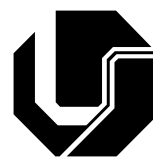




**Universidade Federal de Uberlândia**  
Faculdade de Engenharia Elétrica  
Experimental de Metrologia



### **Programação Procedimental**

*Lista 3 - Laboratório*

Dr. Marcelo Barros de Almeida

Matheus Souza Da Costa

12021EEL021

# Lista de ilustrações

Figura 1 – Questão 1 . . . . .	4
Figura 2 – Questão 2 . . . . .	6
Figura 3 – Questão 3 . . . . .	7
Figura 4 – Questão 4 . . . . .	8
Figura 5 – Questão 5 . . . . .	9
Figura 6 – Questão 6 . . . . .	10
Figura 7 – Questão 7 . . . . .	11
Figura 8 – Questão 8 . . . . .	12
Figura 9 – Questão 9 . . . . .	13
Figura 10 – Questão 10 . . . . .	14

# Sumário

1	Questão . . . . .	4
2	Questão . . . . .	4
3	Questão . . . . .	5
4	Questão . . . . .	5
5	Questão . . . . .	5
6	Questão . . . . .	5
7	Questão . . . . .	5
8	Questão . . . . .	5
9	Questão . . . . .	5
10	Questão . . . . .	5

---

## 1 Questão

Resposta conforme imagem 1



```
1  /*
2  * Arquivo: questao1.c
3  * Compilação: gcc -std=c99 questao1.c -o questao1
4  * Autor: Matheus Souza Da Costa 12021EEL021
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(void) {
11     #if defined(_WIN32) || defined(_WIN64)
12         system("chcp 65001"); // troca para UTF-8
13     #endif
14
15     int numeros[10];
16     int maior = 0;
17     int posicao = 0;
18
19     // Leitura dos números
20     printf("Digite 10 números inteiros:\n");
21     for (int i = 0; i < 10; i++) {
22         scanf("%d", &numeros[i]);
23
24         // Verifica se o número lido é maior do que o maior número atual
25         if (numeros[i] > maior) {
26             maior = numeros[i];
27             posicao = i;
28         }
29     }
30
31     // Imprime a posição do maior número
32     printf("A posição do maior número (%d) é: %d\n", maior, posicao);
33
34     return EXIT_SUCCESS;
35 }
36
```

Figura 1 – Questão 1

## 2 Questão

Resposta conforme imagem 2

---

### **3 Questão**

Resposta conforme imagem 3

### **4 Questão**

Resposta conforme imagem 4

### **5 Questão**

Resposta conforme imagem 5

### **6 Questão**

Resposta conforme imagem 6

### **7 Questão**

Resposta conforme imagem 7

### **8 Questão**


Resposta conforme imagem 8

### **9 Questão**

Resposta conforme imagem 9

### **10 Questão**

Resposta conforme imagem 10



```
1  /*
2   * Arquivo: questao2.c
3   * Compilação: gcc -std=c99 questao2.c -o questao2
4   * Autor: Matheus Souza Da Costa 12021EEL021
5   */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(void) {
11     #if defined(_WIN32) || defined(_WIN64)
12         system("chcp 65001"); // troca para UTF-8
13     #endif
14
15     float numeros[10];
16     float soma = 0.0;
17     float media;
18
19     // Leitura dos números
20     printf("Digite 10 números de ponto flutuante:\n");
21     for (int i = 0; i < 10; i++) {
22         scanf("%f", &numeros[i]);
23         soma += numeros[i];
24     }
25
26     // Cálculo da média
27     media = soma / 10.0;
28
29     // Imprime a média
30     printf("A média dos números é: %.2f\n", media);
31
32
33     return EXIT_SUCCESS;
34 }
```

