

1)

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main()
```

```
    {
```

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

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

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

```
        Console.WriteLine("\nDigite os elementos do vetor:");
```

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

```
        {
```

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

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

```
        }
```

```
        int soma = SomarElementos(vetor);
```

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

```
    }
```

```
    static int SomarElementos(int[] vetor)
```

```
    {
```

```
        int soma = 0;
```

```
        foreach (int elemento in vetor)
```

```
        {
```

```
            soma += elemento;
```

```
        }
```

```
        return soma;
```

```
    }
```

```
}
```

2)

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        double[] vetor = { 1.5, 3.2, 7.8, 4.6, 9.0, 2.4 };
```

```

        Console.WriteLine("Vetor: [" + string.Join(", ", vetor) + "]");

        double maiorElemento = EncontrarMaiorElemento(vetor);

        Console.WriteLine("Maior elemento do vetor: " + maiorElemento);
    }

    static double EncontrarMaiorElemento(double[] vetor)
    {
        double maior = vetor[0];

        foreach (double numero in vetor)
        {
            if (numero > maior)
            {
                maior = numero;
            }
        }

        return maior;
    }
}

3)
using System;

class Program
{
    static void Main()
    {
        double[] vetor = { 1.5, 3.2, 7.8, 4.6, 9.0, 2.4 };

        Console.WriteLine("Vetor: [" + string.Join(", ", vetor) + "]");

        double menorElemento = EncontrarMenorElemento(vetor);

        Console.WriteLine("Menor elemento do vetor: " + menorElemento);
    }

    static double EncontrarMenorElemento(double[] vetor)
    {
        double menor = vetor[0];

        foreach (double numero in vetor)

```

```

    {
        if (numero < menor)
        {
            menor = numero;
        }
    }

    return menor;
}
}

```

4)

```
using System;
```

```
class Program
```

```

{
    static void Main()
    {

        Console.WriteLine("Digite o número de elementos do vetor: ");
        int n = int.Parse(Console.ReadLine());

        int[] vetor = new int[n];
        Random rand = new Random();

        for (int i = 0; i < n; i++)
        {
            vetor[i] = rand.Next(1, 101); // Gera números entre 1 e 100
        }

        Console.WriteLine("Vetor: [" + string.Join(", ", vetor) + "]");

        int quantidadeImpares = ContarImpares(vetor);

        Console.WriteLine("Quantidade de valores ímpares no vetor: " + quantidadeImpares);
    }

    static int ContarImpares(int[] vetor)
    {
        int contador = 0;

        foreach (int numero in vetor)
        {
            if (numero % 2 != 0) // Verifica se o número é ímpar
            {

```

```

        contador++;
    }
}

return contador;
}
}

```

5)

using System;

class Program

{

static void Main()

{

Console.Write("Digite a sequência de DNA (máximo 50 bases - use apenas A, T, C, G):");

string inputDNA = Console.ReadLine().ToUpper();

if (inputDNA.Length > 50 || !IsSequenciaValida(inputDNA))

{

Console.WriteLine("Sequência inválida! Insira até 50 caracteres contendo apenas A, T, C e G.");

return;

}

char[] dna = inputDNA.ToCharArray();

char[] dnaComplementar = GerarVetorComplementar(dna);

Console.WriteLine("Fita DNA: " + new string(dna));

Console.WriteLine("Fita DNA Complementar: " + new string(dnaComplementar));

}

// Função para verificar se a sequência de DNA é válida

static bool IsSequenciaValida(string sequencia)

{

foreach (char baseDNA in sequencia)

{

if (baseDNA != 'A' && baseDNA != 'T' && baseDNA != 'C' && baseDNA != 'G')

{

return false;

}

}

```

        return true;
    }

    static char[] GerarVetorComplementar(char[] dna)
    {
        char[] dnaComplementar = new char[dna.Length];

        for (int i = 0; i < dna.Length; i++)
        {
            switch (dna[i])
            {
                case 'A':
                    dnaComplementar[i] = 'T';
                    break;
                case 'T':
                    dnaComplementar[i] = 'A';
                    break;
                case 'C':
                    dnaComplementar[i] = 'G';
                    break;
                case 'G':
                    dnaComplementar[i] = 'C';
                    break;
            }
        }

        return dnaComplementar;
    }
}

