

Exercícios

• Matrizes

1. Realize o teste de mesa dos algoritmos a seguir, indique o valor das variáveis e o que será apresentado na tela:

ALGORITMO Matriz1

VAR

INTEIRO: mat[3][2] = {{7,6},{8,4},{1,10}};

INTEIRO: i, j;

INÍCIO

i ← 0;

j ← 1;

mat[i][j] ← mat[i][j] + j;

mat[j][i] ← mat[j][i] + mat[i][j];

mat[i][j-1] ← mat[j][j];

mat[j+1][j] ← i + j;

mat[j][j] ← mat[i][j] * mat[j][i];

PARA(i ← 0; i < 3; i ← i + 1) FAÇA

PARA(j ← 0; j < 2; j ← j + 1) FAÇA

ESCREVA(mat[i][j]);

FIM_PARA

FIM_PARA

FIM.

Linha	mat[3][2]	i	j	print
1				
2				
3	{7,6},{8,4},{1,10}			
4				
5		0		
6			1	
7	{7,7},{8,4},{1,10}			
8	{7,7},{14,4},{1,10}			
9	{4,7},{14,4},{1,10}			
10	{4,7},{14,48},{1,10}			
11				
12				
13	{4,7},{14,48},{1,10}			
14				
15				
16				4
17				7
18				14
20				48
21				1
22				10
23				
24				

ALGORITMO Matriz2

```

VAR
    INTEIRO: mat[3][5] = {{1, 7, 3, 4, 5},{2, 6, 5, 1, 0},{4, 9, 0,
3, 6}};
    INTEIRO: vet[3], i, j;
INÍCIO
    PARA(i ← 0; i < 3; i ← i + 1) FAÇA
        vet[i] ← 0;
        PARA(j ← 0; j < 5; j ← j + 1) FAÇA
            vet[i] ← vet[i] + mat[i][j];
        FIM_PARA
    FIM_PARA
    PARA(i ← 0; i < 3; i ← i + 1) FAÇA
        PARA(j ← 0; j < 5; j ← j + 1) FAÇA
            ESCREVA(mat[i][j]);
        FIM_PARA;
        ESCREVA(vet[i]);
    FIM_PARA
FIM.

```

Linha	Mat[3][5]	vet[3]	i	j	print
1					
2					
3	{1,7,3,4,5},{2,6,5,1,0},{4,9,0,3,6}				
4					
5					
6					
7					
8		vet[0]=0			
9					
10		vet[0]=1			
11		vet[0]=8			
12		vet[0]=11			
13		vet[0]=15			
14		vet[0]=20			
15					
16		vet[1]=2			
17		vet[1]=8			
18		vet[1]=13			
20		vet[1]=14			
21		vet[1]=14			
22					
23		vet[2]=4			
24		vet[2]=13			
25		vet[2]=13			
26		vet[2]=16			
27		vet[2]=22			
28					
29					
30					
31					1
32					7
33					3
34					4
35					2
36					20
37					2
38					6
39					5

40					1
41					0
42					14
43					4
44					9
45					0
46					3
47					6
48					22
49					

2. Elabore um algoritmo que leia uma matriz A(5 x 5) do teclado e crie uma matriz B onde cada elemento é o triplo do elemento correspondente de A.

```
//Exerc 2

#include <stdio.h>

int main()
{
    int a[5][5];
    int b[5][5];

    printf("Digite uma matriz 5x5: \n");
    for(int i=0;i<5;i++){
        for(int j=0;j<5;j++){
            printf("elemento[%d][%d]:", i,j);
            scanf("%d", &a[i][j]);
        }
    }

    for(int i=0;i<5;i++){
        for(int j=0;j<5;j++){
            b[i][j] = a[i][j]*3;
        }
    }

    for(int i=0;i<5;i++){
        for(int j=0;j<5;j++){
            printf("%d \n", b[i][j]);
        }
    }
}
```

3. Faça um algoritmo que leia uma matriz de 10 x 8 e apresente na tela o maior e o menor elemento da matriz.

```
//Exerc 3

#include <stdio.h>

int main()
{
    int a[10][8];
    int maior = 0;
    int menor = 0;

    printf("Digite uma matriz 10x8: \n");
    for(int i=0;i<10;i++){
        for(int j=0;j<8;j++){
            printf("elemento[%d][%d]:", i,j);
            scanf("%d", &a[i][j]);
            if(i==0 && j==0){
                maior = a[i][j];
                menor = a[i][j];
            }
            else{
                if(a[i][j] > maior){
                    maior = a[i][j];
                }
                if(a[i][j] < menor){
                    menor = a[i][j];
                }
            }
        }
    }

    printf("Maior número: %i \nMenor número: %i", maior, menor);
}
```