Figura 2 – Questão 2

```

1  /*
2  * Arquivo: questao3.c
3  * Compilação: gcc -std=c99 questao3.c -o questao3
4  * Autor: Matheus Souza Da Costa 12021EEL021
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(void) {
11     #if defined(_WIN32) || defined(_WIN64)
12         system("chcp 65001"); // troca para UTF-8
13     #endif
14
15
16     int numeros[10];
17     int x;
18
19     // Leitura dos 10 números inteiros
20     printf("Digite 10 números inteiros:\n");
21     for (int i = 0; i < 10; i++) {
22         scanf("%d", &numeros[i]);
23     }
24
25     // Leitura do número inteiro x
26     printf("Digite o número inteiro x:\n");
27     scanf("%d", &x);
28
29     // Procura por dois números cuja multiplicação seja igual a x
30     int encontrou = 0;
31     for (int i = 0; i < 10; i++) {
32         for (int j = i + 1; j < 10; j++) {
33             if (numeros[i] * numeros[j] == x) {
34                 printf("Os números %d e %d têm uma multiplicação igual a %d.\n", numeros[i], numeros[j], x);
35                 encontrou = 1;
36             }
37         }
38     }
39
40     // Se nenhum par de números foi encontrado
41     if (!encontrou) {
42         printf("Não existem dois números cuja multiplicação seja igual a %d.\n", x);
43     }
44
45     return EXIT_SUCCESS;
46 }
47

```

Figura 3 – Questão 3

```

1  /*
2  * Arquivo: questao4.c
3  * Compilação: gcc -std=c99 questao4.c -o questao4
4  * Autor: Matheus Souza Da Costa 12021EEL021
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(void) {
11     #if defined(_WIN32) || defined(_WIN64)
12         system("chcp 65001"); // troca para UTF-8
13     #endif
14
15     int n, m;
16
17     // Leitura do tamanho das sequências
18     printf("Digite o tamanho da primeira sequência (n): ");
19     scanf("%d", &n);
20     printf("Digite o tamanho da segunda sequência (m): ");
21     scanf("%d", &m);
22
23     // Verificação se n <= m
24     if (n > m) {
25         printf("Erro: O tamanho da primeira sequência (n) deve ser menor ou igual ao tamanho da segunda sequência (m).\n");
26         return 1; // Encerra o programa com erro
27     }
28
29     // Leitura dos elementos das sequências
30     int v1[n], v2[m];
31     printf("Digite os elementos da primeira sequência (v1):\n");
32     for (int i = 0; i < n; i++) {
33         scanf("%d", &v1[i]);
34     }
35     printf("Digite os elementos da segunda sequência (v2):\n");
36     for (int i = 0; i < m; i++) {
37         scanf("%d", &v2[i]);
38     }
39
40     // Contagem de ocorrências de v1 em v2
41     int ocorrencias = 0;
42     for (int i = 0; i <= m - n; i++) {
43         int igual = 1;
44         for (int j = 0; j < n; j++) {
45             if (v1[j] != v2[i + j]) {
46                 igual = 0;
47                 break;
48             }
49         }
50         if (igual) {
51             ocorrencias++;
52         }
53     }
54
55     // Imprime o resultado
56     printf("A primeira sequência ocorre %d vezes na segunda sequência.\n", ocorrencias);
57     /* -- seu código aqui -- */
58
59     return EXIT_SUCCESS;
60 }
61

