

Lista de Exercícios

Recursividade

Estruturas de Dados — Bacharelado em Sistemas de Informação — Instituto Federal da Bahia

Instrução: Todos os exercícios devem ser respondidos em **Linguagem C**.

Exercício 1. Escreva uma **função recursiva** que receba um inteiro positivo N e calcule o **somatório** de todos os inteiros no intervalo $[1, N]$:

$$S(N) = 1 + 2 + 3 + \dots + N$$

Exercício 2. Escreva uma **função recursiva** que receba dois inteiros positivos k e n e calcule k^n usando **apenas multiplicações**. O programa principal deve solicitar k e n ao usuário e exibir o resultado.

$$k^n = \underbrace{k \times k \times \dots \times k}_{n \text{ vezes}}$$

Exercício 3. Crie um programa que receba um **vetor de números reais** com até 100 elementos. Implemente uma **função recursiva** que inverta a ordem dos elementos do vetor **in-place** (sem usar um vetor auxiliar).

Exercício 4. O **Máximo Divisor Comum** (MDC) de dois inteiros x e y é o maior inteiro que divide ambos. Implemente uma **função recursiva** `mdc(x, y)` baseada no Algoritmo de Euclides:

$$\text{mdc}(x, y) = \begin{cases} x & \text{se } y = 0 \\ \text{mdc}(y, x \bmod y) & \text{caso contrário} \end{cases}$$

Exercício 5. A **multiplicação** de dois inteiros pode ser realizada por somas sucessivas. Proponha uma **função recursiva** `Mult(n1, n2)` que calcule $n_1 \times n_2$ usando apenas somas:

$$\text{Mult}(n_1, n_2) = \begin{cases} 0 & \text{se } n_2 = 0 \\ n_1 + \text{Mult}(n_1, n_2 - 1) & \text{caso contrário} \end{cases}$$

Lembre-se: toda função recursiva deve ter um **caso base** bem definido para evitar loop infinito.