

Resposta da 7ª Lista de Exercícios

Assunto: Funções e passagem por referência com vetor e matriz (Tópico 7)

Resposta pergunta 1:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 void lePonto(double *x, double *y, int nPonto) {
5     printf(" Digite a coordenada x do %do ponto\n", nPonto);
6     scanf("%lf", x);
7     printf(" Digite a coordenada y do %do ponto\n", nPonto);
8     scanf("%lf", y);
9 }
10
11 double distPontos(double x1, double y1, double x2, double y2) {
12     double dxy = (x1 - x2)*(x1 - x2) + (y1 - y2)*(y1 - y2);
13     return sqrt(dxy);
14 }
15
16 int main() {
17     double x1, x2, y1, y2, dxy;
18     lePonto(&x1, &y1, 1);
19     lePonto(&x2, &y2, 2);
20     dxy = distPontos(x1, y1, x2, y2);
21     printf("%.2lf %.2lf %.2lf %.2lf \n", x1, y1, x2, y2);
22     printf(" Distancia = %.2lf\n", dxy);
23     return 0;
24 }
```

Listing 1: Resposta do exercício 1 codificado na linguagem C

Resposta pergunta 2:

```
1 #include <stdio.h>
2
3 int somatorio(int n) {
4     int soma = 0;
5     for (int i = 1; i <= n; i++) {
6         soma += i;
7     }
8     return soma;
9 }
10
11 int main() {
12     int n;
13     printf(" Digite o valor de n:\n");
14     scanf("%d", &n);
15     printf("O somatorio de 1 ate %d = %d\n", n, somatorio(n));
16     return 0;
17 }
```

Listing 2: Resposta do exercício 2 codificado na linguagem C

Resposta pergunta 3:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int ehPrimo(int n) {
5     if (n == 0 || n == 1 || n <= -1) {
6         return -1;
7     }
8     for (int i = 2; i <= sqrt(n); i++) {
9         if (n % i == 0) {
10            return 0;
11        }
12    }
13    return 1;
14 }
15
16 int main() {
17     int n;
18     printf("Digite o numero que deseja verificar se eh primo: \n");
19     scanf("%d", &n);
20     if (ehPrimo(n) == 1) {
21         printf("Numero Primo\n");
22     } else if (ehPrimo(n) == 0) {
23         printf("Numero Composto\n");
24     } else if (ehPrimo(n) == -1) {
25         printf("Nao eh primo nem composto\n");
26     }
27 }
```

Listing 3: Resposta do exercício 3 codificado na linguagem C

Resposta pergunta 4:

```
1 #include <stdio.h>
2 #include <math.h>
3
4 double leCoef(int nCoef) {
5     double x;
6     printf("Digite o %do coeficiente\n", nCoef);
7     scanf("%lf", &x);
8     return x;
9 }
10
11 double calcDelta(double a, double b, double c) {
12     double delta = (b*b) - 4*a*c;
13     if (delta < 0) {
14         printf("Nao possui raizes reais\n");
15     }
16     return (b*b) - 4*a*c;
17 }
18
19 double calcRaiz(double a, double b, double delta, int sinal) {
20     return (-b + sinal*sqrt(delta)) / 2*a;
21 }
```

```

23 int main() {
    double a, b, c, delta, r1, r2;
25 a = leCoef(1);
    b = leCoef(2);
27 c = leCoef(3);
    if (a == 0) {
29     printf("Nao eh equacao do 2o grau\n");
    } else {
31     delta = calcDelta(a, b, c);
        if (delta >= 0) {
33         r1 = calcRaiz(a, b, delta, 1);
            r2 = calcRaiz(a, b, delta, -1);
35         printf("As raizes sao %.2lf e %.2lf\n", r1, r2);
        }
37     }
}

```

Listing 4: Resposta do exercício 4 codificado na linguagem C

Resposta pergunta 5:

```

#include <stdio.h>
2
int fat(int n) {
4     if (n < 0) {
        return -1;
6     }
    if (n == 0) {
8         return 1;
    } else {
10        return n*fat(n-1);
    }
12 }

14 int main() {
    int n;
16    printf("Digite o valor de n: \n");
    scanf("%d", &n);
18    printf("O fatorial de %d = %d\n", n, fat(n));
    return 0;
20 }

