

Nome: Maria Elisa Resende Vieira

Professor : Humberto

Main.cs

```
using System;
```

```
class Program {  
    public static void Main () {  
        Exerc1.Rodar();  
        Exerc2.Rodar();  
        Exerc3.Rodar();  
        Exerc4.Rodar();  
        Exerc5.Rodar();  
        Exerc6.Rodar();  
        Exerc7.Rodar();  
        Exerc8.Rodar();  
        Exerc9.Rodar();  
        Exerc10.Rodar();  
    }  
}
```

Exerc1.cs

using System;

class Exerc1

```
{  
    public static void Rodar ()  
    {  
        int[] N = new int[20];  
  
        // faz a leitura do vetor N[20]  
        Console.WriteLine("Digite 20 números inteiros:");  
        for (int i = 0; i < 20; i++)  
        {  
            Console.Write("N[{0}] = ", i);  
            N[i] = int.Parse(Console.ReadLine());  
        }  
  
        // Vai iniciar o menor elemento com o primeiro elemento do vetor  
        int menor = N[0];  
        int posicao = 0;  
  
        // vai encontrar o menor elemento e sua posição  
        for (int i = 1; i < N.Length; i++)  
        {  
            if (N[i] < menor)  
            {  
                menor = N[i];  
                posicao = i;  
            }  
        }  
  
        // Vai exibir o menor elemento e sua posição  
        Console.WriteLine($"O menor elemento de N é {menor} e sua posição dentro do vetor  
        é: {posicao}");  
    }  
}
```

Exerc2.cs

using System;

class Exerc2

```
{
    public static void Rodar ()
    {
        // Declarei a variavel numAlunos e o numero de alunos
        const int numAlunos = 10;
        double[] notas = new double[numAlunos];

        PreencherNotas(notas);
        CalcularMediaEContarAcimaDaMedia(notas);
    }

    // criei um static void pra preencher as notas dos alunos
    static void PreencherNotas(double[] notas)
    {
        Console.WriteLine("Digite as notas de 10 alunos:");
        for (int i = 0; i < notas.Length; i++)
        {
            Console.Write($"Nota do aluno {i + 1}: ");
            notas[i] = double.Parse(Console.ReadLine());
        }
    }

    // criei um static void pra calcular a media e contar quantos alunos estão acima da media
    static void CalcularMediaEContarAcimaDaMedia(double[] notas)
    {
        double soma = 0;
        for (int i = 0; i < notas.Length; i++)
        {
            soma += notas[i];
        }

        double media = soma / notas.Length;

        int contadorAcimaDaMedia = 0;
        for (int i = 0; i < notas.Length; i++)
        {
            if (notas[i] > media)
            {
                contadorAcimaDaMedia++;
            }
        }

        // fala a media da turma e quantos alunos estão acima da media
    }
}
```

```
        Console.WriteLine($"A média da turma é: {media:F2}");
        Console.WriteLine($"{contadorAcimaDaMedia} aluno(s) obteve(ram) nota acima da
média.");
    }
}
```

Exerc3.cs

using System;

class Exerc3

```
{
    public static void Rodar ()
    {
        int[] vetorX = new int[10];
        PreencherVetor(vetorX);
        int[] negativos = CopiarNegativos(vetorX);
        ExibirVetor(vetorX, "Vetor original:");
        ExibirVetor(negativos, "Vetor de negativos:");
    }

    // Vai preencher o vetor
    static void PreencherVetor(int[] vetor)
    {
        Random random = new Random();
        for (int i = 0; i < vetor.Length; i++)
        {
            vetor[i] = random.Next(-50, 51);
            // Vai gerar números aleatórios entre -50 e 50
        }
    }

    // É uma função que vai copiar valores negativos para um novo vetor
    static int[] CopiarNegativos(int[] vetor)
    {
        int[] tempNegativos = new int[vetor.Length];
        int count = 0;

        for (int i = 0; i < vetor.Length; i++)
        {
            if (vetor[i] < 0)
            {
                tempNegativos[count] = vetor[i];
                count++;
            }
        }

        int[] negativos = new int[count];
        Array.Copy(tempNegativos, negativos, count);
        return negativos;
    }

    // é um procedimento para exibir o conteúdo de um vetor
    static void ExibirVetor(int[] vetor, string mensagem)
```

```
{  
    Console.WriteLine(mensagem);  
    for (int i = 0; i < vetor.Length; i++)  
    {  
        Console.Write(vetor[i] + " ");  
    }  
    Console.WriteLine();  
}  
}
```

