

**IFCE**  
**Fundamentos de Programação**  
Lista de Exercícios 06 – Funções Recursivas

**Professor:** Thiago Alves

**Nome:**

- 1) Crie uma função recursiva `multiploTres(n)` que recebe um natural positivo como parâmetro e retorna o n-ésimo natural múltiplo de 3.
- 2) Faça um função recursiva `somaNaturais(n)` que recebe um natural positivo como parâmetro e retorna a soma dos n primeiros naturais.
- 3) O fatorial duplo de um número natural n é o produto de todos os números de 1 (ou 2) até n, contados de 2 em 2. Por exemplo, o fatorial duplo de 8 é  $8 \times 6 \times 4 \times 2 = 384$ , e o fatorial duplo de 7 é  $7 \times 5 \times 3 \times 1 = 105$ . Defina uma função `fatDuplo(n)` para retornar o fatorial duplo do parâmetro n usando recursividade.
- 4) Defina uma função recursiva `multTodos(lista, tam, x)` que tem uma lista de números, o tamanho da lista e um número como parâmetros, respectivamente. A função deve multiplicar todos os elementos da lista por x e retornar a nova lista.
- 5) Defina uma função recursiva `somaQuaseTodos(lista, tam, x, i)` que tem uma lista de números, o tamanho da lista, um número e uma posição da lista como parâmetros, respectivamente. A função deve somar todos os elementos da lista por x, com exceção do elemento de índice i, e retornar a nova lista.
- 6) Defina uma função recursiva `somaExceto(lista, tam x, y)` que tem uma lista de números, o tamanho da lista, um número x e um número y como parâmetros, respectivamente. A função deve somar todos os elementos da lista por x, com exceção do elemento y, e retornar a nova lista.
- 7) Apresente uma função recursiva `somaLista(lista, tam)` que tem como parâmetros uma lista de números e o tamanho da lista, respectivamente. A função deve retornar a soma dos elementos da lista. A soma dos elementos de uma lista vazia é zero.
- 8) Usando recursão, defina a função `produtoListas(lista1, lista2, tam)` que tem como parâmetros duas listas de números de mesmo tamanho e o tamanho da duas listas, respectivamente. A função deve multiplicar os elementos correspondentes das duas listas e retornar a nova lista. Apresente um programa em que o usuário digita os preços de cada tipo de produto de uma loja. A quantidade de tipos de produtos deve ser definida pelo usuário. Depois, o usuário deve digitar a quantidade de produtos vendidos de cada tipo. Ao final da execução, seu programa deve mostrar a quantidade total de produtos vendidos e o valor total arrecadado com as vendas.
- 9) Defina a função `inverteLista(lista, tam)` recursivamente. A função tem uma lista e o tamanho da lista como parâmetros e deve retornar uma nova lista com os elementos da original mas na ordem invertida. Por exemplo, `inverteLista([1, 2, 3])` deve retornar a lista `[3, 2, 1]`.
- 10) Apresente uma definição recursiva para a função `iguais(lista1, lista2, tam)` que tem como parâmetros duas listas de mesmo tamanho e o tamanho da duas listas, respectivamente. A

função deve retornar `True` se as duas listas são iguais. Caso contrário, a função deve retornar `False`. Por exemplo, `iguais([1,2,5], [1,2,5], 3)` deve retornar `True`.

11) Defina recursivamente a função `palindromo(lista, tam)` que tem como parâmetros uma lista e o tamanho da lista. A função retorna `True` se `lista` é um palíndromo e retorna `False`, caso contrário. Uma lista é um palíndromo se é lida da mesma maneira da esquerda para a direita e da direita para a esquerda. Por exemplo, `[1,2,2,3,2,2,1]` é um palíndromo.

12) Apresente uma definição recursiva `concatena(lista1, tam1, lista2, tam2)`. A função deve retornar a concatenação das listas `lista1` e `lista2`. Por exemplo, `concatena([1,2], 2, [4,5,6], 3)` deve retornar a lista `[1,2,4,5,6]`.