```

Listing 5: Resposta do exercício 5 codificado na linguagem C

Resposta pergunta 6:

```

#include <stdio.h>
2 #include <math.h>

4 int fibonacci(int n, int x, int y) {
    if(n == 0) {
6         return 0;
    }
8     if(n > 1) {
        return y + fibonacci(n - 1, y, x + y);
    }
}

```

```

10 } else {
11     return 1;
12 }
13 }
14
15 int main() {
16     int n;
17     printf(" Digite a posicao: ");
18     scanf("%d",&n);
19     printf("O %do termo de fibonacci = %d\n", n, fibonacci(n, 1, 0));
20     return 0;
21 }

```

Listing 6: Resposta do exercício 6 codificado na linguagem C

Resposta pergunta 7:

```

1 #include <stdio.h>
2
3 int calcMDC(int a, int b) {
4     int resto = a % b;
5     while(resto != 0) {
6         a = b;
7         b = resto;
8         resto = a % b;
9     }
10    return b;
11 }
12
13 int calcMMC(int a, int b, int MDC) {
14     return (a * b)/MDC;
15 }
16
17 int main() {
18     int a, b, c;
19     int mdcAB, mdcABC, mdcABC2, mmcAB, mmcABC;
20     printf(" Calculo do MDC dos numeros A B C\n");
21     printf(" Digite os valores dos numeros de A, B e C\n");
22     scanf("%d%d%d", &a, &b, &c);
23     mdcAB = calcMDC(a, b);
24     mdcABC = calcMDC(mdcAB, c);
25     printf("\nO MDC e: %d\n", mdcABC);
26     mmcAB = calcMMC(a, b, mdcAB);
27     mdcABC2 = calcMDC(mmcAB, c);
28     mmcABC = calcMMC(mmcAB, c, mdcABC2);
29     printf("\nO MMC e: %d\n", mmcABC);
30     return 0;
31 }

```

Listing 7: Resposta do exercício 7 codificado na linguagem C

Resposta pergunta 8:

```

1 #include <stdio.h>
2 #include <stdio.h>
3

```

```

double calcRaiz(double y, int n) {
5   double x[n];
   x[0] = y/2;
7   int i;
   for (i = 1; i < n; i++) {
9       x[i] = (x[i - 1] * x[i - 1] + y) / (2 * x[i - 1]);
   }
11  return x[n - 1];
}

13
int main() {
15  double y, n, raiz;
   printf("Digite o numero: \n");
17  scanf("%lf", &y);
   printf("Digite a quantidade de aproximacoes: \n");
19  scanf("%lf", &n);
   if (y > 0 && n > 0) {
21     raiz = calcRaiz(y, n);
     printf("A raiz eh %.2lf \n", raiz);
23  }
   return 0;
25 }

```

Listing 8: Resposta do exercício 8 codificado na linguagem C

Resposta pergunta 9:

```

1  #include <stdio.h>
   #include <stdio.h>
3
   typedef struct {
5     char nome[128];
     int nPecas;
7     char sexo;
     double salario;
9     char classe;
   }Funcionario;
11
void setListaFunc(Funcionario lF[], int n) {
13  int i;
   for (i = 0; i < n; i++) {
15     printf("Digite o nome do funcionario: \n");
     scanf("%s", lF[i].nome);
17     getchar();
     printf("Digite o numero de pecas produzidas por %s: \n", lF[i].nome);
19     scanf("%d", &lF[i].nPecas);
     getchar();
21     printf("Digite o sexo de %s (F/M): \n", lF[i].nome);
     do {
23         printf("Digite OU F OU M: \n");
         scanf("%c", &lF[i].sexo);
25         getchar();
27     } while(lF[i].sexo != 'F' && lF[i].sexo != 'M');
   }
}

