

Desafio Técnico - Processo Seletivo IN Júnior

Exercício 1

Escreva um programa em que o usuário insira os coeficientes a, b e c de uma equação quadrática da forma:

$$ax^2 + bx + c = 0$$

O programa deve:

1. Calcular o discriminante (Δ) pela fórmula:
$$\Delta = b^2 - 4ac$$
2. Avaliar o valor de Δ para determinar o número de raízes reais:
 - Se $\Delta > 0$, exibir as duas raízes reais e distintas.
 - Se $\Delta = 0$, exibir uma raiz real (raiz dupla).
 - Se $\Delta < 0$, informar que a equação não possui raízes reais.
3. Usar a fórmula de Bhaskara para calcular as raízes (caso existam) e imprimir na tela tais raízes:

$$x1 = \frac{-b + \sqrt{\Delta}}{2a} \quad x2 = \frac{-b - \sqrt{\Delta}}{2a}$$

Exercício 2

Escreva um programa que:

1. Solicite ao usuário um número inteiro positivo e calcule o seu fatorial.
2. Pergunte ao usuário se ele quer calcular o fatorial de outro número. Se o usuário digitar "S" ou "s", o programa deve reiniciar e permitir outro cálculo; se digitar "N" ou "n", o programa deve encerrar.
3. Valide a entrada do usuário para garantir que apenas números inteiros positivos são aceitos. Se o usuário digitar um número inválido, exiba uma mensagem de erro e peça a entrada novamente.

Exercício 3

Crie uma função chamada **verificarIdades** que recebe um array de anos de nascimento de várias pessoas. A função deve calcular a idade de cada pessoa com base no ano atual (inserido pelo usuário) e, para

cada idade, determinar se a pessoa é maior de idade (18 anos ou mais) ou menor de idade (menos de 18 anos). A função deve retornar um array contendo "maior" para pessoas maiores de idade e "menor" para pessoas menores de idade, mantendo a ordem das idades verificadas. Após isso, a função deve imprimir na tela uma mensagem para cada pessoa, no formato:

Pessoa 1: Maior de idade
Pessoa 2: Menor de idade
(...)

Exercício 4

Elabore um algoritmo para ler um valor monetário em reais (R\$) e convertê-lo para as moedas euro (€) e dólar americano (US\$), considerando as seguintes taxas de câmbio:

- Taxa de câmbio para euro: €1 = R\$6,10
- Taxa de câmbio para dólar: \$1 = R\$5,70

Imprima na tela o valor em reais inserido pelo usuário e suas conversões para dólar americano e euro.

Exercício 5

Elabore um algoritmo que peça para o usuário inserir um número inteiro e realize o seguinte:

- Caso o valor seja divisível por 3, imprima "fizz".
- Caso o valor seja divisível por 5 imprima "buzz".
- Caso o valor seja divisível por 3 e 5, ao mesmo tempo, imprima "fizzbuzz".
- Caso contrário imprima nada.

Exercício 6

Escreva um programa que:

1. Solicite ao usuário um número inteiro positivo n , que representa o número de termos que ele deseja ver da sequência de Fibonacci.
2. Implemente uma função recursiva chamada fibonacci que recebe um número n e retorna os n primeiros termos da sequência de Fibonacci, onde:
 - $\text{Fibonacci}(0) = 0$

- $\text{Fibonacci}(1) = 1$
 - Para $n > 1$: $\text{Fibonacci}(n) = \text{Fibonacci}(n-1) + \text{Fibonacci}(n-2)$
3. Exiba os n primeiros termos da sequência de Fibonacci.

Exercício 7

Escreva um programa que verifique se uma frase é um palíndromo. Um palíndromo é uma sequência que pode ser lida da mesma maneira da esquerda para a direita e da direita para a esquerda.

Requisitos:

1. Solicite ao usuário uma frase.
2. Remova todos os espaços e ignore a diferença entre letras maiúsculas e minúsculas.
3. Verifique se a frase é igual quando lida ao contrário e informe se ela é um palíndromo ou não.

Dica: Use `toLowerCase()` e `split()`.