```

6)

using System;

class Program

```

{
    static void Main()
    {

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

        int[] vetor = new int[n];
        Random rand = new Random();

        Console.WriteLine("Vetor sorteado: ");
    }
}

```

```

for (int i = 0; i < n; i++)
{
    vetor[i] = rand.Next(1, 101);
    Console.Write(vetor[i] + " ");
}
Console.WriteLine();

```

```

Console.Write("Digite o número que deseja procurar no vetor: ");
int numeroProcurado = int.Parse(Console.ReadLine());

```

```

bool encontrado = false;
for (int i = 0; i < n; i++)
{
    if (vetor[i] == numeroProcurado)
    {
        Console.WriteLine("Número {0} encontrado na posição {1}.", numeroProcurado, i);
        encontrado = true;
        break; // Sai do loop ao encontrar o número
    }
}

```

```

if (!encontrado)
{
    Console.WriteLine("Número {0} não encontrado no vetor.", numeroProcurado);
}
}
}

```

7)
using System;

```

class Program
{
    static void Main()
    {
        Console.Write("Digite o número de posições dos vetores: ");
        int n = int.Parse(Console.ReadLine());

```

```

        int[] vetor1 = new int[n];
        int[] vetor2 = new int[n];
        int[] vetorResultado = new int[n];
        Random rand = new Random();

```

```

        Console.WriteLine("Vetor 1: ");

```

```

for (int i = 0; i < n; i++)
{
    vetor1[i] = rand.Next(1, 10); // Gera números entre 1 e 9
    Console.Write(vetor1[i] + " ");
}
Console.WriteLine();

Console.WriteLine("Vetor 2: ");
for (int i = 0; i < n; i++)
{
    vetor2[i] = rand.Next(1, 10); // Gera números entre 1 e 9
    Console.Write(vetor2[i] + " ");
}
Console.WriteLine();

for (int i = 0; i < n; i++)
{
    vetorResultado[i] = vetor1[i] * vetor2[i];
}

Console.WriteLine("Vetor Resultado (multiplicação de elementos de mesmo índice): ");
for (int i = 0; i < n; i++)
{
    Console.Write(vetorResultado[i] + " ");
}
Console.WriteLine();
}
}

```

8)

```
using System;
```

```
class Program
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        Console.WriteLine("Digite uma sequência de caracteres: ");
        string entrada = Console.ReadLine();
```

```
        char[] vetor = entrada.ToCharArray();
```

```
        Console.WriteLine("Quantidade de elementos no vetor: " + vetor.Length);
```

```

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

```

9)
using System;

class Program

```

{
    static void Main()
    {

        Console.WriteLine("Digite o número de elementos do vetor: ");
        int n = int.Parse(Console.ReadLine());

        int[] vetor = new int[n];

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

        Console.WriteLine("Elementos do vetor:");
        foreach (int numero in vetor)
        {
            Console.Write(numero + " ");
        }
        Console.WriteLine();

        Console.WriteLine("Digite o valor a ser procurado no vetor: ");
        int valorProcurado = int.Parse(Console.ReadLine());

        int contador = 0;
        foreach (int numero in vetor)
        {
            if (numero == valorProcurado)

```



```

    {
        contador++;
    }
}

```

```

        Console.WriteLine($"O valor {valorProcurado} aparece {contador} vez(es) no vetor.");
    }
}

```

10)
using System;

class Program

```

{
    static void Main()
    {

```

```

        Console.Write("Digite o número de lançamentos do dado: ");
        int n = int.Parse(Console.ReadLine());

```

```

        int[] resultados = new int[n];

```

```

        Random rand = new Random();

```

```

        Console.WriteLine("Resultados dos lançamentos:");
        for (int i = 0; i < n; i++)
        {
            resultados[i] = rand.Next(1, 7); // Gera números entre 1 e 6
            Console.Write(resultados[i] + " ");
        }
        Console.WriteLine();

```

```

        int[] contagemFaces = new int[6];

```

```

        foreach (int resultado in resultados)
        {
            if (resultado >= 1 && resultado <= 6)
            {
                contagemFaces[resultado - 1]++; // Incrementa a contagem da face
correspondente
            }
        }
    }
}

```

```
Console.WriteLine("Ocorrências de cada face do dado:");
for (int i = 0; i < contagemFaces.Length; i++)
{
    Console.WriteLine($"Face {i + 1}: {contagemFaces[i]} ocorrência(s)");
}
}
```