

Lista de Exercícios 1 – Vetores

1. Escreva um programa que leia ou gere um vetor de N elementos inteiros (N deve ser informado pelo usuário) e passe o mesmo como parâmetro para uma função que retorne a soma de seus elementos.

```
using System;
```

```
class Program
```

```
{
```

```
static void Main(string[] args)
```

```
{
```

```
Console.WriteLine("Digite o tamanho do vetor : ");
```

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

```
int[] vetor = new int[n];
```

```
for (int i = 0; i < n; i++)
```

```
{
```

```
Console.WriteLine($"Digite o elemento {i + 1}: ");
```

```
vetor[i] = int.Parse(Console.ReadLine());
```

```
}
```

```
int soma = calculaSoma(vet);
```

```
Console.WriteLine($"A soma dos elementos do vetor é: {soma}");
```

```
}
```

```

static int calculaSoma(int[] vet)
{
    int soma = 0;
    foreach (int num in vet)
    {
        soma += num;
    }
    return soma;
}

```

2. Dado um vetor de n números reais, crie uma função que retorne o maior elemento do vetor, apresente o vetor.

```

using System;
using Arrays;

namespace Hello_World
{
    class Exercicio2_e_3
    {
        static double maiorValor(double[] vetor)

        {
            double maior = vetor[0];
            for (int i = 1; i < vetor.Length; i++)
            {
                if (vetor[i] > maior)
                    maior = vetor[i];
            }
            return maior;
        }
    }
}

```

```
static double menorValor(double[] vetor)
```

```
{  
    double menor = vetor[0];  
    for (int i = 1; i < vetor.Length; i++)  
    {  
        if (vetor[i] > menor)  
            menor = vetor[i];  
    }  
    return menor;  
}
```

```
static void Main()
```

```
{  
    int n;  
    Console.WriteLine("Tamanho do vetor");  
    n = int.Parse(Console.ReadLine());  
    double[] numeros = new double[n];  
    BibliotecaArray.geraVetor(numeros);  
    BibliotecaArray.mostraVetor(numeros);  
    double res = maiorValor(numeros);  
    Console.WriteLine($"\\n Maior valor:{res:F2}");  
    Console.WriteLine($"Menor valor:{menorValor(numeros):F2}");  
    Console.ReadKey();  
}  
}
```

- 3.** Dado um vetor de n números reais, crie uma função que retorne o menor elemento do vetor, apresente o vetor.

```
using System;
```

```
using Arrays;
```

```
namespace Hello_World
```

```

{
class Exercicio2_e_3
{
    static double maiorValor(double[] vetor)

    {
        double maior = vetor[0];
        for (int i = 1; i < vetor.Length; i++)
        {
            if (vetor[i] > maior)
                maior = vetor[i];
        }
        return maior;
    }

    static double menorValor(double[] vetor)

    {
        double menor = vetor[0];
        for (int i = 1; i < vetor.Length; i++)
        {
            if (vetor[i] > menor)
                menor = vetor[i];
        }
        return menor;
    }

    static void Main()
    {
        int n;
        Console.WriteLine("Tamanho do vetor");
        n = int.Parse(Console.ReadLine());
        double[] numeros = new double[n];
        BibliotecaArray.geraVetor(numeros);
        BibliotecaArray.mostraVetor(numeros);
        double res = maiorValor(numeros);
        Console.WriteLine($"\\n Maior valor:{res:F2}");
        Console.WriteLine($"Menor valor:{menorValor(numeros):F2}");
    }
}

```

```

        Console.ReadKey();
    }
}
}

```

4. Escreva um programa que leia ou gere um vetor de N elementos inteiros. A seguir, crie uma função que receba esse vetor e conte quantos valores ímpares existem no vetor. Retorne a quantidade de ímpares.

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main(string[] args)
```

```
    {
```

```
        Console.WriteLine("Informe o tamanho do vetor : ");
```

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

```
        int[] vet = new int[n];
```

```
        for (int i = 0; i < n; i++)
```

```
        {
```

```
            Console.WriteLine($"Informe o elemento {i + 1}: ");
```

```
            vet[i] = int.Parse(Console.ReadLine());
```

```
        }
```

```
        int quantImpares = ContImpares(vet);
```

```
        Console.WriteLine($"A quantidade de valores ímpares no vetor é: {quantImpares}");
```

```
    }
```

```
    static int ContImpares(int[] vet)
```

```
    {
```

```
        int cont = 0;
```

```
        foreach (int num in vet)
```

```
        {
```

```

        if (num % 2 != 0)
        {
            cont++;
        }
    }
    return cont;
}
}

```

5. Leia um vetor DNA de caracteres para receber as letras A, T, C e G que representam as bases do DNA. Este vetor será responsável por representar uma fita de um gene de até 50 bases. Gere o vetor complementar ao vetor DNA e o apresente (Lembrando as bases complementares A=T C=G).

using System;

