

Tarefa Matrizes

☰ Week	Semana 6
☼ Status	Concluído

Mariana Alves Brunetti da Silva

Engenharia de Software 3º Período

1-

```
using System;

namespace MatrizNegativos
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int m, n;

            // Entrada dos valores de M e N
            Console.WriteLine("Digite o número de linhas (M): ");
            m = int.Parse(Console.ReadLine());

            Console.WriteLine("Digite o número de colunas (N): ");
            n = int.Parse(Console.ReadLine());

            // Criando a matriz de MxN
            int[,] mat = new int[m, n];

            // Preenchendo a matriz
            Console.WriteLine("Digite os elementos da matriz:");
            for (int i = 0; i < m; i++)
            {
                for (int j = 0; j < n; j++)
```

```

        {
            Console.WriteLine($"Elemento [{i+1}, {j+1}]: ");
            mat[i, j] = int.Parse(Console.ReadLine());
        }
    }

    // Mostrando os números negativos da matriz
    Console.WriteLine("\nNúmeros negativos na matriz:");
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (mat[i, j] < 0)
            {
                Console.Write(mat[i, j] + " ");
            }
        }
    }

    Console.ReadKey();
}
}
}

```

Saída:

```

Números negativos na matriz:
-3 -5 -1 -2 -4 -6

```

2-

```

using System;

namespace SomaLinhasMatriz
{
    internal class Program
    {
        static void Main(string[] args)

```

```

{
    int n;

    // Lendo o valor de N
    Console.Write("Digite o valor de N (tamanho da matriz NxN): ");
    n = int.Parse(Console.ReadLine());

    // Criando a matriz NxN
    int[,] mat = new int[n, n];

    // Preenchendo a matriz com os valores fornecidos pelo usuário
    Console.WriteLine("Digite os elementos da matriz:");
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            Console.Write($"Elemento [{i + 1}, {j + 1}]: ");
            mat[i, j] = int.Parse(Console.ReadLine());
        }
    }

    // Calculando a soma dos elementos de cada linha e mostrando na tela
    Console.WriteLine("\nSoma dos elementos de cada linha:");
    for (int i = 0; i < n; i++)
    {
        int somaLinha = 0;
        for (int j = 0; j < n; j++)
        {
            somaLinha += mat[i, j]; // Somando os elementos da linha
        }
        Console.Write(somaLinha + " "); // Exibindo a soma da linha
    }

    Console.ReadKey();
}
}
}

```

Saída:

```
Soma dos elementos de cada linha:  
11 19 29
```

3-

```
using System;  
  
namespace MaiorElementoPorLinha  
{  
    internal class Program  
    {  
        static void Main(string[] args)  
        {  
            int n;  
  
            // Lendo o valor de N (tamanho da matriz NxN)  
            Console.WriteLine("Digite o valor de N (ordem da matriz NxN): ");  
            n = int.Parse(Console.ReadLine());  
  
            // Criando a matriz quadrada de ordem N  
            int[,] mat = new int[n, n];  
  
            // Preenchendo a matriz com os valores fornecidos pelo usuário  
            Console.WriteLine("Digite os elementos da matriz:");  
            for (int i = 0; i < n; i++)  
            {  
                for (int j = 0; j < n; j++)  
                {  
                    Console.WriteLine($"Elemento [{i + 1}, {j + 1}]: ");  
                    mat[i, j] = int.Parse(Console.ReadLine());  
                }  
            }  
  
            // Encontrando e exibindo o maior elemento de cada linha  
            Console.WriteLine("\nMaior elemento de cada linha:");  
            for (int i = 0; i < n; i++)
```

```

    {
        int maiorElemento = mat[i, 0]; // Inicializa o maior elemento com o p
        for (int j = 1; j < n; j++) // Percorre os demais elementos da linha
        {
            if (mat[i, j] > maiorElemento)
            {
                maiorElemento = mat[i, j]; // Atualiza o maior elemento se enco
            }
        }
        Console.Write(maiorElemento + " "); // Exibe o maior elemento da li
    }

    Console.ReadKey();
}
}
}

```

Saída:

Maior elemento de cada linha:
 12 7 8 15

4-

```

using System;

namespace SomaAcimaDiagonal
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int n;

            // Lendo o valor de N (ordem da matriz NxN)
            Console.Write("Digite o valor de N (ordem da matriz NxN): ");
            n = int.Parse(Console.ReadLine());
        }
    }
}

```

```

// Criando a matriz quadrada de ordem N
int[,] mat = new int[n, n];

// Preenchendo a matriz com os valores fornecidos pelo usuário
Console.WriteLine("Digite os elementos da matriz:");
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
    {
        Console.Write($"Elemento [{i + 1}, {j + 1}]: ");
        mat[i, j] = int.Parse(Console.ReadLine());
    }
}

// Calculando a soma dos elementos acima da diagonal principal
int soma = 0;
for (int i = 0; i < n; i++)
{
    for (int j = i + 1; j < n; j++) // j > i, ou seja, acima da diagonal principal
    {
        soma += mat[i, j];
    }
}

// Exibindo o resultado
Console.WriteLine("\nSoma dos elementos acima da diagonal principal:");
Console.WriteLine(soma);

Console.ReadKey();
}
}
}

