

# Estruturas de dados

## Recursividade

Marco A L Barbosa

[malbarbo.pro.br](http://malbarbo.pro.br)

### Introdução

- 1) O que é uma função recursiva?
- 2) Qual é a relação entre decomposição de problema e recursividade?
- 3) Qual é a relação entre tipos com autorreferência e funções recursivas?
- 4) O que é recursão estrutural?

### Começando

- 5) Projete uma função recursiva que determine se algum dos elementos de uma lista encadeada é ímpar.
- 6) Projete uma função recursiva que encontre o valor máximo de uma lista encadeada. Se a lista for vazia, a função deve devolver `None`.
- 7) 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 (\*).
- 8) Projete uma função recursiva que conte quantas vezes um número aparece em um arranjo.

### Praticando

- 9) 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.
- 10) Projete uma função recursiva que receba como entrada um número  $a$  (diferente de 0) e um número natural  $n$  e calcule o valor  $a^n$ .
- 11) 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.
- 12) Projete uma função recursiva que encontre o tamanho máximo entre todas as strings de um arranjo de strings.

### Avançando

- 13) Recursão indireta é quando duas ou mais funções chamam uma a outra. Defina duas funções `impar` e `par`, uma em termos da outra, isto é, a função `impar` deve chamar a função `par` e a função `par` deve chamar a função `impar` (a recursão para no caso base).
- 14) Projete uma função recursiva que verifique se um arranjo de números está em ordem não decrescente. Dica: use dois casos base.
- 15) Projete uma função recursiva que receba como entrada uma lista encadeada de números e remova todas as ocorrências no valor 0.

- 16) 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  $n = 16$  e  $x = 7$ , a função deve produzir  $[1, 2, 4]$ .

### Desafios

- 17) Dados duas listas encadeadas  $lsta$  e  $lstb$ , projete uma função recursiva que verifique se  $lsta$  é prefixo de  $lstb$ , isto é  $lstb$  começa com  $lsta$ .
- 18) Projete uma função recursiva que altere um arranjo de números ordenando em ordem não decrescente os seus elementos.