
	<b>MINISTÉRIO DA EDUCAÇÃO</b> <b>UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI</b> <b>CAMPUS SENADOR HELVÍDIO NUNES DE BARROS – PICOS</b>		
<b>Curso: Sistemas de Informação</b>		<b>Período: 2º</b>	<b>Ano/Semestre: 2025.1</b>
<b>Disciplina: Algoritmos e Programação II</b>			<b>Professor: José Denes Lima Araújo</b>

#### 4º ATIVIDADE – FUNÇÕES

1. Crie uma função que recebe um número **N** (dimensão da matriz  $N \times N$ ) fornecido pelo usuário. Em seguida, crie uma matriz  $N \times N$  preenchida inicialmente com zeros. Depois, peça ao usuário para preencher a matriz com uns (1s) de maneira que forme um padrão. O padrão pode ser um triângulo ou um quadrado, conforme o preenchimento do usuário. Por fim, implemente uma função que recebe essa matriz e verifica se a forma desenhada é um **quadrado** ou um **triângulo**. Caso o padrão não seja nenhum desses, a função deve exibir uma mensagem informando que a forma não é nem um quadrado nem um triângulo.

Esta questão envolve a criação de **três funções principais**:

1. Uma função para **criar a matriz preenchida com zeros**.
2. Uma função para **preencher a matriz com 1s** conforme o padrão escolhido pelo usuário.
3. Uma função para **analisar a matriz e identificar a forma** desenhada.

**Exemplo de entrada e saída:**

**Entrada:**

Digite o valor de N (tamanho da matriz  $N \times N$ ): 5

Digite os valores da matriz linha por linha (0 ou 1):

0 1 1 1 0

0 1 1 1 0

0 1 1 1 0

0 0 0 0 0

0 0 0 0 0

**Saída:**

A forma é um quadrado.

**Entrada:**

Digite o valor de N (tamanho da matriz NxN): 5

Digite os valores da matriz linha por linha (0 ou 1):

0 0 1 0 0

0 1 1 1 0

1 1 1 1 1

0 0 0 0 0

0 0 0 0 0

**Saída:**

A forma é um triângulo.

2. Faça uma função que receba um vetor de inteiros de tamanho N, tal que N foi definido pelo usuário, preenchido pelo usuário e mostre:

- O vetor ordenado em ordem crescente;
- A media dos valores do vetor;
- Maior e menor valor do vetor.

O programa deve ser implementado usando 4 funções diferentes, sendo elas:

```
void ordenarVetor(int vetor[], int n);
```

```
float calcularMedia(int vetor[], int n);
```

```
int encontrarMaior(int vetor[], int n, int maior);
```

```
int encontrarMenor(int vetor[], int n, int menor);
```

**Exemplo de entrada e saída:****Entrada:**

Tamanho do vetor: 7

Vetor: 1 9 5 3 6 9 3

**Saída:**

Vetor ordenado: 1 3 3 5 6 9 9

Média: 5,14

Maior: 9

Menor: 1

**3.** Faça uma função que receba um inteiro  $N$ , tal que,  $N > 2$  e  $N$  foi definido pelo usuário e retorne:

A soma dos números primos no intervalo  $(0, N)$ ;

Em seguida, faça uma função do tipo inteiro que recebe a soma dos números primos no intervalo e retorne se a soma é um número par ou ímpar.

**Exemplo de entrada e saída:**

**Entrada:**

Número: 10

**Saída:**

Soma dos números primos no intervalo: 17

17 é ímpar.

**4.** Dado um inteiro  $N$ , tal que  $N$  foi definido pelo usuário, peça ao usuário para preencher um vetor de tamanho  $N$  APENAS com 0s e 1s. Depois, faça uma função que receba o vetor e mostre todas as permutações possíveis entre os 0s e 1s do vetor.

**Exemplo de entrada e saída:**

**Entrada:**

Tamanho do vetor: 3

Vetor: 1 0 1

**Saída:**

0 0 0

0 0 1

0 1 0

0 1 1

1 0 0

1 0 1

1 1 0

1 1 1

5. Crie uma função em linguagem C chamada **buscar\_primeira\_ocorrencia** que receba três parâmetros:

- Um vetor de números inteiros.
- O tamanho do vetor.
- Um número inteiro a ser buscado.