```

```

29 }
31 void setSalarios( Funcionario lF[], int n) {
32     int i;
33     double salarioMin = 880.00;
34     int pecas;
35     for (i = 0; i < n; i++) {
36         if (lF[i].nPecas <= 30) {
37             lF[i].salario = salarioMin;
38             lF[i].classe = 'A';
39         }
40         else if (lF[i].nPecas <= 31 && lF[i].nPecas <= 35) {
41             pecas = lF[i].nPecas - 30;
42             lF[i].salario = salarioMin + (0.03 * pecas * salarioMin);
43             lF[i].classe = 'B';
44         }
45         else if (lF[i].nPecas > 35) {
46             pecas = lF[i].nPecas - 30;
47             lF[i].salario = salarioMin + (0.05 * pecas * salarioMin);
48             lF[i].classe = 'C';
49         }
50     }
51 }

53 double calcFolha( Funcionario lF[], int n) {
54     int i;
55     double folhaPag = 0;
56     for (i = 0; i < n; i++) {
57         folhaPag += lF[i].salario;
58     }
59     return folhaPag;
60 }

61 double mediaFolhaM( Funcionario lF[], int n) {
62     int i;
63     double folhaPag = 0;
64     int nM = 0;
65     for (i = 0; i < n; i++) {
66         if (lF[i].sexo == 'M') {
67             folhaPag += lF[i].salario;
68             nM++;
69         }
70     }
71     return folhaPag/nM;
72 }

73 }

75 double mediaClasse( Funcionario lF[], int n, char classe) {
76     int i;
77     int media = 0;
78     int numClasse = 0;
79     for (i = 0; i < n; i++) {
80         if (lF[i].classe == classe && lF[i].sexo == 'F') {
81             media += lF[i].nPecas;

```

```

83         numClasse++;
84     }
85 }
86 if (numClasse != 0) {
87     return media/numClasse;
88 }
89 return 0;
90 }
91
92 void mediaTodasClassesM( Funcionario lF[], int n) {
93     double mediaA = mediaClasse(lF, n, 'A');
94     double mediaB = mediaClasse(lF, n, 'B');
95     double mediaC = mediaClasse(lF, n, 'C');
96     printf("A media de pecas feitas por mulheres na classe A foi: %lf \n",
97         mediaA);
98     printf("A media de pecas feitas por mulheres na classe B foi: %lf \n",
99         mediaB);
100    printf("A media de pecas feitas por mulheres na classe C foi: %lf \n",
101        mediaC);
102 }
103
104 int maiorSalario(Funcionario lF[], int n) {
105     double maior = lF[0].salario;
106     int i;
107     int posicao = 0;
108     for (i = 1; i < n; i++) {
109         if (lF[i].salario > maior) {
110             maior = lF[i].salario;
111             posicao = i;
112         }
113     }
114     return posicao;
115 }
116
117 void printFuncionario(Funcionario lF[], int posicao) {
118     printf("Nome: %s \n", lF[posicao].nome);
119     printf("Sexo: %c\n", lF[posicao].sexo);
120     printf("Numero de pecas produzidas: %d\n", lF[posicao].nPecas);
121     printf("Classe: %c\n", lF[posicao].classe);
122     printf("Salario: %lf\n", lF[posicao].salario);
123     printf("\n\n");
124 }
125
126 void printAll(Funcionario lF[], int n) {
127     int i;
128     for (i = 0; i < n; i++) {
129         printFuncionario(lF, i);
130     }
131 }
132
133 int main() {
134     int nFuncionarios;
135     int posicaoMaiorSal;
136     printf("Digite o numero de funcionarios: \n");

```

```

scanf("%d", &nFuncionarios);
135 getchar();
Funcionario listaFuncionarios[nFuncionarios];
137 int op;
do {
139     printf(" Digite o numero da operacao que deseja executar: \n");
scanf("%d", &op);
141     getchar();
switch (op) {
143         case 1:
setListaFunc(listaFuncionarios, nFuncionarios);
145         break;
case 2:
147         setSalarios(listaFuncionarios, nFuncionarios);
printAll(listaFuncionarios, nFuncionarios);
149         break;
case 3:
151         printf("O total em de pagamento eh: %lf \n", calcFolha(
listaFuncionarios, nFuncionarios));
break;
153         case 4:
printf("A media dos salarios dos homens eh: %lf \n",
155         mediaFolhaM(listaFuncionarios, nFuncionarios));
break;
157         case 5:
mediaTodasClassesM(listaFuncionarios, nFuncionarios);
break;
159         case 6:
printf("O Funcionario do mes eh: \n");
161         posicaoMaiorSal = maiorSalario(listaFuncionarios,
nFuncionarios);
printFuncionario(listaFuncionarios, posicaoMaiorSal);
163         break;
case 0:
165         printAll(listaFuncionarios, nFuncionarios);
printf("Encerrando o sistema... \n");
167         break;
default:
169         printf("ERRO: Digite uma opcao valida \n");
break;
171     }
} while (op);
173 return 0;
}