Exerc4.cs

using System;

class Exerc4

```
{
    public static void Rodar ()
    {
        int[] vetorX = new int[10];
        int[] vetorY = new int[10];

        PreencherVetores(vetorX, vetorY);
        int[] vetorIntercalado = GerarVetorIntercalado(vetorX, vetorY);

        ExibirVetor(vetorX, "Vetor X:");
        ExibirVetor(vetorY, "Vetor Y:");
        ExibirVetor(vetorIntercalado, "Vetor Intercalado:");
    }

    // Vai preencher dois vetores
    static void PreencherVetores(int[] vetorX, int[] vetorY)
    {
        Random random = new Random();
        for (int i = 0; i < 10; i++)
        {
            vetorX[i] = random.Next(1, 101);
            // Vai gerar números aleatórios entre 1 e 100 para vetorX
            vetorY[i] = random.Next(1, 101);
            // Vai gerar números aleatórios entre 1 e 100 para vetorY
        }
    }

    // Vai gerar um novo vetor intercalado
    static int[] GerarVetorIntercalado(int[] vetorX, int[] vetorY)
    {
        int[] vetorIntercalado = new int[20];
        for (int i = 0; i < 10; i++)
        {
            vetorIntercalado[i * 2] = vetorX[i];
            vetorIntercalado[i * 2 + 1] = vetorY[i];
        }
        return vetorIntercalado;
    }

    // Vai exibir o conteúdo de um vetor
    static void ExibirVetor(int[] vetor, string mensagem)
    {
        Console.WriteLine(mensagem);
    }
}
```

```
    for (int i = 0; i < vetor.Length; i++)  
    {  
        Console.Write(vetor[i] + " ");  
    }  
    Console.WriteLine();  
}  
}
```


Exerc5.cs

```
using System;

class Exerc5
{
    public static void Rodar ()
    {
        int[] numerosSorteados = SortearNumeros();

        TentarAcertarNumeros(numerosSorteados);

        Console.WriteLine("Números sorteados: " + string.Join(", ", numerosSorteados));
    }

    // Vai sortiar 3 números entre 10 e 50 e retorna em um vetor
    static int[] SortearNumeros()
    {
        Random random = new Random();
        int[] numeros = new int[3];

        for (int i = 0; i < numeros.Length; i++)
        {
            numeros[i] = random.Next(10, 51);
            // Vai gerar números aleatórios entre 10 e 50 (inclusive)
        }

        return numeros;
    }

    // Vai lê as tentativas do usuário até acertar um número sorteado
    static void TentarAcertarNumeros(int[] numerosSorteados)
    {
        bool acertou = false;

        while (!acertou)
        {
            Console.Write("Tente adivinhar um dos números sorteados: ");
            int tentativa;

            // Confere se o usuario está colocando um numero entre 10 e 50
            while (!int.TryParse(Console.ReadLine(), out tentativa) || tentativa < 10 || tentativa >
50)
            {
                Console.WriteLine("Entrada inválida. Por favor, insira um número entre 10 e 50:");
            }

            for (int i = 0; i < numerosSorteados.Length; i++)
```

```
{
    if (tentativa == numerosSorteados[i])
    {
        Console.WriteLine("Parabéns! Você acertou um dos números sorteados.");
        acertou = true;
        break;
    }
}
// Pede ao usuario que tente novamente
if (!acertou)
{
    Console.WriteLine("Tente novamente!");
}
}
}
```

Exerc6.cs

using System;

class Exerc6

{

public static void Rodar ()

{

//Armazenei as temperaturas de cada dia de outubro

double[] temperaturas = new double[31];

//Coloquei valores aleatorios entre 15 e 40°C , gerados pelo sistema.