4. Crie um algoritmo que leia uma matriz de dimensão (10x6) e apresente na tela a quantidade de elementos pares desta matriz.

```
//Exerc 4

#include <stdio.h>

int main()
{
    int a[10][6];
    int p = 0;

    printf("Digite uma matriz 10x8: \n");
    for(int i=0;i<10;i++){
        for(int j=0;j<8;j++){
            printf("elemento[%d][%d]:", i,j);
            scanf("%d", &a[i][j]);
            if(a[i][j] % 2 == 0){
                p++;
            }
        }
    }

    printf("Quantidade de números pares: %i", p);
}
```

5. Faça um algoritmo que leia uma matriz A (5 x 5). Construa uma matriz B de mesma dimensão onde cada elemento de B deverá ser o dobro de cada elemento correspondente de A, exceto para os valores da diagonal principal (números 1 na matriz **de exemplo** abaixo), cujos valores deverão ser o triplo de cada elemento correspondente da matriz A.

	0	1	2
0	1	0	0
1	0	1	0
2	0	0	1

```
//Exerc 5

#include <stdio.h>

int main()
{
    int a[5][5];
    int b[5][5];

    printf("Digite uma matriz 5x5: \n");
    for(int i=0;i<5;i++){
        for(int j=0;j<5;j++){
            printf("elemento[%d][%d]:", i,j);
            scanf("%d", &a[i][j]);
        }
    }

    for(int i=0;i<5;i++){
        for(int j=0;j<5;j++){
            if(i == j){
                b[i][j] = a[i][j]*3;
            }
            else{
                b[i][j] = a[i][j]*2;
            }
        }
    }

    for(int i=0;i<5;i++){
        for(int j=0;j<5;j++){
            printf("%d \n", b[i][j]);
        }
    }
}
```

armazene a soma dos elementos de A na primeira coluna e a multiplicação dos elementos de A na segunda coluna.

```
//Exerc 6

#include <stdio.h>

int main()
{
    int a[5][2];
    int b[5][2];

    printf("Digite uma matriz 5x2: \n");
    for(int i=0;i<5;i++){
        for(int j=0;j<2;j++){
            printf("elemento[%d][%d]:", i,j);
            scanf("%d", &a[i][j]);
        }
    }

    for(int i=0;i<5;i++){
        for(int j=0;j<2;j++){
            if(j==0){
                b[i][j] = a[i][j] + a[i][j];
            }
            else if(j==1){
                b[i][j] = a[i][j] * a[i][j];
            }
        }
    }

    for(int i=0;i<5;i++){
        for(int j=0;j<2;j++){
            printf("%d \n", b[i][j]);
        }
    }
}
```

7. Construa um algoritmo que leia do teclado uma matriz quadrada (12 x 12) e verifique se ela é uma matriz identidade. Uma matriz quadrada possui o mesmo número de linhas e colunas. Uma matriz identidade possui 1 nos elementos da diagonal principal e 0 nos demais. Exemplo: matriz identidade de dimensão 3x3.

	0	1	2
0	1	0	0
1	0	1	0
2	0	0	1

//Exerc 7

```
#include <stdio.h>
#include <stdbool.h>

int main()
{
    int a[12][12];
    bool ident = true;

    printf("Digite uma matriz 12x12: \n");
    for(int i=0;i<12;i++){
        for(int j=0;j<12;j++){
            printf("elemento[%d][%d]:", i,j);
            scanf("%d", &a[i][j]);
        }
    }

    for(int i=0;i<12;i++){
        for(int j=0;j<12;j++){
            if(i == j){
                if(a[i][j] != 1){
                    ident = false;
                }
            }
            else{
                if(a[i][j] == 0){
                    ident = false;
                }
            }
        }
    }

    if(ident){
        printf("É uma matriz identidade.");
    }
    else{
        printf("Não é uma matriz identidade.");
    }
}
```

8. Construa um algoritmo que leia uma matriz de dimensão 7 x 4 e, em um vetor de 7 elementos, armazene o menor elemento de cada linha da matriz. Exemplo:

	0	1	2	3
0	5	4	2	8
1	3	6	9	-1
2	0	-3	4	7
...
6	4	9	7	6

0	1	2	3	4	5	6
2	-	-3	4
	1		.		.	

```
//Exerc 8

#include <stdio.h>

int main()
{
    int a[7][4];
    int b[7];
    int menor;

    printf("Digite uma matriz 7x4: \n");
    for(int i=0;i<7;i++){
        for(int j=0;j<4;j++){
            printf("elemento[%d][%d]:", i,j);
            scanf("%d", &a[i][j]);

            if(j==0){
                menor = a[i][j];
            }
            else{
                if(a[i][j]<menor){
                    menor = a[i][j];
                }
            }
            b[i] = menor;
        }
    }

    printf("Menores valores: \n");
    for(int i=0;i<7;i++){
        printf("%i\n", b[i]);
    }
}
```


