, separação em 458–459  
 vazio 442–443  
**string.h**, cabeçalho 129, 442  
**strlen()**, função 444–446  
**strrchr()**, função 456–458  
**strstr()**, função 454–455  
**strtod()**, função 466–468  
**strtok()**, função 458–459  
**struct**, definidor de tipo 511  
 subalgoritmo 77  
 subproblema 75  
 subprograma 250, 254  
 subprograma em linguagem algorítmica 270–272  
 subtração 81–82, 117  
   inteira de ponteiro 382–383  
   ponteiros, entre 383  
 subtraço (símbolo) xli, 110  
 sudo, comando (Unix/Linux/Mac OS) 697  
**switch-case**, instrução 190–194  
**system("pause")**, instrução 64

## T

[**TAB**], tecla 138–142  
 tabela-verdade 84, 119  
 tabuada de multiplicação 361–362  
 tabulação 138–140  
 tamanho de arquivo 611–615  
 tamanho de array 374  
   constante simbólica, definido com 376  
   corrupção de memória e 393–394  
   definição de array, em 374  
   definido em tempo de execução 650  
   definido em tempo de programação 650–651  
**fgets()**, usado com 589–590  
   iniciação e 377  
   iniciação, omitido em 377, 438  
   leitura de string, usado em 442, 586  
**LeString()**, usado com 442  
   multidimensional 399  
   número mágico e 376  
   parâmetro, recebido como 390  
**scanf()**, usado com 586  
   **sizeof** de, cálculo com 380–382  
 tamanho de bloco de memória 567, 593–594, 651–652,  
   653–654, 657  
 tamanho de identificador 111  
 tamanho de partição de arquivo 591, 596

tamanho de partição usada em execução de programa, de 655–656

tamanho de registro de arquivo 591, 596

tamanho de resultado de expressão 379–380

tamanho de texto digitado via teclado 601–603

tamanho de tipo 379–380

tamanho de variável 379–380

- estrutura 520–521
- fator de escala e 382
- sizeof**, calculado com 379–382
- size\_t** e 379
- string 444–446, 457
- união 520–521

Taylor, série de 293–297

teclado, leitura via 51

tela, congelamento da 64–65

tela e console 51

tela, escrita de dados na 51

telefone, teste do 328, 331

tentativa e erro, programação por 348

terminal de instrução 173

teste de algoritmo 96–107

teste de programa 136

teste do telefone 328, 331

teste exaustivo 136

**time()**, função 202, 302

**time.h**, cabeçalho 129, 202

tipo de dado 112–113

- agregado 374
- booleano 80
- char** 112
- char \*** 439
- conversão de 122–125
- definição de 510
- definido pelo programador 510
- derivado 112, 510
- double** 112–113
- double \*** 252
- embutido 112
- estrutura, de 510–511
- estruturado 374
- int** 112
- int \*** 251
- inteiro 80
- largura de 112
- primitivo 112
- real 80
- registro variante, de 522–524
- size\_t** 379–382
- typedef** em definição de, uso de 510
- união, de 520–521
- void** 256
- void \*** 654–655

token 458

**tolower()**, função 463

**toupper()**, função 463

traço (símbolo)

- acesso, usado em operador de 515–516
- decremento, usado em operador de 125–126
- estrutura, usado em acesso a 512–513
- inversão de sinal, usado como operador de 81, 117
- subtração, usado como operador de 81, 117

tradução de programa 47–49

tradutor 47–49

tralha (símbolo) 133

tratamento de exceção 259

três pontos (símbolo)

- elipse, usados como `xl`
- parâmetro, usados como 571

triângulo, checagem de lados de 208–210

troca de valores entre variáveis 99, 264–269

troco, cálculo de 421–424

truncamento 123, 352–354

Tudor, banco de dados da dinastia 601

**Tudor.bin**, arquivo 621–624

**Tudor.txt**, arquivo 601

**typedef** 510

## U

**%u**, especificador de formato 382

underline `xli`

underscore `xli`

união 520–521

- acesso a campo de 520
- compartilhamento de memória de 520–521
- definição de 520
- iniciação de 521, 524
- registro variante de, uso em 522–524