```

Listing 9: Resposta do exercício 9 codificado na linguagem C

Resposta pergunta 10:

```

#include <stdio.h>
2 #include <stdlib.h>
#include <time.h>
4 #define N 100

6 void imprimirMenu() {

```



```

8     printf("1 - Soma de vetores \n");
9     printf("2 - Produto vetorial (vetores no espaco)\n");
10    printf("3 - Produto vetor por escalar \n");
11    printf("0 - Sair \n\n");
12    printf("Digite sua opcao: ");
13 }
14 void printVetor(int n, int v1[]) {
15     for (int i = 0; i < n; i++) {
16         printf("%d ", v1[i]);
17     }
18     printf("\n");
19 }
20 void gerarVetor(int n, int v1[], int inicio, int fim, char nome) {
21     printf("Vetor %dx1 %c\n", n, nome);
22     for (int i = 0; i < n; i++) {
23         v1[i] = inicio + (rand() % (fim-inicio));
24         printf("%d ", v1[i]);
25     }
26     printf("\n");
27 }
28 }
29 void somaVetores(int n, int v1[], int v2[]) {
30     int v3[n];
31     printf("Resultado da soma de vetores \n");
32     for (int i = 0; i < n; i++) {
33         v3[i] = v1[i] + v2[i];
34         printf("%d ", v3[i]);
35     }
36     printf("\n");
37 }
38 }
39 void prodVetorial(int n, int v[], int w[]) {
40     int v3[n];
41     v3[0] = v[1] * w[2] - v[2] * w[1];
42     v3[1] = v[2] * w[0] - v[0] * w[2];
43     v3[2] = v[0] * w[1] - v[1] * w[0];
44     printf("Resultado do produto vetorial \n ");
45     printVetor(n, v3);
46 }
47 }
48 void prodVetorEscalar(int n, int v1[], int escalar) {
49     int v2[n];
50     for (int i = 0; i < n; i++) {
51         v2[i] = v1[i] * escalar;
52     }
53     printf("Resultado do produto vetor por escalar \n");
54     printVetor(n, v2);
55 }
56 }
57 int setDimensao() {
58     int n;
59     printf("Digite a dimensao do vetor: \n");
60 }

```

```

scanf("%d", &n);
62 return n;
}
64
int setIntIni() {
66     int i;
    printf("Digite o inicio do intervalo para gerar os valores dos vetores: \n
");
68     scanf("%d", &i);
    return i;
70 }

int setIntFim() {
72     int f;
74     printf("Digite o fim do intervalo para gerar os valores dos vetores: \n");
    scanf("%d", &f);
76     return f;
}
78

int main() {
80     srand(time(NULL));
    int op = 1;
82     do {
        imprimirMenu();
84         scanf("%d", &op);
        if (op == 1) {
86             printf("Dados dos Vetores 1 \n");
                int n = setDimensao();
88                 int i = setIntIni();
                int f = setIntFim();
90                 int v1[n], v2[n];
                gerarVetor(n, v1, i, f, 'A');
92                 gerarVetor(n, v2, i, f, 'B');
                somaVetores(n, v1, v2);
94             } else if (op == 2) {
                printf("Dados do Vetor 1\n");
96                 int n = 3;
                int i1 = setIntIni();
98                 int f1 = setIntFim();
                int v1[n];
100                gerarVetor(n, v1, i1, f1, 'A');
                printf("Dados do Vetor 2\n");
102                int i2 = setIntIni();
                int f2 = setIntFim();
104                int v2[n];
                gerarVetor(n, v2, i2, f2, 'B');
106                prodVetorial(n, v1, v2);
            } else if (op == 3) {
108                printf("Dados da Matriz 1\n");
                int n = setDimensao();
110                int i = setIntIni();
                int f = setIntFim();
112                int v1[n];
                gerarVetor(n, v1, i, f, 'A');

