

Exe 1:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Formatters;
using System.Text;
using System.Threading.Tasks;

namespace exe_1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int linha, coluna;

            int diagonalP = 0;
            int diagonalS = 0;

            Console.WriteLine("Informe a quantidade de linhas da matriz:");
            linha = int.Parse(Console.ReadLine());

            Console.WriteLine("Informe a quantidade de colunas da matriz:");
            coluna = int.Parse(Console.ReadLine());

            int[,] matriz = new int[linha, coluna];

            for (int i = 0; i < matriz.GetLength(0); i++)
            {
                for (int j = 0; j < matriz.GetLength(1); j++)
                {
```

```
        Console.WriteLine($"Informe a posição linha {i} coluna {j}:");  
        matriz[i, j] = int.Parse(Console.ReadLine());  
    }  
}
```

```
// Soma das Linhas //
```

```
for (int i = 0; i < matriz.GetLength(0); i++)  
{  
    int somaL = 0;  
    for (int j = 0; j < matriz.GetLength(1); j++)  
    {  
        somaL = somaL + matriz[i, j];  
    }  
    Console.WriteLine($"A soma da linha {i} é: {somaL}");  
}  
Console.WriteLine();
```

```
// Soma das Colunas //
```

```
for (int j = 0; j < matriz.GetLength(1); j++)  
{  
    int somaC = 0;  
    for (int i = 0; i < matriz.GetLength(0); i++)  
    {  
        somaC = somaC + matriz[i, j];  
    }  
    Console.WriteLine($"A soma da coluna {j} é: {somaC}");  
}  
Console.WriteLine();
```

```
// Soma da Diagonal Principal //
```

```
for (int i = 0, j = 0; i < matriz.GetLength(0); i++, j++)
```

```
{
```

```
    diagonalP = diagonalP + matriz[i, j];
```

```
}
```

```
Console.WriteLine($"A soma da diagonal Principal é {diagonalP}");
```

```
// Soma da Diagonal Secundária //
```

```
for (int i = 0; i < matriz.GetLength(0); i++)
```

```
{
```

```
    diagonalS = diagonalS + matriz[i, matriz.GetLength(0) - 1 - i];
```

```
}
```

```
Console.WriteLine($"A soma da diagonal Secundária é {diagonalS}");
```

```
Console.ReadLine();
```

```
for (int i = 0; i < matriz.GetLength(0); i++)
```

```
{
```

```

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

        Console.WriteLine();
    }

    Console.ReadLine();
}
}
}

```

Exe 2:

```

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

```

```

namespace exe_2

```

```

{
    internal class Program
    {
        static void Main(string[] args)
        {
            int n, m;
            string comparacao = "nada";

            Console.WriteLine("Informe a quantidade de linhas das matrizes:");
            n = int.Parse(Console.ReadLine());

```

```
Console.WriteLine("Informe a quantidade de colunas das matrizes:");
```

```
m = int.Parse(Console.ReadLine());
```

```
Console.WriteLine("-----");
```

```
Console.WriteLine("Matriz 1");
```

```
int[,] matriz1 = new int[n, m];
```

```
for (int i = 0; i < matriz1.GetLength(0); i++)
```

```
{
```

```
    for (int j = 0; j < matriz1.GetLength(1); j++)
```

```
    {
```

```
        Console.Write($"Informe a posição da linha {i} coluna {j}:");
```

```
        matriz1[i, j] = int.Parse(Console.ReadLine());
```

```
    }
```

```
}
```

```
Console.WriteLine("-----");
```

```
Console.WriteLine("Matriz 2");
```

```
int[,] matriz2 = new int[n, m];
```

```
for (int i = 0; i < matriz2.GetLength(0); i++)
```

```
{
```

```
    for (int j = 0; j < matriz2.GetLength(1); j++)
```

```
    {
```

```
        Console.Write($"Informe a posição da linha {i} coluna {j}:");
```

```
        matriz2[i, j] = int.Parse(Console.ReadLine());
```

```
    }
```

```
}
```

```

for (int i = 0, check = 0 ; i < matriz1.GetLength(0) || check == 1; i++)
{
    for (int j = 0; j < matriz1.GetLength(1); j++)
    {
        if (matriz1[i,j] == matriz2[i,j])
        {
            comparacao = "são iguais";
        }
        else
        {
            comparacao = "não são iguais";
            check++;
        }
    }
}

Console.WriteLine();

Console.WriteLine($"As matrizes {comparacao}");

Console.ReadLine();
}
}
}

```

Exe 3:

```

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

```

```
namespace exe_3
{
    internal class Program
    {
        static void Main(string[] args)
        {
            // declaração de vaiaveis //

            // matriz //
            double[,] matriz = new double[10, 10];

            // leitura da matriz //

            for (int i = 0; i < matriz.GetLength(0); i++)
            {
                for (int j = 0; j < matriz.GetLength(1); j++)
                {
                    Console.WriteLine($"Informe a posição linha {i} coluna {j}");
                    matriz[i, j] = double.Parse(Console.ReadLine());
                }
            }

            // Impressão da matriz //

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

```

    }

    Console.WriteLine();
}

// Troca de elementos //

for (int i = 0, j = 0; i < matriz.GetLength(0); i++, j++)
{
    if (i == j)
    {
        matriz[i, j] = Math.Sqrt(matriz[i, j]);
    }
}

// impressão da matriz com a troca de elementos da dp //

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

    Console.WriteLine();
}

Console.ReadLine();

}

}

}

```


Exe 4:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace exe_4
{
    internal class Program
    {
        static void Main(string[] args)
        {
            double somacP = 0, somaC35 = 0, somaC12= 0;

            double[,] matriz = new double[3, 6];