```

Saída:

```

Soma dos elementos acima da diagonal principal:
12

```

5-

```
using System;

namespace SomaMatrizes
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int m, n;

            // Lendo o valor de M (linhas) e N (colunas)
            Console.Write("Digite o número de linhas (M): ");
            m = int.Parse(Console.ReadLine());

            Console.Write("Digite o número de colunas (N): ");
            n = int.Parse(Console.ReadLine());

            // Criando as matrizes A, B e C de tamanho MxN
            int[,] A = new int[m, n];
            int[,] B = new int[m, n];
            int[,] C = new int[m, n];

            // Preenchendo a matriz A
            Console.WriteLine("Digite os elementos da matriz A:");
            for (int i = 0; i < m; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    Console.Write($"Elemento A[{i + 1}, {j + 1}]: ");
                    A[i, j] = int.Parse(Console.ReadLine());
                }
            }

            // Preenchendo a matriz B
            Console.WriteLine("Digite os elementos da matriz B:");
            for (int i = 0; i
```

Saída:

Matriz C (soma de A e B):

5 9 7

5 13 9

6-

```
using System;

namespace MatrizOperacoes
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int n;

            // Lendo o valor de N (ordem da matriz)
            Console.Write("Digite o valor de N (ordem da matriz): ");
            n = int.Parse(Console.ReadLine());

            // Criando a matriz quadrada de ordem N
            double[,] mat = new double[n, n];

            // Preenchendo a matriz com os valores fornecidos pelo usuário
            Console.WriteLine("Digite os elementos da matriz:");
            for (int i = 0; i < n; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    mat[i, j] = double.Parse(Console.ReadLine());
                }
            }

            // a) Calcular e imprimir a soma de todos os elementos positivos da matriz
            double somaPositivos = 0;
            for (int i = 0; i < n; i++)
```



```

{
    for (int j = 0; j < n; j++)
    {
        if (mat[i, j] > 0)
        {
            somaPositivos += mat[i, j];
        }
    }
}
Console.WriteLine($"SOMA DOS POSITIVOS: {somaPositivos:F1}");

// b) Ler o índice de uma linha e imprimir todos os elementos dessa linha
Console.Write("Digite o índice da linha (1 a N): ");
int linha = int.Parse(Console.ReadLine()) - 1; // Subtrai 1 para usar o índice

Console.Write("LINHA ESCOLHIDA: ");
for (int j = 0; j < n; j++)
{
    Console.Write($"{mat[linha, j]:F1} ");
}
Console.WriteLine();

// c) Ler o índice de uma coluna e imprimir todos os elementos dessa coluna
Console.Write("Digite o índice da coluna (1 a N): ");
int coluna = int.Parse(Console.ReadLine()) - 1; // Subtrai 1 para usar o índice

Console.Write("COLUNA ESCOLHIDA: ");
for (int i = 0; i < n; i++)
{
    Console.Write($"{mat[i, coluna]:F1} ");
}
Console.WriteLine();

// d) Imprimir os elementos da diagonal principal da matriz
Console.Write("DIAGONAL PRINCIPAL: ");
for (int i = 0; i < n; i++)
{
    Console.Write($"{mat[i, i]:F1} ");
}

```

```

    }
    Console.WriteLine();

    // e) Alterar a matriz elevando ao quadrado todos os números negativo
    // e imprimir a matriz alterada
    Console.WriteLine("MATRIZ ALTERADA:");
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (mat[i, j] < 0)
            {
                mat[i, j] = mat[i, j] * mat[i, j]; // Eleva ao quadrado os negativos
            }
            Console.Write($"{mat[i, j]:F1} ");
        }
        Console.WriteLine();
    }

    Console.ReadKey();
}
}
}

```

Saída:

```

SOMA DOS POSITIVOS: 40.0
LINHA ESCOLHIDA: -2.0 3.0 5.0
COLUNA ESCOLHIDA: 10.0 5.0 4.0
DIAGONAL PRINCIPAL: 7.0 3.0 4.0
MATRIZ ALTERADA:
7.0 64.0 10.0
4.0 3.0 5.0
11.0

```

7-

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Matriz7
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int m, n, fila;
            int [,] mat;
            string[] valores;

            Console.Write("Digite a quantidade de filas: ");
            m = int.Parse(Console.ReadLine());

            Console.WriteLine("Digite a quantidade de soldados: ");
            n = int.Parse(Console.ReadLine());

            mat = new int[m, n];

            for (int i = 0; i < m; i++)
            {
                Console.Write($"Fila " {i+1} + : ");
                valores = Console.ReadLine().Split(" ");
                for (int j = 0; j < n; j++)
                {
                    mat[i, j] = int.Parse(valores[j]);
                }
            }

            Console.Write("Fila a ser movimentada: ");
            fila = int.Parse(Console.ReadLine());
        }
    }
}

```

```

// decrementar o valor da fila, visto que a posição inicial é 0
fila--;

// salvando o último soldado da fila que vai para a primeira posição poste
// n - 1: porque não tem a 5ª posição, pois vai de 0 a 4
ultimoDaFila = MatchCasing[fila, n - 1];

// missão:
for (int i = n - 1; i > 0; i--)
{
    // mat[1, 4]
    // mat[1, 3] = 9271
    mat[fila, i] = mat[fila, i - 1];
}

mat[fila, 0] = ultimoDaFila;
}
}

```

Saída:

```

Digite a quantidade de filas: 3
Digite a quantidade de soldados por fila: 4
Fila 1: 1 2 3 4
Fila 2: 5 6 7 8
Fila 3: 9 10 11 12
Fila a ser movimentada: 2

```