```

```

114         printf("Dados do Escalar\n ");
115         int escalar;
116         scanf("%d", &escalar);
117         prodVetorEscalar(n, v1, escalar);
118     } else {
119         printf("Opcao invalida\n\n ");
120     }
121 } while (op != 0);
122 return 0;
}

```

Listing 10: Resposta do exercício 10 codificado na linguagem C

Resposta pergunta 11:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4 #define N 100
5
6 void atraso(int milliseconds) {
7     long pause;
8     clock_t now, then;
9     pause = milliseconds*(CLOCKS_PER_SEC/1000);
10    now = then = clock();
11    while( (now-then) < pause )
12        now = clock();
13}
14
15 void imprimirMenu() {
16     printf("1 - Soma de matrizes \n");
17     printf("2 - Transposta de uma Matriz \n");
18     printf("3 - Produto matriz por vetor \n");
19     printf("4 - Produto de matrizes \n");
20     printf("0 - Sair \n");
21     printf("Digite sua opcao: \n");
22     printf("\n\n");
23}
24
25 void printMatriz(int m, int n, int m1[m][n]) {
26     for (int i = 0; i < m; i++) {
27         for (int j = 0; j < n; j++) {
28             printf("%d ", m1[i][j]);
29         }
30         printf("\n");
31     }
32     printf("\n");
33}
34
35 void gerarMatriz(int m, int n, int m1[][n], int inicio, int fim, char nome) {
36     srand(time(NULL));
37     printf("Matriz %dx%d %c\n", m, n, nome);
38     for (int i = 0; i < m; i++) {
39         for (int j = 0; j < n; j++) {
40             atraso(200);

```

```

41         m1[i][j] = inicio + (rand() % (fim-inicio));
42         printf("%d ", m1[i][j]);
43     }
44     printf("\n");
45 }
46 printf("\n");
47 }
48
49 void somaMatriz(int m, int n, int m1[][n], int m2[][n]) {
50     int m3[m][n];
51     printf("Resultado da soma de matrizes \n");
52     for (int i = 0; i < m; i++) {
53         for (int j = 0; j < n; j++) {
54             m3[i][j] = m1[i][j] + m2[i][j];
55             printf("%d ", m3[i][j]);
56         }
57         printf("\n");
58     }
59 }
60
61 void transpostaMatriz(int m, int n, int matriz[][n], int matrizT[][m]) {
62     printf("Resultado da transposicao de matriz \n");
63     for (int i = 0; i < m; i++) {
64         for (int j = 0; j < n; j++) {
65             matrizT[j][i] = matriz[i][j];
66         }
67     }
68 }
69
70 void multiMatriz(int m1, int n1, int m2, int n2, int mm1[][n1], int mm2[][n2])
71 {
72     int mm3[m1][n2];
73     if (n1 == m2) {
74         for (int i = 0; i < m1; i++) {
75             for (int j = 0; j < n2; j++) {
76                 mm3[i][j] = 0;
77             }
78             for (int i = 0; i < m1; i++) {
79                 for (int j = 0; j < n2; j++) {
80                     for (int k = 0; k < n1; k++) {
81                         mm3[i][j] = mm3[i][j] + (mm1[i][k] * mm2[k][j]);
82                     }
83                 }
84             }
85             printf("Resultado da multiplicacao \n\n ");
86             printMatriz(m1, n2, mm3);
87         } else {
88             printf("ERRO: Dimensoes Invalidas \n\n ");
89         }
90     }
91 }
92
93 int setLinhas() {
94     int m;

```

```

95     printf(" Digite a quantidade de linhas da matriz: \n");
96     scanf("%d", &m);
97     return m;
98 }
99 int setColunas() {
100     int n;
101     printf(" Digite a quantidade de colunas da matriz: \n");
102     scanf("%d", &n);
103     return n;
104 }
105
106 int setIntIni() {
107     int i;
108     printf(" Digite o inicio do intervalo para gerar os valores das matrizes: \
109     n");
110     scanf("%d", &i);
111     return i;
112 }
113
114 int setIntFim() {
115     int f;
116     printf(" Digite o fim do intervalo para gerar os valores das matrizes: \n")
117     ;
118     scanf("%d", &f);
119     return f;
120 }
121 int main() {
122     int op = 1;
123     int m, n, i, f;
124     int m1, n1, i1, f1;
125     int m2, n2, i2, f2;
126     do {
127         imprimirMenu();
128         scanf("%d", &op);
129         if (op == 1) {
130             m = setLinhas();
131             n = setColunas();
132             i = setIntIni();
133             f = setIntFim();
134             int m1[m][n], m2[m][n];
135             gerarMatriz(m, n, m1, i, f, 'A');
136             gerarMatriz(m, n, m2, i, f, 'B');
137             somaMatriz(m, n, m1, m2);
138         } else if (op == 2) {
139             m = setLinhas();
140             n = setColunas();
141             i = setIntIni();
142             f = setIntFim();
143             int m1[m][n];
144             int m2[n][m];
145             gerarMatriz(m, n, m1, i, f, 'A');

```