```

Figura 4 – Questão 4



```

1  /*
2  * Arquivo: questao5.c
3  * Compilação: gcc -std=c99 questao5.c -o questao5
4  * Autor: Matheus Souza Da Costa 12021EEL021
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(void) {
11     #if defined(_WIN32) || defined(_WIN64)
12         system("chcp 65001"); // troca para UTF-8
13     #endif
14
15     int n1, n2;
16
17     // Leitura do tamanho das sequências
18     printf("Digite o tamanho da primeira sequência (n1): ");
19     scanf("%d", &n1);
20     printf("Digite o tamanho da segunda sequência (n2): ");
21     scanf("%d", &n2);
22
23     // Verificação se o tamanho das sequências é válido
24     if (n1 > 50 || n2 > 50 || n1 < 0 || n2 < 0) {
25         printf("Erro: Tamanho das sequências deve ser no máximo 50 e positivo.\n");
26         return 1; // Encerra o programa com erro
27     }
28
29     // Leitura dos elementos das sequências
30     int v1[n1], v2[n2];
31     printf("Digite os elementos da primeira sequência (v1) em ordem crescente:\n");
32     for (int i = 0; i < n1; i++) {
33         scanf("%d", &v1[i]);
34     }
35     printf("Digite os elementos da segunda sequência (v2) em ordem crescente:\n");
36     for (int i = 0; i < n2; i++) {
37         scanf("%d", &v2[i]);
38     }
39
40     // Intercalação das sequências em v3
41     int v3[n1 + n2];
42     int i = 0, j = 0, k = 0;
43
44     while (i < n1 && j < n2) {
45         if (v1[i] < v2[j]) {
46             v3[k++] = v1[i++];
47         } else {
48             v3[k++] = v2[j++];
49         }
50     }
51
52     // Copia os elementos restantes de v1 e v2, se houverem
53     while (i < n1) {
54         v3[k++] = v1[i++];
55     }
56
57     while (j < n2) {
58         v3[k++] = v2[j++];
59     }
60
61     // Imprime o vetor intercalado v3
62     printf("Vetor intercalado em ordem crescente (v3):\n");
63     for (int i = 0; i < n1 + n2; i++) {
64         printf("%d ", v3[i]);
65     }
66     printf("\n");
67
68     return EXIT_SUCCESS;
69 }
70
71

```

Figura 5 – Questão 5

```

1  /*
2  * Arquivo: questao6.c
3  * Compilação: gcc -std=c99 questao6.c -o questao6
4  * Autor: Matheus Souza Da Costa 12021EEL021
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(void) {
11     #if defined(_WIN32) || defined(_WIN64)
12         system("chcp 65001"); // troca para UTF-8
13     #endif
14
15     int m, n;
16
17     // Leitura das dimensões da matriz
18     printf("Digite o número de linhas (m): ");
19     scanf("%d", &m);
20     printf("Digite o número de colunas (n): ");
21     scanf("%d", &n);
22
23     // Verificação das dimensões da matriz
24     if (m <= 0 || n <= 0 || m > 100 || n > 100) {
25         printf("Erro: As dimensões da matriz devem estar entre 1 e 100.\n");
26         return 1; // Encerra o programa com erro
27     }
28
29     // Leitura da matriz
30     int matriz[m][n];
31     printf("Digite os elementos da matriz %dx%d:\n", m, n);
32     for (int i = 0; i < m; i++) {
33         for (int j = 0; j < n; j++) {
34             scanf("%d", &matriz[i][j]);
35         }
36     }
37
38     // Inicialização das variáveis para encontrar o elemento com maior e menor frequência
39     int elementoMenorFreq, elementoMaiorFreq;
40     int menorFrequencia = m * n;
41     int maiorFrequencia = 0;
42
43     // Contagem das frequências dos elementos
44     for (int i = 0; i < m; i++) {
45         for (int j = 0; j < n; j++) {
46             int elementoAtual = matriz[i][j];
47             int frequenciaAtual = 1;
48
49             // Verifica a frequência do elemento atual em toda a matriz
50             for (int x = 0; x < m; x++) {
51                 for (int y = 0; y < n; y++) {
52                     if (x != i || y != j) {
53                         if (matriz[x][y] == elementoAtual) {
54                             frequenciaAtual++;
55                         }
56                     }
57                 }
58             }
59
60             // Verifica se a frequência atual é a menor ou a maior
61             if (frequenciaAtual < menorFrequencia) {
62                 menorFrequencia = frequenciaAtual;
63                 elementoMenorFreq = elementoAtual;
64             }
65             if (frequenciaAtual > maiorFrequencia) {
66                 maiorFrequencia = frequenciaAtual;
67                 elementoMaiorFreq = elementoAtual;
68             }
69         }
70     }
71
72     // Imprime os elementos com menor e maior frequência
73     printf("Elemento com menor frequência: %d (ocorreu %d vezes)\n", elementoMenorFreq, menorFrequencia);
74     printf("Elemento com maior frequência: %d (ocorreu %d vezes)\n", elementoMaiorFreq, maiorFrequencia);
75
76     return EXIT_SUCCESS;
77 }
78
79