```

namespace ConsoleApp1
{
    internal class ex5
    {
        static void Main()
        {
            string dna;
            Console.WriteLine("Entre com as bases");
            dna = Console.ReadLine();
            int n = dna.Length;
            char[] complementar = new char[n];
            for (int i = 0; i < n; i++)
            {
                switch (dna[i])
                {
                    case 'T':
                        complementar[i] = 'A'; break;
                    case 'A':
                        complementar[i] = 'T'; break;
                    case 'G':
                        complementar[i] = 'C'; break;
                }
            }
        }
    }
}

```

```

    }
} //fim for
//vetor complementar
Console.WriteLine("***DNA***");
Console.WriteLine(dna);
for (int i = 0; i < n; i++)
    Console.Write("Complementar[i]");
Console.ReadKey();
}
}
}

```

6. Escreve um programa que sorteio, aleatoriamente, N números e armazene estes em um vetor. Em seguida, o usuário digita um número e seu programa em C deve acusar se o número digitado está no vetor ou não. Se estiver, diga a posição que está. using System;

```

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

        Console.WriteLine("Informe o tamanho do vetor : ");
        int n = int.Parse(Console.ReadLine());

        int[] vetor = new int[n];

        Random random = new Random();

        for (int i = 0; i < n; i++)
        {
            vetor[i] = random.Next(1, 101);
        }
    }
}

```

```
Console.WriteLine("Números sorteados:");
```

```
foreach (int num in vetor)
```

```
{
```

```
    Console.Write(num + " ");
```

```
}
```

```
Console.WriteLine();
```

```
Console.Write("Digite um número para verificar se está no vetor: ");
```

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

```
int posicao = verificaNum(vet, numeroParaVerifica);
```

```
if (posicao != -1)
```

```
{
```

```
    Console.WriteLine($"O número {numeroParaVerificar} está na posição {posicao} do vetor.");
```

```
}
```

```
else
```

```
{
```

```
    Console.WriteLine($"O número {numeroParaVerificar} não está no vetor.");
```

```
}
```

```
}
```

```
// Função que verifica se o número está no vetor e retorna sua posição
```

```
static int VerificarNumero(int[] vetor, int num)
```

```
{
```

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

```
    {
```

```
        if (vetor[i] == num)
```

```
        {
```

```
            return i;
```

```
        }
```

```
    }
```

```
    return -1;
```

```
}
```

```
}
```


7. Escreva um programa que leia ou gere dois **vetores** de N posições e faça a multiplicação dos elementos de mesmo índice, colocando o resultado em um terceiro vetor. Mostre o vetor resultante.

```
using System;
```

```
class Program
{
    static double [] lerVetor(int tamanho)
    {
        double[] vetor = new double[tamanho];
        for (int i = 0; i < tamanho; i++)
        {
            Console.Write($"Digite um valor pra posição: ");
            vetor[i] = Convert.ToDouble(Console.ReadLine());
        }
        return vetor;
    }

    static void Main(string[] args)
    {
        Console.Write("Digite o tamanho dos vetores: ");
        int N = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Digite os elementos do primeiro vetor:");
        double[] vetor1 = lerVetor(N);

        Console.WriteLine("Digite os elementos do segundo vetor:");
        double[] vetor2 = lerVetor(N);

        double[] vetorResultante = new double[N];
        for (int i = 0; i < N; i++)
        {
            vetorResultante[i] = vetor1[i] * vetor2[i];
        }
    }
}
```

```

        Console.WriteLine("O vetor resultante da multiplicação é:");
        Console.WriteLine(string.Join(" ", vetorResultante));
    }
}

```

- 8.** Leia um vetor de caracteres. Utilize a função Length para obter a quantidade de elementos do vetor de caracteres. Escreva o vetor lido em ordem inversa.

```
using System;
```

```

class Program
{
    static void Main ()

```

```

    {
        {

```

```

            Console.Write("Digite o tamanho do vetor de caracteres: ");
            int tamanho = Convert.ToInt32(Console.ReadLine());

```

```

            char[] vetor = new char[tamanho];

```

```

            Console.WriteLine("Digite caracteres:");
            for (int i = 0; i < tamanho; i++)
            {
                Console.Write($"Posição ");
                vetor[i] = Console.ReadKey().KeyChar;
                Console.WriteLine(); /
            }

```

```

            Console.WriteLine($"
Quantidade de elementos no vetor: ");

```

```

            Console.WriteLine("Vetor em ordem inversa:");
            for (int i = vetor.Length - 1; i >= 0; i--)
            {
                Console.Write(vetor[i]);
            }

```

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

9. Receber um vetor de N posições do tipo inteiro verificar quantas vezes um dado valor informado pelo usuário se encontra no vetor. Apresente também todos elementos do vetor.

