

Lista de Exercícios – Recursividade em C

Questão 1

Implemente uma função recursiva para calcular o fatorial de um número inteiro positivo e utilize essa função para calcular o número de combinações, conforme a fórmula:

$$C(n, k) = \frac{n!}{k! \cdot (n - k)!}$$

Exemplo: Entrada: $n = 5$, $k = 2$ Saída esperada: 10

Questão 2

Implemente uma função recursiva que receba um número inteiro n e retorne a soma dos n primeiros números pares positivos.

Exemplo: Entrada: $n = 4$ Saída esperada: 20

Questão 3

Escreva uma função recursiva que receba um número inteiro positivo e retorne a soma de seus dígitos.

Exemplo: Entrada: 1234 Saída esperada: 10

Questão 4

Implemente uma função recursiva que calcule a^b , onde a e b são inteiros e $b \geq 0$.

Exemplo: Entrada: $a = 2$, $b = 4$ Saída esperada: 16

Questão 5

Escreva uma função recursiva que verifique se uma palavra é um palíndromo (ou seja, se pode ser lida da mesma forma da esquerda para a direita e vice-versa).

Exemplo: Entrada: "radar" Saída esperada: "é palíndromo"

Questão 6

Implemente duas funções recursivas em C chamadas `verificaPar(int n)` e `verificaImpar(int n)` que utilizem **recursividade indireta** para verificar se os dígitos de um número inteiro positivo alternam entre par e ímpar, da direita para a esquerda.

- A verificação começa pelo último dígito (direita).
- Se o dígito for par, o próximo deve ser ímpar, e vice-versa.
- As funções devem se chamar alternadamente até o número acabar (ou até a alternância ser quebrada).

Exemplos:

- Entrada: 3816 → Saída: Válido
- Entrada: 3826 → Saída: Inválido