```

Figura 6 – Questão 6

```

1  /*
2  * Arquivo: questao7.c
3  * Compilação: gcc -std=c99 questao7.c -o questao7
4  * Autor: Matheus Souza Da Costa 12021EE1021
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(void) {
11     #if defined(_WIN32) || defined(_WIN64)
12         system("chcp 65001"); // troca para UTF-8
13     #endif
14
15     int m, n;
16
17     // Leitura das dimensões da matriz
18     printf("Digite o número de linhas (m): ");
19     scanf("%d", &m);
20     printf("Digite o número de colunas (n): ");
21     scanf("%d", &n);
22
23     // Verificação das dimensões da matriz
24     if (m <= 0 || n <= 0 || m > 100 || n > 100) {
25         printf("Erro: As dimensões da matriz devem estar entre 1 e 100.\n");
26         return 1; // Encerra o programa com erro
27     }
28
29     // Leitura da matriz de números reais
30     double matriz[m][n];
31     printf("Digite os elementos da matriz %dx%d:\n", m, n);
32     for (int i = 0; i < m; i++) {
33         for (int j = 0; j < n; j++) {
34             scanf("%lf", &matriz[i][j]);
35         }
36     }
37
38     // Verificação de elementos repetidos
39     int encontrouRepetido = 0;
40
41     for (int i = 0; i < m; i++) {
42         for (int j = 0; j < n; j++) {
43             double elementoAtual = matriz[i][j];
44
45             // Verifica se o elemento atual se repete em toda a matriz
46             for (int x = 0; x < m; x++) {
47                 for (int y = 0; y < n; y++) {
48                     if (x != i || y != j) {
49                         if (matriz[x][y] == elementoAtual) {
50                             encontrouRepetido = 1;
51                             break;
52                         }
53                     }
54                 }
55                 if (encontrouRepetido) {
56                     break;
57                 }
58             }
59             if (encontrouRepetido) {
60                 break;
61             }
62         }
63         if (encontrouRepetido) {
64             break;
65         }
66     }
67
68     if (encontrouRepetido) {
69         printf("Existem elementos repetidos na matriz.\n");
70     } else {
71         printf("Não existem elementos repetidos na matriz.\n");
72     }
73
74     return EXIT_SUCCESS;
75 }
76

```

Figura 7 – Questão 7

```

1  /*
2   * Arquivo: questao8.c
3   * Compilação: gcc -std=c99 questao8.c -o questao8
4   * Autor: Matheus Souza Da Costa 12021EEL021
5   */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(void) {
11     #if defined(_WIN32) || defined(_WIN64)
12         system("chcp 65001"); // troca para UTF-8
13     #endif
14
15     int matriz[4][4];
16     int transposta[4][4];
17
18     // Leitura dos elementos da matriz
19     printf("Digite os elementos da matriz 4x4:\n");
20     for (int i = 0; i < 4; i++) {
21         for (int j = 0; j < 4; j++) {
22             scanf("%d", &matriz[i][j]);
23         }
24     }
25
26     // Cálculo da matriz transposta
27     for (int i = 0; i < 4; i++) {
28         for (int j = 0; j < 4; j++) {
29             transposta[j][i] = matriz[i][j];
30         }
31     }
32
33     // Impressão da matriz original
34     printf("Matriz original:\n");
35     for (int i = 0; i < 4; i++) {
36         for (int j = 0; j < 4; j++) {
37             printf("%d ", matriz[i][j]);
38         }
39         printf("\n");
40     }
41
42     // Impressão da matriz transposta
43     printf("Matriz transposta:\n");
44     for (int i = 0; i < 4; i++) {
45         for (int j = 0; j < 4; j++) {
46             printf("%d ", transposta[i][j]);
47         }
48         printf("\n");
49     }
50
51     return EXIT_SUCCESS;
52 }
53

