

# Exercícios com Vetores e Matrizes em C#

## Exercícios 1 a 10 – Vetores (com resolução)

### Exercício 1 – Armazenar e Exibir Notas com Situação

**Enunciado:** Peça ao usuário para informar 5 notas de um aluno. Exiba todas as notas com a mensagem “Aprovado” se a nota for maior ou igual a 7, ou “Reprovado” caso contrário.

```
int[] notas = new int[5];

for (int i = 0; i < notas.Length; i++)
{
    Console.Write($"Digite a nota {i + 1}: ");
    notas[i] = int.Parse(Console.ReadLine());
}

for (int i = 0; i < notas.Length; i++)
{
    string situacao = notas[i] >= 7 ? "Aprovado" : "Reprovado";
    Console.WriteLine($"Nota {i + 1}: {notas[i]} - {situacao}");
}
```

### Exercício 2 – Identificar o Maior Valor

**Enunciado:** Leia 10 números inteiros e mostre qual é o maior valor entre eles.

```
int[] numeros = new int[10];
int maior = int.MinValue;

for (int i = 0; i < numeros.Length; i++)
{
    Console.Write($"Digite o número {i + 1}: ");
    numeros[i] = int.Parse(Console.ReadLine());

    if (numeros[i] > maior)
        maior = numeros[i];
}

Console.WriteLine($"Maior número digitado: {maior}");
```

### Exercício 3 – Contar Pares e Ímpares

**Enunciado:** Leia 8 números e mostre quantos são pares e quantos são ímpares.

```
int[] valores = new int[8];
int pares = 0, impares = 0;

for (int i = 0; i < valores.Length; i++)
{
    Console.Write($"Número {i + 1}: ");
    valores[i] = int.Parse(Console.ReadLine());
}
```

```

        if (valores[i] % 2 == 0)
            pares++;
        else
            impares++;
    }

    Console.WriteLine($"Pares: {pares} | Ímpares: {impares}");

```

#### Exercício 4 – Cálculo de Média de Notas

**Enunciado:** Solicite ao usuário as notas de 4 provas. Calcule a média e diga se o aluno está aprovado (média  $\geq 6$ ).

```

double[] notas = new double[4];
double soma = 0;

for (int i = 0; i < notas.Length; i++)
{
    Console.Write($"Nota {i + 1}: ");
    notas[i] = double.Parse(Console.ReadLine());
    soma += notas[i];
}

double media = soma / notas.Length;
string status = media >= 6 ? "Aprovado" : "Reprovado";
Console.WriteLine($"Média: {media:F1} - {status}");

```

#### Exercício 5 – Contar Valores Positivos

**Enunciado:** Peça 6 números ao usuário e informe quantos são positivos.

```

int[] numeros = new int[6];
int positivos = 0;

for (int i = 0; i < numeros.Length; i++)
{
    Console.Write($"Número {i + 1}: ");
    numeros[i] = int.Parse(Console.ReadLine());

    if (numeros[i] > 0)
        positivos++;
}

Console.WriteLine($"Quantidade de números positivos: {positivos}");

```

#### Exercício 6 – Verificar se há algum número negativo

```

int[] entradas = new int[8];
bool temNegativo = false;

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

```

```

{
    Console.Write($"Digite o número {i + 1}: ");
    entradas[i] = int.Parse(Console.ReadLine());

    if (entradas[i] < 0)
        temNegativo = true;
}

Console.WriteLine(temNegativo ? "Há número negativo." : "Nenhum número negativo foi digitado.");

```

### Exercício 7 – Buscar um valor no vetor

```

int[] dados = new int[10];

for (int i = 0; i < dados.Length; i++)
{
    Console.Write($"Número {i + 1}: ");
    dados[i] = int.Parse(Console.ReadLine());
}

Console.Write("Digite um número para buscar: ");
int buscado = int.Parse(Console.ReadLine());
bool encontrado = false;

foreach (int numero in dados)
{
    if (numero == buscado)
    {
        encontrado = true;
        break;
    }
}

Console.WriteLine(encontrado ? "Número encontrado." : "Número não encontrado.");