            // Leitura da matriz //
            for (int i = 0; i < matriz.GetLength(0); i++)
            {
                for (int j = 0; j < matriz.GetLength(1); j++)
                {
                    Console.Write($"Informe a posição [linha {i}, coluna {j}]:");
                    matriz[i, j] = double.Parse(Console.ReadLine());
                }
            }

            // imprimindo matriz //
            Console.WriteLine();
```

```

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

// somando elementos das colunas pares da matriz //
for (int j = 0; j < matriz.GetLength(1); j = j + 2)
{
    for (int i = 0; i < matriz.GetLength(0); i++)
    {
        somacP = somacP + matriz[i, j];
    }
}
Console.WriteLine($"A soma das colunas pares é: {somacP}");

// soma das colunas 3 e 5 //
for (int j = 3; j < matriz.GetLength(1); j = j + 2)
{
    for (int i = 0; i < matriz.GetLength(0); i++)
    {
        somaC35 = somaC35 + matriz[i, j];
    }
}
Console.WriteLine($"Soma coluna 3 e 5: {somaC35}");
Console.WriteLine();

```

```

// soma das colunas 1 e 2 //
for (int j = 1; j < matriz.GetLength(1); j++)
{
    for (int i = 0; i < matriz.GetLength(0); i++)
    {
        if (j == 1 || j == 2)
        {
            somaC12 = somaC12 + matriz[i, j];
        }
    }
}

// substituição dos elementos da coluna 5 pela soma das c12 //
for (int j = 5; j < matriz.GetLength(1); j++)
{
    for (int i = 0; i < matriz.GetLength(0); i++)
    {
        matriz[i, j] = somaC12;
    }
}

// imprimindo matriz //
Console.WriteLine("Matriz com elementos da coluna 5 com o valor da soma da coluna 1 e 2");
for (int i = 0; i < matriz.GetLength(0); i++)
{
    for (int j = 0; j < matriz.GetLength(1); j++)
    {
        Console.Write(matriz[i, j] + " ");
    }
}

```

```
        Console.WriteLine();  
    }  
  
    Console.ReadLine();  
  
    }  
    }  
}
```

Exe 5:

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace exe_5  
{  
    internal class Program  
    {  
        static void Main(string[] args)  
        {  
            int somaL1 = 0;  
  
            int[,] matriz = new int[3, 3];  
  
            // Leitura da Matriz //  
            for (int i = 0; i < matriz.GetLength(0); i++)
```

```
{
    for (int j = 0; j < matriz.GetLength(1); j++)
    {
        Console.WriteLine($"Informe a posição da [linha {i}, coluna {j}]:");
        matriz[i, j] = int.Parse(Console.ReadLine());
    }
}
```

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

```
// Soma das linhas //
for (int i = 0; i < matriz.GetLength(0); i++)
{
    for (int j = 0; j < matriz.GetLength(1); j++)
    {
        if (i == 0)
        {
            somaL1 = somaL1 + matriz[i, j];
        }
    }
}

Console.WriteLine(somaL1);
```

```
        Console.ReadLine();

    }

}

}
```

Exe 6:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace exe_6
{
    internal class Program
    {
        static void Main(string[] args)
        {
            // declaração de variáveis //

            char check = 'f';

            int[,] matriz = new int[4, 4];

            // Leitura da Matriz //
            for (int i = 0; i < matriz.GetLength(0); i++)
            {
                for (int j = 0; j < matriz.GetLength(1); j++)
```

```

    {
        Console.WriteLine($"Informe a posição da [linha {i}, coluna {j}]:");
        matriz[i, j] = int.Parse(Console.ReadLine());
    }
}

Console.WriteLine();

// Impressão da Matriz //
Console.WriteLine("Matriz:");
for (int i = 0; i < matriz.GetLength(0); i++)
{
    for (int j = 0; j < matriz.GetLength(1); j++)
    {
        Console.Write(matriz[i, j] + " ");
    }
    Console.WriteLine();
}

// Leitura da matriz, com verificação das posições //
for (int i = 0; i < matriz.GetLength(0); i++)
{
    for (int j = 0; j < matriz.GetLength(1); j++)
    {
        if (matriz[1,0] == 0 && matriz[2, 0] == 0 && matriz[3, 0] == 0 && matriz[2, 1] == 0
&& matriz[3, 1] == 0 && matriz[3, 2] == 0)
        {
            check = 'v';
        }
    }
}

```

```

// Resultado dependendo da verificação //
if (check == 'v')
{
    Console.WriteLine("A matriz é um quadrado perfeito.");
}
else
{
    Console.WriteLine("A matriz não é um quadrado perfeito.");
}

Console.ReadLine();
}
}
}

```

Exe 7: parei nessa parte e não consegui mais:

```

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

namespace exe_7
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int m, n, entrada = 0, cont = 0;

```



```
Console.WriteLine("Informe a quantidade de linhas da matriz:");
m = int.Parse(Console.ReadLine());

Console.WriteLine("Informe a quantidade de colunas da matriz:");
n = int.Parse(Console.ReadLine());

int[,] matriz = new int[m, n];

for (int i = 0; i < matriz.GetLength(0); i++)
{
    for (int j = 0; j < matriz.GetLength(1); j++)
    {
        Console.Write($"Informe a posição [linha {i}, coluna {j}]:");
        matriz[i, j] = int.Parse(Console.ReadLine());
    }
}

for (int i = 0; i < matriz.GetLength(0); i++)
{
    for (int j = 0; j < matriz.GetLength(1); i++)
    {
        if (matriz[i, j] == 0)
        {
            }

        }
    }
}
}
```

Exe 8: Não consegui fazer.