9. Elabore um algoritmo que leia dois vetores A e B de 15 números do teclado. Após a leitura, crie uma matriz de 15 linhas e 2 colunas onde a primeira coluna armazena os elementos de A e a segunda coluna armazena os elementos de B. Apresente a matriz na tela. Exemplo:

0	1	2	3	4	...	14
10	20	30	40	50	...	150

0	1	2	3	4	...	14
5	15	2	35	45	...	145

	0	1
0	10	5
1	20	15
2	30	25
...
14	150	145

```
//Exerc 9

#include <stdio.h>

int main()
{
    int a[15];
    int b[15];
    int m[15][2];

    printf("Digite 15 números: \n");
    for(int i=0;i<15;i++){
        scanf("%i", &a[i]);
    }
    printf("Digite mais 15 números: \n");
    for(int i=0;i<15;i++){
        scanf("%i", &b[i]);
    }

    for(int i=0;i<15;i++){
        for(int j=0;j<2;j++){
            if(j == 0){
                m[i][j] = a[i];
            }
            else{
                m[i][j] = b[i];
            }
            printf("elemento[%d][%d]:%d \n", i,j,m[i][j]);
        }
    }
}
```

10. Elabore um algoritmo que leia do teclado uma matriz triangular superior de dimensão 20x20. Uma matriz triangular superior é quadrada e possui 0 nos elementos abaixo da diagonal principal e qualquer outro valor nos elementos da diagonal principal e acima dela. Exemplo: matriz triangular superior de tamanho 4.

	0	1	2	3
0	x	x	x	x
1	0	x	x	x
2	0	0	x	x
3	0	0	0	x

```
//Exerc 10

#include <stdio.h>

int main()
{
    int a[20][20];
    int b[20][20];

    printf("Digite uma matriz 20x20: \n");
    for(int i=0;i<20;i++){
        for(int j=0;j<20;j++){
            printf("elemento[%d][%d]:", i,j);
            scanf("%d", &a[i][j]);
        }
    }

    for(int i=0;i<20;i++){
        for(int j=0;j<20;j++){
            if(i<=j){
                b[i][j] = a[i][j];
            }
            else{
                b[i][j] = 0;
            }
        }
    }

    for(int i=0;i<20;i++){
        for(int j=0;j<20;j++){
            printf("%d \n", b[i][j]);
        }
    }
}
```

- 11.** Elabore um algoritmo que leia uma matriz A dimensão 3x4 e crie uma matriz B que representa a transposta de A. Apresente na tela a matriz B. Exemplo:

	0	1	2	3		0	1	2
0	1	2	3	4	⇒	0	1	2
1	5	6	7	8		1	2	10
2	9	10	11	12		2	3	11
						3	4	12

```

//Exerc 11

#include <stdio.h>

int main()
{
    int a[3][4];
    int b[4][3];

    printf("Digite uma matriz 3x4: \n");
    for(int i=0;i<3;i++){
        for(int j=0;j<4;j++){
            printf("elemento[%d][%d]:", i,j);
            scanf("%d", &a[i][j]);
        }
    }

    for(int i=0;i<4;i++){
        for(int j=0;j<3;j++){
            b[i][j] = a[j][i];
        }
    }

    for(int i=0;i<4;i++){
        for(int j=0;j<3;j++){
            printf("%d \n", b[i][j]);
        }
    }
}

```

- 12.** Construa um algoritmo que leia duas matrizes do teclado de dimensão 5 x 3 e realize a troca dos elementos destas matrizes.

```
//Exerc 12
1.
2. #include <stdio.h>
3.
4. void set(int m1[5][3], int m2[5][3]);
5.
6. int a[5][3];
7. int b[5][3];
8. int c[5][3];
9.
10. int main()
11. {
12.
13.     printf("Digite uma matriz 5x3: \n");
14.     for(int i=0;i<5;i++){
15.         for(int j=0;j<3;j++){
16.             printf("elemento[%d][%d]:", i,j);
17.             scanf("%d", &a[i][j]);
18.         }
19.     }
20.
21.     printf("Digite outra matriz 5x3: \n");
22.     for(int i=0;i<5;i++){
23.         for(int j=0;j<3;j++){
24.             printf("elemento[%d][%d]:", i,j);
25.             scanf("%d", &b[i][j]);
26.         }
27.     }
28.
29.     set(c,b);
30.     set(b,a);
31.     set(a,c);
32.
33.     for(int i=0;i<4;i++){
34.         for(int j=0;j<3;j++){
35.             printf("%d \n", a[i][j]);
36.         }
37.     }
38.     for(int i=0;i<4;i++){
39.         for(int j=0;j<3;j++){
40.             printf("%d \n", b[i][j]);
41.         }
42.     }
43. }
44.
45. void set(int m1[5][3], int m2[5][3]){
46.
47.     for(int i=0;i<5;i++){
48.         for(int j=0;j<3;j++){
49.             m1[i][j] = m2[i][j];
50.         }
51.     }
52. }
```

