

Trabalho 5 - Revisão Geral

TDS 23-2M - Prof. Esp. Iuri Nascimento Santos

Questão 1: Operações Básicas

1.1. Crie uma função chamada `soma` que recebe dois números como parâmetros e retorna a soma deles.

1.2. Crie uma função chamada `multiplicacao` que recebe dois números como parâmetros e retorna o resultado da multiplicação.

1.3. Chame ambas as funções com valores de exemplo e exiba os resultados no console.

Questão 2: Manipulação de Strings

2.1. Crie uma função chamada `inverterString` que recebe uma string como parâmetro e retorna a string invertida.

2.2. Crie uma função chamada `contarVogais` que recebe uma string como parâmetro e retorna o número de vogais na string.

2.3. Chame ambas as funções com strings de exemplo e exiba os resultados no console.

Questão 3: Objetos e Métodos

3.1. Crie um objeto chamado `pessoa` com propriedades como `nome`, `idade` e `profissao`.

3.2. Adicione um método ao objeto `pessoa` chamado `apresentar`, que exiba no console uma mensagem com as informações da pessoa.

3.3. Crie pelo menos duas instâncias do objeto `pessoa` com dados diferentes e chame o método `apresentar` para cada uma.

Questão 4: Arrays e Loops

4.1. Crie um array chamado `numeros` com pelo menos 5 números inteiros.

4.2. Crie uma função chamada `somarArray` que recebe um array de números como parâmetro e retorna a soma de todos os elementos.

4.3. Utilize um loop para percorrer o array `numeros` e exiba no console a soma total.

Questão 5: Projeto Final - Agenda de Contatos

5.1. Crie uma função chamada `criarContato` que recebe informações como nome, telefone e email, e retorna um objeto representando um contato.

5.2. Crie um array chamado `agenda` que irá armazenar diferentes contatos.

5.3. Implemente uma função chamada `adicionarContato` que recebe um contato como parâmetro e o adiciona ao array `agenda`.

5.4. Implemente uma função chamada `listarContatos` que percorre o array `agenda` e exibe no console as informações de cada contato.

Observações:

- Cada questão deve ser implementada em uma seção separada do código.
- Utilize boas práticas de programação, como comentários explicativos e nomes de variáveis significativos.
- Teste suas funções com exemplos para garantir que estão funcionando corretamente.
- Organize seu código de forma estruturada e legível.