```
using System;
```

```
class Program
```

```
{  
    static void Main()  
    {
```

```
        Console.Write("Digite o tam do vetor: ");  
        int tam = Convert.ToInt32(Console.ReadLine());
```

```
        int[] vetor = new int[tam];
```

```
        Console.WriteLine("Digite os elementos do vetor:");  
        for (int i = 0; i < tam; i++)  
        {  
            Console.Write($"Elemento {i + 1}: ");  
            vetor[i] = Convert.ToInt32(Console.ReadLine());  
        }
```

```
        Console.Write("Digite o valor a ser contado no vetor: ");  
        int valorConta = Convert.ToInt32(Console.ReadLine());
```

```
        int cont = 0;  
        for (int i = 0; i < tamanho; i++)  
        {  
            if (vetor[i] == valorConta)  
            {  
                cont++;  
            }  
        }
```

```

    }
}

Console.WriteLine("\nElementos do vetor:");
for (int i = 0; i < tam; i++)
{
    Console.Write(vetor[i] + " ");
}
Console.WriteLine();
Console.WriteLine($"O valor aparece {vezes} vezes no vetor.");
}
}

```

10. Tentando descobrir se um dado era viciado, um dono de cassino honesto o lançou N vezes. Dados os n resultados dos lançamentos que devem ser armazenados em um vetor, determinar o número de ocorrências de cada face.

11. Uma brincadeira que crianças adoram é se comunicar na *língua do P*, acrescentando *pê* antes de cada sílaba, como uma forma de código para dificultar que outras pessoas entendam a conversa (pê-va pê-mos pê-no pê-ci pê-ne pê-ma?). Jacy e Kátia adaptaram a língua do P para mensagens eletrônicas, acrescentando a letra P minúscula 'p' antes de cada letra das palavras de uma mensagem. Um exemplo de mensagem codificada e a respectiva mensagem decodificada é mostrada na figura abaixo.

```
using System;
```

```
namespace ConsoleApp1
```

```
{
```

```
    class Exercicio11
```

```
    {
```

```
        static void Main()
```

```
        {
```

```

string frase;

Console.WriteLine("Entre com a frase codificada");

frase = Console.ReadLine();

for(int i = 0;i<frase.Length -1;i++)

{

    if(frase[i] != 'p')

        Console.WriteLine(frase[i]);

    else

        if (frase[i+1]=='p')

            Console.Write(frase[i]);

}

Console.ReadKey();

}

}

}

```

Mensagem codificada	Mensagem decodificada
pVpampops papo pcpipnpepmpa	Vamos ao cinema

Sua tarefa é escrever um programa que decodifique uma mensagem escrita na língua do P eletrônica de Jacy e Kátia.

COMPUTAÇÃO
PROF. MATHEUS FRANCO

Entrada A entrada consiste de uma única linha, contendo uma mensagem escrita na língua do P eletrônico de Jacy e Kátia.

Saída Seu programa deve produzir uma única linha, contendo a mensagem decodificada.

Restrições A mensagem contém apenas letras maiúsculas e minúsculas e espaços em branco. A mensagem tem entre 1 e 1000 caracteres. Não há dois espaços em branco consecutivos na mensagem.

Exemplos

Entrada pUpm pfpiplpmpe plpepgpapl	Saída Um filme legal
Entrada pA pppapppa pdpo pPpapppa	Saída A papa do Papa

Dica: Para cada posição verifique o caractere armazenado antes de
apresenta-lo. **12. Carnaval**

O Carnaval é um feriado celebrado normalmente em fevereiro; em muitas cidades brasileiras, a principal atração são os desfiles de escolas de samba. As várias agremiações desfilam ao som de seus sambas-enredos e são julgadas pela liga das escolas de samba para determinar a campeã do Carnaval.

Cada agremiação é avaliada em vários quesitos; em cada quesito, cada escola recebe cinco notas que variam de 5,0 a 10,0. A nota final da escola em um dado quesito é a soma das três notas centrais recebidas pela escola, excluindo a maior e a menor das cinco notas.

Como existem muitas escolas de samba e muitos quesitos, o presidente da liga pediu que você escrevesse um programa que, dadas as notas da agremiação, calcula a sua nota final num dado quesito.

Entrada

A entrada contém uma única linha, contendo cinco números N_i ($1 \leq i \leq 5$), todos com uma casa decimal, indicando as notas recebidas pela agremiação em um dos quesitos.

Saída

Seu programa deve imprimir uma única linha, contendo um único número com exatamente uma casa decimal, a nota final da escola de samba no quesito considerado.

```
using System;
```

```
class Program
```

```
{
```

```
static void Main()

{

    double[] notas = new double[5];


    Console.WriteLine("Digite as notas :");

    for (int i = 0; i < 5; i++)

    {

        Console.Write($"Nota ");

        notas[i] = Convert.ToDouble(Console.ReadLine());


        while (notas[i] < 5.0 || notas[i] > 10.0)

        {

            Console.WriteLine("Nota inválida! ");

            Console.Write($"Nota {i + 1}: ");

            notas[i] = Convert.ToDouble(Console.ReadLine());

        }

    }


    Array.Sort(notas);
```

```
double notaFinal = notas[1] + notas[2] + notas[3];
```

Exemplos

Entrada	Saída
6.4 8.2 8.2 7.4 9.1	29.8



COMPUTAÇÃO
PROF. MATHEUS FRANCO

Entrada	Saída
10.0 10.0 5.0 5.0 10.0	25.0