Random random = new Random();

for (int i = 0; i < temperaturas.Length; i++)

{

temperaturas[i] = random.Next(15, 41); // O programa gera um número aleatório entre 15 e 40

}

// Criei variáveis para armazenar a menor e a maior temperatura

double menorTemperatura = temperaturas[0];

double maiorTemperatura = temperaturas[0];

double somaTemperaturas = 0;

// Vai calcula a menor, maior e a soma das temperaturas

foreach (double temperatura in temperaturas)

{

if (temperatura < menorTemperatura)

menorTemperatura = temperatura;

if (temperatura > maiorTemperatura)

maiorTemperatura = temperatura;

somaTemperaturas += temperatura;

}

// Vai calcular a temperatura média

double temperaturaMedia = somaTemperaturas / temperaturas.Length;

// Essa parte eu coloquei para que o programa mostrasse a temperatura média com apenas duas casas decimais

double temperaturaMediaArredondada = Math.Round(temperaturaMedia, 1);

// Vai contar o número de dias com temperatura inferior à média

int diasInferiorMedia = 0;

foreach (double temperatura in temperaturas)

{

if (temperatura < temperaturaMediaArredondada)

diasInferiorMedia += 1;

}

```
// Essa parte imprime os resultados
Console.WriteLine($"Menor temperatura: {menorTemperatura}°C");
Console.WriteLine($"Maior temperatura: {maiorTemperatura}°C");
Console.WriteLine($"Temperatura média: {temperaturaMediaArredondada}°C");
Console.WriteLine($"Número de dias com temperatura inferior à média:
{diasInferiorMedia}");
    }
}
```

Exerc7.cs

```
using System;
class Exerc7 {

    public static void Rodar ()
    {
        // Nessa parte declarei a matriz 5 x 5
        int[,] matriz = new int[5, 5];

        // Vai preencher a matriz
        PreencherMatriz(matriz);

        // Vai exibir a matriz já preenchida
        Console.WriteLine("Matriz 5x5 preenchida:");
        ExibirMatriz(matriz);

        // Vai calcular e exibir as somas feitas
        Console.WriteLine($"Soma da linha 4: {SomaLinha4(matriz)}");
        Console.WriteLine($"Soma da coluna 2: {SomaColuna2(matriz)}");
        Console.WriteLine($"Soma da diagonal principal: {SomaDiagonalPrincipal(matriz)}");
        Console.WriteLine($"Soma da diagonal secundária:
{SomaDiagonalSecundaria(matriz)}");
        Console.WriteLine($"Soma de todos os elementos:
{SomaTodosElementos(matriz)}");
    }

    static void PreencherMatriz(int[,] matriz)
    {
        Random random = new Random();
        for (int i = 0; i < 5; i++)
        {
            for (int j = 0; j < 5; j++)
            {
                matriz[i, j] = random.Next(1, 101);
            }
        }
    }

    // Esse static void vai exibir a matriz
    static void ExibirMatriz(int[,] matriz)
    {
        for (int i = 0; i < 5; i++)
        {
            for (int j = 0; j < 5; j++)
            {
                Console.Write(matriz[i, j] + "\t");
            }
        }
    }
}
```

```

        }
        Console.WriteLine();
    }
}
// Esse static void vai calcular a soma da linha 4
static int SomaLinha4(int[,] matriz)
{
    int soma = 0;
    for (int j = 0; j < 5; j++)
    {
        soma += matriz[3, j];
    }
    return soma;
}

// Esse static void vai calcular a soma da coluna 2
static int SomaColuna2(int[,] matriz)
{
    int soma = 0;
    for (int i = 0; i < 5; i++)
    {
        soma += matriz[i, 1];
    }
    return soma;
}

// Esse static void vai calcular a soma da diagonal principal
static int SomaDiagonalPrincipal(int[,] matriz)
{
    int soma = 0;
    for (int i = 0; i < 5; i++)
    {
        soma += matriz[i, i];
    }
    return soma;
}

// Esse static void vai calcular a soma da diagonal secundária
static int SomaDiagonalSecundaria(int[,] matriz)
{
    int soma = 0;
    for (int i = 0; i < 5; i++)
    {
        soma += matriz[i, 4 - i];
    }
    return soma;
}

