

Recursividade - Prática

Marco A L Barbosa

malbarbo.pro.br

- 1) Projete uma função recursiva que determine se algum dos elementos de uma lista encadeada é ímpar.
- 2) Projete uma função recursiva que receba como entrada uma lista encadeada e crie uma nova lista encadeada com os elementos positivos da lista de entrada.
- 3) Projete uma função recursiva que receba como entrada uma lista encadeada de strings e um número natural n , e modifique as strings da lista para que todas fiquem com tamanho n . Se um string tem tamanho maior que n , os caracteres do final devem ser descartados. Se uma string tem tamanho menor que n , espaços em branco devem ser adicionados ao final da string.
- 4) Projete uma função recursiva que encontre o valor máximo de uma lista encadeada. Se a lista for vazia, a função deve devolver `None`.
- 5) Projete uma função recursiva que receba como entrada um número natural n e calcule o produto dos números $1, 2, \dots, n$.
- 6) Projete uma função recursiva que receba como entrada uma string e um número natural n e devolva a string repetida n vezes. Por exemplo, para a string `'casa'` e $n = 3$, a função deve produzir `'casacasacasa'`. Não use o operador de repetição de string (`*`)!
- 7) Projete uma função recursiva que receba como entrada dois números naturais, n e x , onde $x \leq n$ e devolva um arranjo com os divisores de n que são menores ou iguais a x . Por exemplo, para $n = 12$ e $x = 5$, a função deve produzir `[1, 2, 3, 4]`.
- 8) Projete uma função recursiva que receba como entrada um número a (diferente de 0) e um número natural n e calcule o valor a^n .
- 9) Projete uma função recursiva que conte quantas vezes um número aparece em um arranjo.
- 10) Projete uma função recursiva que concatene todos as strings de um arranjo de strings.
- 11) Projete uma função recursiva que encontre o tamanho máximo entre todas as strings de um arranjo de strings.
- 12) Projete uma função recursiva que determine se um valor está em um arranjo ordenado. Implemente a função usando busca binária.
- 13) (Desafio) Recursão indireta é quando duas ou mais funções chamam uma a outra. Projete uma função `par`, que verifique se um número natural é par, e uma função `ímpar`, que verifique se um número natural é ímpar, que são implementadas uma em termos da outra.
- 14) (Desafio) Projete uma função recursiva que verifique se um arranjo de números está em ordem não decrescente. Dica: use dois casos base.
- 15) (Desafio) Dados duas listas encadeadas `lsta` e `lstb`, projete uma função recursiva que verifique se `lsta` é prefixo de `lstb`, isto é `lstb` começa com `lsta`.