

## Programação funcional

### Lista 2 - manipulação de Listas

A lista a seguir contém exercícios focados em funções e na manipulação de listas.

1. Gere os primeiros  $n$  números ímpares a partir de 1.

Exemplo:

`primeirosImpares 6`  $\rightarrow$  `[1,3,5,7,9,11]`

`primeirosImpares 1`  $\rightarrow$  `[1]`

2. Crie uma sequência infinita de um caractere  $c$  e devolva os primeiros  $n$  caracteres.

Exemplo:

`repeteChar 6 'x'`  $\rightarrow$  `"xxxxxx"`

`repeteChar 3 'A'`  $\rightarrow$  `"AAA"`

3. Defina a função *oDoMeio3* que, dados 3 inteiros, devolve o inteiro do meio (ou seja, a mediana).

Exemplo:

`oDoMeio3 4 2 7`  $=$  `4`

`oDoMeio3 2 (-1) 10`  $=$  `2`

4. Defina a função *quantosAcimaDaMedia* que, dados 3 inteiros, retorna o número de inteiros acima da média.

Exemplo:

`quantosAcimaDaMedia 1 2 3`  $=$  `1`

`quantosAcimaDaMedia 4 9 12`  $=$  `2`

5. Defina a função *numero2* que, dados dois caracteres, retorna um inteiro em que o algarismo das dezenas é o primeiro carácter e o das unidades é o segundo, ou retorna zero se algum dos caracteres não for um dígito.

Exemplo:

`numero2 '2' '3'`  $=$  `23`

`numero2 '4' '1'`  $=$  `41`

6. Defina a função *anoBissexto* que, dado um ano (um inteiro), devolve `True` se o ano é bissexto, `False` se não. (Nota: um ano bissexto é divisível por 400 ou é divisível por 4 mas não é divisível por 100)

7. Crie uma função que divida uma lista ao meio e devolva as duas partes (como duas listas dentro de uma lista). Caso o número de elementos seja ímpar, a primeira parte deve ter um a mais.

Exemplo:

metade [1,2,3,4] → [[1,2],[3,4]]

metade [10,20,30,40,50] → [[10,20,30],[40,50]]

8. Crie uma função que elimine o primeiro e o último elemento de uma lista. Se tiver menos de dois elementos, devolva uma lista vazia.

Exemplo:

semExtremos [1,2,3,4,5] → [2,3,4]

semExtremos [9] → []

9. Dadas duas extremidades, produza a lista com todos os valores entre elas, incluindo as extremidades.

Exemplo:

intervalo 5 9 → [5,6,7,8,9]

intervalo 9 5 → [9,8,7,6,5]

10. Dada uma lista, divida-a ao meio e inverta a ordem das duas metades. Se o número de elementos for ímpar, o elemento central fica na segunda metade.

Exemplo:

inverteMetades [1,2,3,4] → [3,4,1,2]

inverteMetades [10,20,30,40,50] → [30,40,50,10,20]

11. Defina uma função que insere um elemento em uma posição dada, deslocando os demais à direita. Se a posição for maior que o tamanho, insira no final.

Exemplo:

inserePos 2 99 [10,20,30,40] → [10,20,99,30,40]

inserePos 10 5 [1,2,3] → [1,2,3,5]

12. Defina uma função que dado um inteiro troca o algarismo das unidades com o das dezenas.

Exemplo:

trocaPosicao 123 = 132

13. Escreva, em Haskell, a função `invertInt::Int->Int` que inverta os dígitos de um número inteiro.

Exemplo: `invertInt 123 = 321`.

14. Crie uma função que determine se um número é palíndromo (igual quando invertido).

Exemplo:

palindromo 121  $\rightarrow$  True  
palindromo 123  $\rightarrow$  False  
palindromo 1001  $\rightarrow$  True

15. Defina a função dezenas que, dado um inteiro, retorna o dígito das dezenas.

Exemplo:

dezenas 1234  $\rightarrow$  3

dezenas 507  $\rightarrow$  0

dezenas 9  $\rightarrow$  0

16. Defina uma função que, dado um inteiro, troque o algarismo das unidades com o das dezenas.

Exemplo:

swapUD 1234  $\rightarrow$  1243

swapUD 90  $\rightarrow$  09

swapUD 7  $\rightarrow$  7