```

```
// Esse static void vai calcular a soma de todos os elementos
static int SomaTodosElementos(int[,] matriz)
{
    int soma = 0;
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            soma += matriz[i, j];
        }
    }
    return soma;
}
}
```

Exerc8.cs

```
using System;
class Exerc8 {

    public static void Rodar ()
    {
        // Declararei a matriz 4x4
        int[,] matriz = new int[4, 4];

        // Vai preencher a matriz
        PreencherMatriz(matriz);

        // Vai exibir a matriz preenchida
        Console.WriteLine("Matriz 4x4 preenchida:");
        ExibirMatriz(matriz);

        // Vai exibir os elementos da diagonal principal
        Console.WriteLine("Elementos da diagonal principal:");
        ExibirDiagonalPrincipal(matriz);

        // Vai calcular e exibir a soma dos elementos abaixo da diagonal principal
        int somaAbaixoDiagonal = SomaAbaixoDiagonalPrincipal(matriz);
        Console.WriteLine($"Soma dos elementos abaixo da diagonal principal:
{somaAbaixoDiagonal}");
    }

    static void PreencherMatriz(int[,] matriz)
    {
        Random random = new Random();
        for (int i = 0; i < 4; i++)
        {
            for (int j = 0; j < 4; j++)
            {
                matriz[i, j] = random.Next(1, 101);
            }
        }
        // Vai gerar números aleatórios entre 1 e 100
    }

    static void ExibirMatriz(int[,] matriz)
    {
        for (int i = 0; i < 4; i++)
        {
            for (int j = 0; j < 4; j++)
            {
                Console.Write(matriz[i, j] + "\t");
            }
        }
    }
}
```



```

        Console.WriteLine();
    }
}

static void ExibirDiagonalPrincipal(int[,] matriz)
{
    for (int i = 0; i < 4; i++)
    {
        Console.WriteLine(matriz[i, i]);
    }
}

static int SomaAbaixoDiagonalPrincipal(int[,] matriz)
{
    int soma = 0;
    for (int i = 1; i < 4; i++)
    // começa do índice 1, pois não há elementos abaixo da diagonal na linha 0
    {
        for (int j = 0; j < i; j++)
        {
            soma += matriz[i, j];
        }
    }
    return soma;
}
}

```

Exerc9.cs

using System;

class Exerc9

```
{
    public static void Rodar ()
    {
        // Declararei as matrizes 4x6
        int[,] matrizA = new int[4, 6];
        int[,] matrizB = new int[4, 6];

        // Vai preencher as matrizes
        PreencherMatriz(matrizA, "A");
        PreencherMatriz(matrizB, "B");

        // Vai exibir as matrizes preenchidas
        Console.WriteLine("Matriz A:");
        ExibirMatriz(matrizA);
        Console.WriteLine("Matriz B:");
        ExibirMatriz(matrizB);

        // Vai calcular e exibir a matriz S (soma de A e B)
        int[,] matrizS = SomaMatrizes(matrizA, matrizB);
        Console.WriteLine("Matriz S (A + B):");
        ExibirMatriz(matrizS);

        // Vai calcular e exibir a matriz D (diferença de A e B)
        int[,] matrizD = DiferencaMatrizes(matrizA, matrizB);
        Console.WriteLine("Matriz D (A - B):");
        ExibirMatriz(matrizD);
    }
}
```

static void PreencherMatriz(int[,] matriz, string nome)

```
{
    Random random = new Random();
    for (int i = 0; i < matriz.GetLength(0); i++)
    {
        for (int j = 0; j < matriz.GetLength(1); j++)
        {
            matriz[i, j] = random.Next(1, 101);
        }
        // Vai gerar números aleatórios entre 1 e 100
    }
}
```