```

147     transpostaMatriz(m, n, m1, m2);
148     printMatriz(n, m, m2);
149 } else if (op == 3) {
150     printf("Dados da Matriz 1\n\n ");
151     m1 = setLinhas();
152     n1 = setColunas();
153     i1 = setIntIni();
154     f1 = setIntFim();
155     int mm1[m1][n1];
156     gerarMatriz(m1, n1, mm1, i1, f1, 'A');
157     printf("Dados do Vetor (Ou Matriz 1xN)\n\n ");
158     m2 = setLinhas();
159     n2 = 1;
160     i2 = setIntIni();
161     f2 = setIntFim();
162     int mm2[m2][n2];
163     gerarMatriz(m2, n2, mm2, i2, f2, 'v');
164     multiMatriz(m1, n1, m2, n2, mm1, mm2);
165 } else if (op == 4) {
166     printf("Dados da Matriz 1\n");
167     m1 = setLinhas();
168     n1 = setColunas();
169     i1 = setIntIni();
170     f1 = setIntFim();
171     int mm1[m1][n1];
172     gerarMatriz(m1, n1, mm1, i1, f1, 'A');
173     printf("Dados da Matriz 2\n");
174     m2 = setLinhas();
175     n2 = setColunas();
176     i2 = setIntIni();
177     f2 = setIntFim();
178     int mm2[m2][n2];
179     gerarMatriz(m2, n2, mm2, i2, f2, 'B');
180     multiMatriz(m1, n1, m2, n2, mm1, mm2);
181 } else {
182     printf("Opcao invalida\n\n ");
183 }
184 } while (op != 0);
185 return 0;
186 }

```

Listing 11: Resposta do exercício 11 codificado na linguagem C

Resposta pergunta 12:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 void preencheVetorRandom(int vetor[], int n, int min, int max){
6     for (int i = 0; i < n; i++){
7         vetor[i] = min + rand() % (max - min);
8     }
9 }

```

```

11 void imprimeVetor(int vetor [], int n){
12     for (int i = 0; i < n; i++){
13         printf("%d ", vetor[i]);
14     }
15     printf("\n");
16 }
17
18 void faz_algo_com_vetor1(int vetor [], int n){
19     for (int i = 1; i < n; i++){
20         for (int j = 0; j < n-1; j++){
21             if (vetor[j] > vetor[j+1]){
22                 int aux = vetor[j];
23                 vetor[j] = vetor[j+1];
24                 vetor[j+1] = aux;
25             }
26         }
27     }
28 }
29
30 int main(){
31     srand(time(NULL));
32     int n;
33     int min;
34     int max;
35     printf(" Digite uma quantidade de dados do vetor: ");
36     scanf("%d", &n);
37     printf(" Digite o valor minimo dos dados: ");
38     scanf("%d", &min);
39     printf(" Digite o valor maximo dos dados: ");
40     scanf("%d", &max);
41     int vetor[n];
42     preencheVetorRandom(vetor, n, min, max);
43     printf(" Vetor antes de chamar Faz Algo com Vetor 1: \n");
44     imprimeVetor(vetor, n);
45     faz_algo_com_vetor1(vetor, n);
46     printf(" Vetor depois de chamar Faz Algo com Vetor 1: \n");
47     imprimeVetor(vetor, n);
48     return 0;
49 }

```

Listing 12: Resposta do exercício 12 codificado na linguagem C

Resposta pergunta 13:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 void preencheVetorRandom(int vetor [], int n, int min, int max){
6     for (int i = 0; i < n; i++){
7         vetor[i] = min + rand() % (max - min);
8     }
9 }
10
11 void imprimeVetor(int vetor [], int n){

```