13. O tempo que um determinado avião leva para percorrer o trecho entre duas localidades distintas está disponível através da seguinte tabela:

cidades	0	1	2	3	4	5	6
0	-	02	11	06	15	11	01
1	02	-	07	12	04	02	15
2	11	07	-	11	08	03	13
3	06	12	11	-	10	02	01
4	15	04	08	10	-	05	13
5	11	02	03	02	05	-	14
6	01	15	13	01	13	14	-

- a) Construa um algoritmo que leia a tabela acima e informe ao usuário o tempo necessário para percorrer duas cidades por ele fornecidas, até o momento em que ele fornecer duas cidades iguais (fonte e destino). Não deve ser lido informações para a diagonal principal.

```
//Exerc 13a

#include <stdio.h>
#include <stdbool.h>

int main()
{
    int m[7][7] = {{0,2,11,6,15,11,1},{2,0,7,12,4,2,15},{11,7,0,11,8,3,13},
    {6,12,11,0,10,2,1},{15,4,8,10,0,5,13},{11,2,3,2,5,0,14},
    {1,15,13,1,13,14,0}};

    bool p = true;
    int r[2];

    while(p){

        printf("Digite o número de duas cidades:");
        scanf("%d %d", &r[0], &r[1]);

        if(r[0] == r[1]){
            return 0;
        }

        printf("Tempo necessário de viagem: %dh \n", m[r[0]][r[1]]);

    }
}
```

- b) Elabore um algoritmo que imprima a tabela sem repetições (apenas o triângulo superior ou o triângulo inferior).

```
//Exerc 13b

#include <stdio.h>

int main()
{
    int m[7][7] = {{0,2,11,6,15,11,1},{2,0,7,12,4,2,15},{11,7,0,11,8,3,13},
    {6,12,11,0,10,2,1},{15,4,8,10,0,5,13},{11,2,3,2,5,0,14},
    {1,15,13,1,13,14,0}};

    for(int i=0;i<7;i++){
        for(int j=0;j<7;j++){
            if(i<j){
                printf("%d \n", m[i][j]);
            }
        }
    }
}
```

c) Desenvolva um algoritmo que permita ao usuário informar várias cidades, até inserir a cidade “7”, e que imprima o tempo total para cumprir todo o percurso especificado entre as cidades fornecidas. Ou seja:

- O usuário deve informar as cidades que ele quer cadastrar a distância.
- Notem que é um percurso, então, por exemplo, o segundo trajeto inicia da cidade destino do primeiro trajeto.

Exemplo do “c”:

Digite cidade de origem: 1
 Digite cidade de destino: 3
 Digite o tempo entre a cidade 1 e 3: 10
 Digite cidade de destino: 5
 Digite o tempo entre a cidade 3 e 5: 21
 Digite cidade de destino: 2
 Digite o tempo entre a cidade 5 e 2: 39
 Digite cidade de destino: 7

O tempo total do seu trajeto é: 70 minutos

		Destino						
Origem	cidades	0	1	2	3	4	5	6
	0	-						
	1		-		10			
	2			-				
	3				-		21	
	4					-		
	5			39			-	
	6							-

```
//Exerc 13c
53.
54.  #include <stdio.h>
55.
56.  int soma = 0;
57.  int calc();
58.
59.  int main()
60.  {
61.      calc();
62.  }
63.
64.  int calc(){
65.
66.      int m[7][7];
67.
68.      int r[3];
69.
70.      printf("Digite a cidade de origem: ");
71.      scanf("%d", &r[0]);
72.
73.      if(r[0] == 7){
74.          printf("O tempo total do seu trajeto é: %i", soma);
75.          return 0;
76.      }
77.
78.      printf("Digite a cidade de destino: ");
79.      scanf("%d", &r[1]);
80.
81.      if(r[1] == 7){
82.          printf("O tempo total do seu trajeto é: %i", soma);
83.          return 0;
84.      }
85.
86.      printf("Digite o tempo entre a cidade %d e %d: ", r[0], r[1]);
87.      scanf("%d", &r[2]);
88.
89.      m[r[0]][r[1]] = r[2];
90.      soma += r[2];
91.
92.      main();
93. }
```