static void ExibirMatriz(int[,] matriz)

```
{
```

```

    for (int i = 0; i < matriz.GetLength(0); i++)
    {
        for (int j = 0; j < matriz.GetLength(1); j++)
        {
            Console.Write(matriz[i, j] + "\t");
        }
        Console.WriteLine();
    }
}

static int[,] SomaMatrizes(int[,] matrizA, int[,] matrizB)
{
    int[,] matrizS = new int[4, 6];
    for (int i = 0; i < matrizA.GetLength(0); i++)
    {
        for (int j = 0; j < matrizA.GetLength(1); j++)
        {
            matrizS[i, j] = matrizA[i, j] + matrizB[i, j];
        }
    }
    return matrizS;
}

static int[,] DiferencaMatrizes(int[,] matrizA, int[,] matrizB)
{
    int[,] matrizD = new int[4, 6];
    for (int i = 0; i < matrizA.GetLength(0); i++)
    {
        for (int j = 0; j < matrizA.GetLength(1); j++)
        {
            matrizD[i, j] = matrizA[i, j] - matrizB[i, j];
        }
    }
    return matrizD;
}
}

```

Exerc10.cs

```
using System;
class Exerc10 {

    public static void Rodar ()
    {
        // Declararei a matriz 10x10
        int[,] matriz = new int[10, 10];

        // Vai preencher a matriz
        PreencherMatriz(matriz);

        // Vai exibir a matriz preenchida
        Console.WriteLine("Matriz original 10x10:");
        ExibirMatriz(matriz);

        // Vai trocar a linha 2 com a linha 8
        TrocarLinhas(matriz, 1, 7);
        Console.WriteLine("Matriz após trocar a linha 2 com a linha 8:");
        ExibirMatriz(matriz);

        // Vai trocar a coluna 4 com a coluna 10
        TrocarColunas(matriz, 3, 9);
        Console.WriteLine("Matriz após trocar a coluna 4 com a coluna 10:");
        ExibirMatriz(matriz);

        // Vai trocar a diagonal principal com a diagonal secundária
        TrocarDiagonais(matriz);
        Console.WriteLine("Matriz após trocar a diagonal principal com a diagonal secundária:");
        ExibirMatriz(matriz);

        // Vai trocar a linha 5 com a coluna 10
        TrocarLinhaComColuna(matriz, 4, 9);
        Console.WriteLine("Matriz após trocar a linha 5 com a coluna 10:");
        ExibirMatriz(matriz);
    }

    // Esse static void vai preencher a matriz
    static void PreencherMatriz(int[,] matriz)
    {
        Random random = new Random();
        for (int i = 0; i < 10; i++)
        {
            for (int j = 0; j < 10; j++)
            {
                matriz[i, j] = random.Next(1, 101); // Vai gerar números aleatórios entre 1 e 100
            }
        }
    }
}
```

```

    }
  }
}

```

// Esse static void vai exibir a matriz
 static void ExibirMatriz(int[,] matriz)

```

{
    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < 10; j++)
        {
            Console.Write(matriz[i, j] + "\t");
        }
        Console.WriteLine();
    }
}

```

// Esse static void vai trocar a linha 2 com a linha 8
 static void TrocarLinhas(int[,] matriz, int linha1, int linha2)

```

{
    for (int j = 0; j < 10; j++)
    {
        int temp = matriz[linha1, j];
        matriz[linha1, j] = matriz[linha2, j];
        matriz[linha2, j] = temp;
    }
}

```

// Esse static void vai trocar a coluna 4 com a coluna 10
 static void TrocarColunas(int[,] matriz, int coluna1, int coluna2)

```

{
    for (int i = 0; i < 10; i++)
    {
        int temp = matriz[i, coluna1];
        matriz[i, coluna1] = matriz[i, coluna2];
        matriz[i, coluna2] = temp;
    }
}

```

// Esse static void vai trocar a diagonal principal com a diagonal secundária
 static void TrocarDiagonais(int[,] matriz)

```

{
    for (int i = 0; i < 10; i++)
    {
        int temp = matriz[i, i];
        matriz[i, i] = matriz[i, 9 - i];
        matriz[i, 9 - i] = temp;
    }
}

```

```
// Esse static void vai trocar a linha 5 com a coluna 10
static void TrocarLinhaComColuna(int[,] matriz, int linha, int coluna)
{
    for (int i = 0; i < 10; i++)
    {
        int temp = matriz[linha, i];
        matriz[linha, i] = matriz[i, coluna];
        matriz[i, coluna] = temp;
    }
}
```