```

Figura 8 – Questão 8

```

1  /*
2  * Arquivo: questao9.c
3  * Compilação: gcc -std=c99 questao9.c -o questao9
4  * Autor: Matheus Souza Da Costa 12021EEL021
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(void) {
11     #if defined(_WIN32) || defined(_WIN64)
12         system("chcp 65001"); // troca para UTF-8
13     #endif
14
15     int n; // Tamanho da matriz (n x n)
16
17     // Leitura do tamanho da matriz
18     printf("Digite o tamanho da matriz (n x n): ");
19     scanf("%d", &n);
20
21     int matriz[n][n];
22
23     // Leitura dos elementos da matriz
24     printf("Digite os elementos da matriz %dx%d:\n", n, n);
25     for (int i = 0; i < n; i++) {
26         for (int j = 0; j < n; j++) {
27             scanf("%d", &matriz[i][j]);
28         }
29     }
30
31     // Cálculo da soma da primeira linha (usaremos isso como referência)
32     int soma_referencia = 0;
33     for (int j = 0; j < n; j++) {
34         soma_referencia += matriz[0][j];
35     }
36
37     // Verificação das somas das linhas e colunas
38     for (int i = 1; i < n; i++) {
39         int soma_linha = 0;
40         int soma_coluna = 0;
41
42         for (int j = 0; j < n; j++) {
43             soma_linha += matriz[i][j];
44             soma_coluna += matriz[j][i];
45         }
46
47         if (soma_linha != soma_referencia || soma_coluna != soma_referencia) {
48             printf("Nao e um quadrado magico.\n");
49             return 0;
50         }
51     }
52
53     // Verificação da soma da diagonal principal
54     int soma_diag_principal = 0;
55     for (int i = 0; i < n; i++) {
56         soma_diag_principal += matriz[i][i];
57     }
58
59     if (soma_diag_principal != soma_referencia) {
60         printf("Nao e um quadrado magico.\n");
61         return 0;
62     }
63
64     // Verificação da soma da diagonal secundária
65     int soma_diag_secundaria = 0;
66     for (int i = 0; i < n; i++) {
67         soma_diag_secundaria += matriz[i][n - 1 - i];
68     }
69
70     if (soma_diag_secundaria != soma_referencia) {
71         printf("Nao e um quadrado magico.\n");
72         return 0;
73     }
74
75     // Se passou por todas as verificações, é um quadrado mágico
76     printf("E um quadrado magico.\n");
77     return EXIT_SUCCESS;
78 }
79

```

Figura 9 – Questão 9

```

1  /*
2  * Arquivo: questao10.c
3  * Compilação: gcc -std=c99 questao10.c -o questao10
4  * Autor: Matheus Souza Da Costa 12021EEL021
5  */
6
7  #include <stdio.h>
8  #include <stdlib.h>
9
10 int main(void) {
11     #if defined(_WIN32) || defined(_WIN64)
12         system("chcp 65001"); // troca para UTF-8
13     #endif
14
15     int n;
16
17     // Leitura do tamanho das matrizes
18     printf("Digite o tamanho das matrizes (n x n): ");
19     scanf("%d", &n);
20
21     // Declaração das matrizes A, B e C
22     int A[n][n], B[n][n], C[n][n];
23
24     // Leitura dos elementos da matriz A
25     printf("Digite os elementos da matriz A %dx%d:\n", n, n);
26     for (int i = 0; i < n; i++) {
27         for (int j = 0; j < n; j++) {
28             scanf("%d", &A[i][j]);
29         }
30     }
31
32     // Leitura dos elementos da matriz B
33     printf("Digite os elementos da matriz B %dx%d:\n", n, n);
34     for (int i = 0; i < n; i++) {
35         for (int j = 0; j < n; j++) {
36             scanf("%d", &B[i][j]);
37         }
38     }
39
40     // Cálculo da matriz C = A x B
41     for (int i = 0; i < n; i++) {
42         for (int j = 0; j < n; j++) {
43             C[i][j] = 0;
44             for (int k = 0; k < n; k++) {
45                 C[i][j] += A[i][k] * B[k][j];
46             }
47         }
48     }
49
50     // Impressão da matriz C
51     printf("Matriz resultante C %dx%d:\n", n, n);
52     for (int i = 0; i < n; i++) {
53         for (int j = 0; j < n; j++) {
54             printf("%d ", C[i][j]);
55         }
56         printf("\n");
57     }
58     return EXIT_SUCCESS;
59 }
60

```

Figura 10 – Questão 10