A função deve retornar o **índice da primeira ocorrência** do valor no vetor. Se o valor não for encontrado, a função deve retornar -1.

**Exemplo:**

**Entrada:**

numero\_buscado = 2

vetor = {1, 2, 2, 3, 4, 5}

**Saída:**

Índice da primeira ocorrência: 1

6. Crie uma função em linguagem C chamada **calcular\_expressao** que receba dois números inteiros positivos **n** e **k** e calcule o valor da seguinte expressão:

$$\text{Resultado} = \frac{n!}{k^n}$$

**Onde:**

- **n!** é o fatorial de n.
- **k<sup>n</sup>** é a potência de k elevado a n.

**Requisitos:**

- Não utilize funções prontas.
- Crie outras duas funções:
  - **fatorial(int n)** que retorne o fatorial de n.
  - **potencia(int k, int n)** que retorne a potência de k elevado a n.

**Exemplo:**

**Entrada:** n = 4; k = 2;

**Saída:** Resultado = 1,5

7. Escreva uma função que recebe 3 valores reais X, Y e Z e verifique se esses valores podem ser os comprimentos dos lados de um triângulo e, neste caso, exibir o tipo de triângulo formado. Para que X, Y e Z formem um triângulo é necessário que a seguinte propriedade seja satisfeita:

- O comprimento de cada lado de um triângulo é menor do que a soma do comprimento dos outros dois lados.

O procedimento deve identificar o tipo de triângulo formado observando as seguintes definições:

- **Triângulo Equilátero:** os comprimentos dos 3 lados são iguais.
- **Triângulo Isósceles:** os comprimentos de 2 lados são iguais.
- **Triângulo Escaleno:** os comprimentos dos 3 lados são diferentes.

**Exemplos:**

**Entrada:** X=5, Y=5, Z=5.

**Saída:** Triangulo Equilátero.

**Entrada:** X = 1, Y = 2, Z = 10.

**Saída:** Não formam um triângulo.

8. Crie uma função que some os elementos de duas matrizes de inteiros, de forma que cada elemento de uma linha da primeira matriz seja somado com o elemento de uma linha correspondente da segunda matriz, considerando que as linhas da segunda matriz devem ser percorridas em ordem invertida em relação à primeira.

**Especificações:**

- O programa deve solicitar ao usuário o número de linhas e colunas das matrizes.
- As duas matrizes devem ser inicializadas com os tamanhos digitados e preenchidas com valores fornecidos pelo usuário.
- A função responsável pela soma pode armazenar o resultado em uma matriz global, não sendo necessário retornar valores.

**Após a operação, o programa deve imprimir:**

- A primeira matriz,
- A segunda matriz,
- A matriz resultante da soma.

**OBS:** a imagem mostra como é percorrido as matrizes a cada loop. Basta colocar o resultado na mesma sequência da matriz X.

**1:**

x	0	1	2
0			
1			
2			

y	0	1	2
0			
1			
2			

**2:**

x	0	1	2
0			
1			
2			

y	0	1	2
0			
1			
2			

**3:**

x	0	1	2
0			
1			
2			

y	0	1	2
0			
1			
2			

**4:**

x	0	1	2
0			
1			
2			

y	0	1	2
0			
1			
2			

Assim por diante...

**Exemplo:**

**Entrada:**

**Linha = 3**

**Coluna = 3**

**Matriz1[Linha][Coluna] = {{1,2,3},  
{4,5,6},  
{7,8,9}}**

**Matriz2[Linha][Coluna] = {{1,2,3},  
{4,5,6},  
{7,8,9}}**

**Saída:**

**Matriz 1:**

1	2	3
---	---	---

4	5	6
---	---	---

7	8	9
---	---	---

**Matriz 2:**

1	2	3
---	---	---

4	5	6
---	---	---

7	8	9
---	---	---

**Matriz resultado:**

4	4	4
---	---	---

10	10	10
----	----	----

16	16	16
----	----	----