```

13     for (int i = 0; i < n; i++){
14         printf("%d ", vetor[i]);
15     }
16     printf("\n");
17 }
18
19 double media(int vetor [], int n){
20     double media = 0;
21     for (int i = 0; i < n; i++){
22         media += vetor[i];
23     }
24     return media/n;
25 }
26
27 void ordena(int vetor [], int n){
28     for (int i = 1; i < n; i++){
29         for (int j = 0; j < n-1; j++){
30             if (vetor[j] > vetor[j+1]){
31                 int aux = vetor[j];
32                 vetor[j] = vetor[j+1];
33                 vetor[j+1] = aux;
34             }
35         }
36     }
37 }
38
39 int mediana(int vetor [], int n){
40     ordena(vetor, n);
41     return vetor[n/2];
42 }
43
44 int idMaior(int vetor [], int n){
45     int maior = vetor[0];
46     int idMaior = 0;
47     for (int i = 1; i < n; i++){
48         if (vetor[i] > maior){
49             maior = vetor[i];
50             idMaior = i;
51         }
52     }
53     return idMaior;
54 }
55
56 int moda(int vetor [], int n){
57     int freq[n];
58     for (int i = 0; i < n; i++){
59         freq[i] = 0;
60     }
61     for (int i = 0; i < n; i++){
62         for (int j = 0; j < n; j++){
63             if (i != j){
64                 if (vetor[i] == vetor[j]){
65                     freq[i]++;
66                 }
67             }
68         }
69     }

```



```

    }
67 }
    }
69 int id = idMaior(freq, n);
    return vetor[id];
71 }

73 int main(){
    srand(time(NULL));
75 int n;
    int min;
77 int max;
    printf(" Digite uma quantidade de dados acima de 100: ");
79 scanf("%d", &n);
    printf(" Digite o valor minimo dos dados: ");
81 scanf("%d", &min);
    printf(" Digite o valor maximo dos dados: ");
83 scanf("%d", &max);
    int vetor[n];
85 preencheVetorRandom(vetor, n, min, max);
    imprimeVetor(vetor, n);
87 double vmedia = media(vetor, n);
    int vmediana = mediana(vetor, n);
89 int vmoda = moda(vetor, n);
    printf("\nMedia: %.2lf \nMediana: %d \nModa: %d", vmedia, vmediana, vmoda);
91 return 0;
}

```

Listing 13: Resposta do exercício 13 codificado na linguagem C

Resposta pergunta 16:

```

#include <stdio.h>
2
int checkAno(int ano) {
4     if (ano > 0 && ano < 10000) {
        return 1;
6     }
    return 0;
8 }

10 int checkAnoB(int ano) {
    if (ano % 400 == 0) {
12     return 1;
    } else if ((ano % 4 == 0) && (ano % 100 != 0)) {
14     return 1;
    }
16     return 0;
}

18
int checkMes(int mes) {
20     if (mes > 0 && mes < 13) {
        return 1;
22     }
    return 0;
}

```

```

24 }
26 int diasDoMes(int mes, int ano) {
28     if (checkMes(mes)) {
30         if (mes == 1) {
32             return 31;
34         }
36         if (mes == 2) {
38             if (checkAnoB(ano)) {
40                 return 29;
42             }
44             return 28;
46         } else if (mes == 3) {
48             return 31;
50         } else if (mes == 4) {
52             return 30;
54         } else if (mes == 5) {
56             return 31;
58         } else if (mes == 6) {
60             return 30;
62         } else if (mes == 7) {
64             return 31;
66         } else if (mes == 8) {
68             return 31;
70         } else if (mes == 9) {
72             return 30;
74         } else if (mes == 10) {
76             return 31;
78         } else if (mes == 11) {
80             return 30;
82         } else if (mes == 12) {
84             return 31;
86         }
88     }
90     return 0;
92 }
94
96 int validateDia(int dia, int mes, int ano) {
98     if (checkAno(ano)) {
100         if (checkMes(mes)) {
102             if (dia > 0 && dia <= diasDoMes(mes, ano)) {
104                 return 1;
106             }
108         }
110     }
112     return 0;
114 }
116
118 int main() {
120     int d, m, a;
122     printf(" Digite o ano: \n");
124     scanf("%d", &a);
126     printf(" Digite o mes: \n");
128     scanf("%d", &m);

```

