

1º Período – Conceitos de algoritmos
Professor Gilberto Viana de Oliveira
Prática sobre funções

Todas as atividades devem ser realizadas utilizando o conceito de modularização. As funções devem, na medida do possível, serem para entrada/saída ou processamento. Utilize os conceitos de retorne, parâmetros e referencia caso queira trabalhar com variáveis que necessitem de alterações foram das suas funções original. Não utilize variáveis globais.

1. Crie uma função que receba um número real e verifique se ele é um número positivo, negativo ou zero. Crie a função `verifiqueNumero` e retorne o valor para a função `inicio`. Na função `inicio`, a partir do valor retornado, imprima o que aquele número é.
2. Crie uma função que receba três valores e retorne o maior valor.
3. Faça uma função que receba como parâmetro o numero de vezes que o usuário deseja rolar um dado. A função deve fazer a rolagem do dado e retornar a quantidade de vezes que o dado gerou o valor 6.
4. Variação da questão 3. Faça uma função que receba como parâmetro o número de vezes para rolar o dado e as faces desse dado. Ao final, imprima na função `inicio` quantas vezes CADA número foi sorteado.
5. Crie uma função chamada `soma(a, b)` que receba dois números e retorne o resultado. Em seguida, no programa principal, peça para o usuário digitar dois valores e exiba a soma.
6. Crie uma função `ehPar(n)` que retorne *verdadeiro* se o número for par e *falso* se for ímpar.
No programa principal, leia um número e imprima “Par” ou “Ímpar” chamando a função.
7. Escreva uma função `celsiusParaFahrenheit(c)` que receba uma temperatura em Celsius e retorne o valor equivalente em Fahrenheit. Depois, leia um valor e exiba o resultado.
8. Crie uma função `fatorial(n)` que calcule o fatorial de um número usando um laço. No programa (`inicio`), leia um número inteiro e mostre seu fatorial fazendo a chamada a função.
9. Crie uma função chamada `somarArray(valores)` que receba um array de números como parâmetro e retorne a soma de todos os elementos.

No programa principal:

1. Leia 5 números do usuário
 2. Armazene-os em um array
 3. Chame a função `somarArray`
 4. Exiba o resultado
10. Crie uma função chamada `filtrarMaiores(lista, limite)` que receba:
- um array de números

- um número limite

A função deve retornar um novo array contendo apenas os elementos do array original que sejam maiores que o limite. Números melhores devem ser substituídos por -1.

No programa principal:

1. Leia 6 números do usuário e armazene em um array
2. Leia um valor para o limite
3. Chame `filtrarMaiores(lista, limite)`
4. Mostre o array filtrado ao usuário

11. Crie duas funções:

- `mediaSimples(lista)` → retorna a média aritmética.
- `mediaPonderada(valores, pesos)` → retorna a média ponderada.

No programa principal:

- leia 3 valores
- leia 3 pesos
- exiba a média simples e a média ponderada.