```

### Exercício 8 – Contar valores acima da média

```

double[] valores = new double[5];
double soma = 0;

for (int i = 0; i < valores.Length; i++)
{
    Console.Write($"Digite o valor {i + 1}: ");
    valores[i] = double.Parse(Console.ReadLine());
    soma += valores[i];
}

double media = soma / valores.Length;
int acima = 0;
foreach (double v in valores)
{

```

```

        if (v > media)
            acima++;
    }
    Console.WriteLine($"{acima} valores estão acima da média de {media:F2}.");

```

### Exercício 9 – Verificar se vetor está ordenado

```

int[] ordem = new int[6];
bool crescente = true;

for (int i = 0; i < ordem.Length; i++)
{
    Console.Write($"Número {i + 1}: ");
    ordem[i] = int.Parse(Console.ReadLine());
    if (i > 0 && ordem[i] < ordem[i - 1])
        crescente = false;
}
Console.WriteLine(crescente ? "Ordem crescente." : "Não está em ordem crescente.");

```

### Exercício 10 – Exibir em ordem inversa

```

int[] numeros = new int[7];
for (int i = 0; i < numeros.Length; i++)
{
    Console.Write($"Número {i + 1}: ");
    numeros[i] = int.Parse(Console.ReadLine());
}
Console.WriteLine("Ordem inversa:");
for (int i = numeros.Length - 1; i >= 0; i--)
{
    Console.WriteLine(numeros[i]);
}

```

## ▮ Exercícios 11 a 20 – Matrizes (com resolução)

[... conteúdo omitido aqui para brevidade, será incluído na próxima mensagem caso deseje ...]

## ▮ Exercícios 21 a 30 – Enunciados com Dicas (sem código)

### Exercício 21 – Quantidade de valores acima de um limite

**Enunciado:** Solicite ao usuário que preencha um vetor de 10 posições com números reais. Em seguida, peça um valor de referência e informe quantos elementos do vetor são maiores que ele. **Dica:** use um contador dentro de um laço for.

### Exercício 22 – Vetor de aprovados

**Enunciado:** Peça 8 notas (0 a 10) para preencher um vetor. Depois, mostre apenas as notas que são maiores ou iguais a 6. **Dica:** use uma estrutura condicional dentro do loop para filtrar os valores.

### Exercício 23 – Classificação de temperaturas

**Enunciado:** Leia 5 temperaturas e classifique cada uma como "fria" (<18), "agradável" (18-25) ou "quente" (>25). **Dica:** use `if-else` com três condições distintas.

### Exercício 24 – Trocar valores negativos por zero

**Enunciado:** Preencha um vetor com 10 números. Substitua todos os valores negativos por zero e exiba o vetor resultante. **Dica:** percorra o vetor e use atribuição condicional.

### Exercício 25 – Soma dos elementos da diagonal principal

**Enunciado:** Crie uma matriz 4x4 e solicite os valores. Em seguida, calcule e exiba a soma da diagonal principal. **Dica:** use um único laço `for` onde os índices de linha e coluna são iguais.

### Exercício 26 – Quantidade de pares na matriz

**Enunciado:** Preencha uma matriz 3x3 com números inteiros. Conte quantos deles são pares. **Dica:** use `foreach` ou laços aninhados com verificação de paridade.

### Exercício 27 – Verificar simetria de matriz

**Enunciado:** Crie uma matriz 3x3 e verifique se ela é simétrica em relação à diagonal principal. **Dica:** compare se `matriz[i,j] == matriz[j,i]` para todas as posições.

### Exercício 28 – Identificar menor valor da matriz

**Enunciado:** Solicite os elementos de uma matriz 4x4 e mostre qual é o menor valor armazenado. **Dica:** use uma variável auxiliar inicializada com `int.MaxValue`.

### Exercício 29 – Valores acima da média da matriz

**Enunciado:** Peça os valores de uma matriz 3x3, calcule a média de todos os elementos e mostre apenas os que estão acima dessa média. **Dica:** primeiro percorra para somar, depois para comparar.

### Exercício 30 – Procurar número e exibir posição

**Enunciado:** Crie uma matriz 3x3 com valores inteiros fornecidos pelo usuário. Peça um número e informe se ele existe na matriz, e em caso afirmativo, mostre a posição (linha e coluna). **Dica:** use dois laços `for` e uma comparação com o valor buscado.