**union**, definidor de tipo 520–521

Unix, sistema operacional

- ambiente de trabalho no 52
- ar do, comando 696
- arquivo no, conceito de 556
- arquivo no, encerramento de leitura via teclado de 570
- canalização (pipe) no 604
- compilador GCC para 53
- cp do, comando 698
- [CTRL]+[D] no, significado de 570
- execução de programa no 64
- impressora como arquivo no, especificação de 558
- LEITURAFACIL** no, compilação com Clang da biblioteca 696–697
- LEITURAFACIL** no, compilação com GCC da biblioteca 696
- LEITURAFACIL** no, integração da biblioteca 698
- linha de comando no, argumento de 460
- Linux e `xli`
- make do, comando 695, 696
- makefile no, arquivo 696
- ranlib do, comando 698
- redirecionamento de entrada no 604
- redirecionamento de saída no 604
- rm do, comando 563

Unix, sistema operacional (*continuação*)  
 stream no 560  
 stream no, portabilidade de 556  
 usabilidade 50  
 usuário e programa minimalista xxxv

## V

validação de dados 285  
 data, de 363–366, 483–486  
 esvaziamento de buffer e 578–580  
 laço de repetição e 580–585  
 número de identificação, de 470–472  
 número inteiro, de 580–585, 691  
 número natural, de 284–286  
 número real, de 691–692  
 opção, de 693–694  
 string, de 692–693  
 valor absoluto 257, 298  
 valor booleano 82–83  
 valor de caractere 111–112  
 valor de variável 46, 66, 79, 94, 121  
 valor lógico 80, 82–83  
 variável 46  
 agregada 374  
 alocação de 650  
 anônima 253, 651  
 booleana 83  
 conceito de 46  
**const**, definida com 387–389  
 contagem, de 176, 181, 328  
 conteúdo de 46, 66, 79, 94, 121  
 declaração de 80  
 dinâmica 650  
 duração de 277–280  
 endereço de 250  
 escopo de 280–284  
 estática 650  
 estrutura 510–515  
 estruturada 374  
 global 280  
 heterogênea 374, 510  
 homogênea 374  
 iniciação de 121  
 local 282  
 lógica 83  
 não iniciada 147, 280  
 nome de 92, 110–111, 121, 329–333  
 nomenclatura de 121  
 ponteiro para 250–252  
 simbólica 47, 79  
**static**, definida com 278–279, 281–282  
 união 520–521  
 valor de 46, 66, 79, 94, 121  
 vazamento de memória 657, 664  
 verdadeiro (pseudolinguagem), constante lógica 82  
 violação de memória 465

vírgula, operador 201–202  
 vírgula (símbolo)  
 operador vírgula, usada como 201–202  
 separador de declarações, usada como 120, 201  
 separador de parâmetros, usada como 201, 260, 269  
 separador de variáveis em definição, usada como 120  
**(void)**, operador de conversão 124–125  
**void \*** (ponteiro para **void**), tipo 654–655  
**void**, tipo 256–257  
 alusão, uso em 270  
**main()**, usado com 270, 461  
 parâmetro de função, como 257  
 tipo de retorno de função, como 256

## W

"**w+b**", modo de abertura de arquivo 561, 599  
 "**w+**", modo de abertura de arquivo 560, 564, 599  
 "**w+t**", modo de abertura de arquivo 561, 599  
**-Wall**, opção de compilação 61  
**"wb"**, modo de abertura de arquivo 561, 594, 611, 623  
**while**, instrução 174–178, 185, 194–196, 199, 328  
 Windows, sistema operacional Microsoft xxxvii, xxxix  
 ambiente de trabalho no 52  
 arquivo no, desbloqueio de 563  
 arquivo no, encerramento de leitura via teclado de 570  
 canalização (pipe) no 604  
 CodeBlocks no, instalação de 54  
 [CTRL]+[C] no, significado de 65, 177  
 [CTRL]+[Z] no, significado de 570  
 execução de programa no 54–55, 64–65  
 impressora como arquivo no, especificação de 558  
**LEITURAFACIL** no, compilação da biblioteca 695  
**LEITURAFACIL** no, integração da biblioteca 697  
 linha de comando no, argumento de 460  
 redirecionamento de entrada no 604  
 redirecionamento de saída no 604  
 stream no, portabilidade de 556  
**"w"**, modo de abertura de arquivo 560, 563, 618  
**"wt"**, modo de abertura de arquivo 561  
[www.ulysseso.com/ip](http://www.ulysseso.com/ip), site xxxviii, 52, 54, 66, 128, 141

## Z

zumbi 396  
 pilha de execução, habitante da 394–397, 518