Considere o seguinte array de objetos para os exercícios 8 e 9:

```
const booksByCategory = [
  {
    category: "Romance",
    books: [
      {
        title: "O Sol Também é Uma Estrela",
        author: "Nicola Yoon",
      },
      {
        title: "O Noivo da Minha Melhor Amiga",
        author: "Emily Giffin",
      },
      {
        title: "Orgulho e Preconceito",
        author: "Jane Austen",
      },
    ],
  },
  {
    category: "Fantasia",
    books: [
      {
        title: "Harry Potter e a Pedra Filosofal",
        author: "J.K. Rowling",
      },
    ],
  },
]
```

```

        {
            title: "O Hobbit",
            author: "J.R.R. Tolkien",
        },
        {
            title: "Eragon",
            author: "Christopher Paolini",
        },
        {
            title: "O Nome do Vento",
            author: "Patrick Rothfuss",
        },
    ],
},
{
    category: "Suspense",
    books: [
        {
            title: "A Garota no Trem",
            author: "Paula Hawkins",
        },
        {
            title: "O Silêncio dos Inocentes",
            author: "Thomas Harris",
        },
    ],
},
{
    category: "Ficção Científica",
    books: [
        {
            title: "Duna",
            author: "Frank Herbert",
        },
        {
            title: "Neuromancer",
            author: "William Gibson",
        },
        {
            title: "Fundação",
            author: "Isaac Asimov",
        },
        {
            title: "1984",
            author: "George Orwell",
        },
        {
            title: "O Fim da Infância",
            author: "Arthur C. Clarke",
        },
    ],
},
];

```

Exercício 8

Implemente uma função chamada **countBooksInCategory** que conte o número de livros em cada categoria e retorne um objeto com o nome da categoria como chave e a contagem de livros como valor.

Exercício 9

Escreva uma função chamada **authors** que retorne um array com os nomes de todos os autores presentes no objeto **booksByCategory**, desconsiderando autores repetidos.

Exercício 10

Escreva um programa para simular uma fila de espera em uma central de atendimento ao cliente. O programa deve iniciar exibindo na tela um menu interativo contendo a lista de todos os clientes esperando atendimento, mostrando a posição ao lado do nome (ex.: 1º João, 2º Maria, etc).

O menu deve permitir escolher entre as seguintes opções:

- 1 - “Novo Cliente”: Adiciona um novo cliente ao final da fila (solicitando o nome do cliente).
- 2 - “Atender Cliente”: Retira o primeiro cliente da fila e exibe o nome do cliente que está sendo atendido.
- 3 - “Sair”: Encerra o programa.

O programa só deve ser encerrado ao escolher a opção 3. Caso contrário, ele deve retornar ao menu principal.

Exercício 11

Escreva um programa que receba uma lista de números inteiros e armazene-os em um array. O programa deve verificar quantos números do vetor estão em ordem crescente em relação ao número anterior e exibir essa informação.

Requisitos:

1. Solicite ao usuário a quantidade de números que ele deseja inserir no vetor e, em seguida, peça que insira cada número.
2. Verifique cada par consecutivo de números no vetor e identifique se estão em ordem crescente (ou seja, se o próximo número é maior que o anterior).
3. Imprima quantos números estão em ordem crescente em relação ao número anterior.

Exemplo: [6,1,3,2,1,7]

Impressão no console: 2

Exercício 12

Implemente uma função chamada **encontrarElementoUnico** que recebe um array de números inteiros onde cada elemento aparece exatamente duas vezes, exceto por um elemento que aparece exatamente uma vez. A função deve retornar o elemento que aparece apenas uma vez. Tal elemento deve ser impresso na tela.

Exercício 13

Você está desenvolvendo um sistema para gerenciar jogadores de um time de esportes. Cada jogador deve ter as seguintes propriedades:

- Nome (string)
- Idade (número)
- Posição (string)
- Pontuação (número)

Implemente as seguintes funcionalidades:

1. Crie um array vazio chamado **time** para armazenar os jogadores cadastrados.
2. Escreva uma função chamada **adicionarJogador** que:
 - Recebe como parâmetros os dados de um jogador (nome, idade, posição e pontuação). Observação: esses dados devem ser inseridos pelo usuário.
 - Cria um objeto representando o jogador e o adiciona ao array **time**.
3. Escreva uma função chamada **buscarPorPosicao** que:
 - Recebe como parâmetro uma string representando a posição. Observação: esse dado deve ser inserido pelo usuário.

- Retorna todos os jogadores dessa posição. Se não houver jogadores, exiba uma mensagem apropriada.
- 4. Escreva uma função chamada `listarTime` que:
 - Imprime todos os jogadores cadastrados no time no console, um por linha.
- 5. Escreva uma função chamada `calcularPontuacaoMedia` que:
 - Calcula e retorna a pontuação média de todos os jogadores do time.
- 6. O programa deve iniciar exibindo o seguinte menu interativo, que deve permitir ao usuário escolher uma das opções:
 - 1 - Adicionar jogador
 - 2 - Buscar por posição
 - 3 - Listar time
 - 4 - Calcular pontuação média
 - 5 - Sair

O programa só deve ser encerrado ao escolher a opção 5. Caso contrário, ele deve retornar ao menu interativo.