```

78     printf("Digite o dia: \n");
       scanf("%d", &d);
80     if (validateDia(d, m, a)) {
           printf("A data %d do %d de %d eh valida\n", d, m, a);
82     } else {
           printf("A data %d do %d de %d eh invalida\n", d, m, a);
84     }
       return 0;
86 }

```

Listing 14: Resposta do exercício 16 codificado na linguagem C

Resposta pergunta 17:

```

#include <stdio.h>
2 #include <stdlib.h>

4 void escreverTabuleiro(char matriz[][3]) {
     for (int i = 0; i < 3; i++) {
6         for (int j = 0; j < 3; j++) {
             if (j % 3 == 0){
8                 printf("|");
             }
10            if (matriz[i][j] == '-'){
                printf("-|");
12            } else if (matriz[i][j] == 'x'){
                printf("x|");
14            } else if (matriz[i][j] == 'o'){
                printf("o|");
16            }
            }
18            if (i < 2){
                printf("\n-----\n");
20            }
        }
22    printf("\n");
    }

24 // -1 indica que alguem ganhou
26 int verificaJogo(char matriz[][3]) {
     int result = 0;
28     if (matriz[0][0] == matriz[0][1] && matriz[0][1] == matriz[0][2] && matriz
        [0][0] != '-')
         result = -1;
30     else if (matriz[1][0] == matriz[1][1] && matriz[1][1] == matriz[1][2] &&
        matriz[1][0] != '-')
         result = -1;
32     else if (matriz[2][0] == matriz[2][1] && matriz[2][1] == matriz[2][2] &&
        matriz[2][0] != '-')
         result = -1;
34     else if (matriz[0][0] == matriz[1][0] && matriz[1][0] == matriz[2][0] &&
        matriz[0][0] != '-')
         result = -1;
36     else if (matriz[0][1] == matriz[1][1] && matriz[1][1] == matriz[2][1] &&
        matriz[0][1] != '-')

```

```

    result = -1;
38 else if (matriz[0][2] == matriz[1][2] && matriz[1][2] == matriz[2][2] &&
    matriz[0][2] != '-')
    result = -1;
40 else if (matriz[0][0] == matriz[1][1] && matriz[1][1] == matriz[2][2] &&
    matriz[0][0] != '-')
    result = -1;
42 else if (matriz[0][2] == matriz[1][1] && matriz[1][1] == matriz[2][0] &&
    matriz[0][2] != '-')
    result = -1;
44 return result;
}
46
void preencheVazio(char matriz[][3]) {
48     for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
50             matriz[i][j] = '-';
        }
52     }
}
54
void preencherTabuleiro(char matriz[][3], int movLinha, int movColuna, int
    player) {
56     if (player == 0) {
        matriz[movLinha][movColuna] = 'x';
58     } else {
        matriz[movLinha][movColuna] = 'o';
60     }
}
62
int main() {
64     char matriz[3][3];
    int counter = 0;
66     int result = 0;
    int pos;
68     int player = 0;
    preencheVazio(matriz);
70     while (counter < 9 && result == 0) { //ninguem ganhou ainda
        do {
72             printf("Digite uma posicao vazia de 0 a 8\n");
            scanf("%d", &pos);
74         } while (pos > 8 || pos < 0 || matriz[pos / 3][pos % 3] != '-');
        preencherTabuleiro(matriz, pos/3, pos%3, player);
76         result = verificaJogo(matriz);
        if (result == -1) {
78             if (player == 0)
                result = 1; //jogador 1 ganhou
80             else
                result = 2; //jogador 2 ganhou
82         }
        escreverTabuleiro(matriz);
84         player = ~player; //not
        counter++;
86     }
}

```

```
88     if (result == 1){
90         printf("Jogador x ganhou\n");
92     } else if (result == 2) {
94         printf("Jogador o ganhou\n");
96     } else {
98         printf("Velha\n");
100     }
102     return 0;
104 }
```

Listing 15: Resposta do exercício 17 codificado na linguagem C