

Lista de Exercícios – Matriz (vetor multidimensional)

Monitor: Rafael Sian de Freitas

Exercícios de Fixação

- 1** – Faça um programa que soma todos os elementos de uma matriz 4x4.
- 2** – Faça um programa que soma todos os elementos da primeira linha de uma matriz 5x5.
- 3** – Faça um programa que imprima os elementos da diagonal principal de uma matriz 5x5.
- 4** – Faça um programa que multiplique todos os elementos de uma matriz 3x3 por um número **N**, exceto os elementos da diagonal principal.
- 5** – Faça um programa que receba valores inteiros para uma matriz 5x5 e imprima a sua matriz transposta.
- 6** – Faça um programa que receba do usuário um tamanho de uma matriz de no máximo 5x5. Utilizando o tamanho informado, declare duas matrizes e realize as seguintes operações e mostre em uma nova matriz 3 o resultado:
 - a) Soma;
 - b) Subtração;
 - c) Multiplicação.
- 7** – Um sistema linear de equações pode ser representado através de matrizes:

$$\begin{cases} 4x + y = 20 \\ x - 10y = 69 \end{cases} \quad \begin{pmatrix} 4 & 1 \\ 1 & -10 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 20 \\ 69 \end{pmatrix}$$

Faça um programa que leia as 3 matrizes de um sistema linear e imprima as duas equações assim como no exemplo acima. Obs.: Para as duas matrizes que só possuem 1 coluna pode ser utilizado vetores unidimensionais.

8 – Supondo que existem várias pedras do jogo de xadrez em um tabuleiro. Para facilitar a indicação das peças, vamos supor que:

1 – peões 2 – torres 3 – cavalos 4 – bispos 5 – reis 6 – rainhas 7 – vazio

O tabuleiro é o seguinte:

2	4	3	7	5	3	4	2
7	1	1	1	7	1	7	1
7	7	7	7	7	7	7	7
1	7	7	7	7	7	7	7
1	7	7	1	7	7	7	1
2	3	7	7	7	7	3	7
7	1	1	7	1	7	1	7
7	4	7	5	6	7	4	2

Faça um programa que determine a soma de todos os espaços vazios e a quantidade de peões.

9 – Utilizando o enunciado do exercício anterior, faça um programa que mostre a quantidade de cada tipo de peça existente no tabuleiro.

10 – A distância entre várias cidades é dada pela tabela abaixo (em km):

	1	2	3	4	5
1	00	20	62	21	02
2	20	00	12	23	45
3	62	12	00	03	10
4	21	23	03	00	06
5	02	45	10	06	00

Faça um programa que leia a tabela acima e informe ao usuário a distância entre duas cidades por ele requisitadas, até que ele entre com o código 0 para ambas as cidades.

11 – Faça um programa que receba uma matriz $n \times m$ e crie um vetor unidimensional $n*m$ que receba todos os valores da matriz.

12 – Dizemos que uma matriz inteira é uma matriz de permutação se em cada linha e em cada coluna houver $n-1$ elementos nulos e um único elemento igual a 1.

Exemplo:

A matriz $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ é uma matriz de permutação.

Já a matriz $\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ não é uma matriz de permutação.

Faça um programa que verifique se uma matriz de ordem **N** é uma matriz de permutação.

13 – Dada uma matriz de tamanho **NxM** verifique quantas linhas e colunas são nulas.

Exemplo:

$N = 3$ e $M = 4$

A matriz $\begin{pmatrix} 1 & 0 & 9 \\ 4 & 0 & 6 \\ 0 & 0 & 0 \\ 0 & 0 & 5 \end{pmatrix}$ possui 1 coluna e 1 linha nula.

Faça um programa que receba de entrada o número de linhas e colunas de uma matriz inteira e verifique quantas linhas e colunas essa matriz possui que são nulas.

14 – Implemente um sistema de correção de provas que funciona da seguinte maneira: existe um vetor de tamanho 5 que armazena o gabarito da prova e uma matriz de tamanho 5×30 que armazena as respostas dos alunos.

15 – Modifique o exercício anterior para corrigir as provas e mostrar quantas alternativas cada aluno acertou. Dica: Cada aluno é identificado pelo seu índice na matriz.

Exercícios Resolvidos

1 – Faça um programa que receba números aleatórios para uma matriz 6x6.

```
#include <stdio.h>
#include <time.h>
#define T 6

int main()
{
    srand(time(NULL));
    int m[T][T], i, j;

    for (i=0; i<T; i++)
    {
        for (j=0; j<T; j++)
        {
            m[i][j] = rand();
        }
    }

    for (i=0; i<T; i++)
    {
        for (j=0; j<T; j++)
        {
            printf("\t\t%d ", m[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

2 – Faça um programa que imprima a soma de todos os elementos da diagonal secundária de uma matriz NxN.

```
#include <stdio.h>

int main()
{
    int i, j, n, somaDiagSec = 0;

    printf ("Digite um numero N: ");
    scanf ("%d", &n);

    int m[n][n];

    for (i=0; i<n; i++)
    {
        for (j=0; j<n; j++)
        {
            printf ("Digite o elemento %d %d da matriz: ", i, j);
            scanf ("%d", &m[i][j]);
        }
    }

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

```

j = n - 1;
for (i=0; i<n; i++)
{
    somaDiagSec += m[i][j];
    j--;
}

printf("\nSoma da diagonal secundaria: %d", somaDiagSec);

return 0;
}

```

3 – Faça um programa que mostre a soma dos elementos de todas as linhas e colunas, a soma dos elementos da diagonal principal e a soma dos elementos da diagonal secundária de uma matriz 3x3.

```

#include <stdio.h>
#include <time.h>
#define T 3

int main()
{
    srand(time(NULL));
    int m[T][T], i, j, soma = 0, somaDiagPrinc = 0, somaDiagSec = 0;

    for (i=0; i<T; i++)
    {
        for (j=0; j<T; j++)
        {
            m[i][j] = rand()%10;
        }
    }

    for (i=0; i<T; i++)
    {
        for (j=0; j<T; j++)
        {
            printf("%d \t\t", m[i][j]);
        }
        printf("\n");
    }

    for (i=0; i<T; i++)
    {
        for (j=0; j<T; j++)
        {
            soma += m[i][j];
        }
        printf("\n");
    }

    for (i=0; i<T; i++)
    {
        somaDiagPrinc += m[i][i];
    }

    j = T - 1;
    for (i=0; i<T; i++)
    {
        somaDiagSec += m[i][j];
        j--;
    }

    printf("Soma dos elementos: %d\nSoma da diagonal principal: %d\nSoma da diagonal secundaria: %d", soma, somaDiagPrinc, somaDiagSec);

    return 0;
}

```

Desafios

1 – Dizemos que uma matriz é um quadrado mágico se a soma da diagonal principal, a soma da diagonal secundária, a soma de cada linha e a soma de cada coluna são todos iguais.

Exemplo:

A matriz $\begin{pmatrix} 8 & 0 & 7 \\ 4 & 5 & 6 \\ 3 & 10 & 2 \end{pmatrix}$ é um quadrado mágico.

Faça um programa que verifique se uma matriz de ordem **N** é um quadrado mágico.

2 – Dada uma matriz real NxM, verifique se existem número repetidos na matriz.