

Bacharelado em Sistemas de Informação

Programação Funcional

Lista de Revisão - Linguagem Haskell

1. Defina uma função que, dada uma lista numérica, retorna uma tupla-2, tal que contenha o maior valor da lista, bem como sua posição relativa.
2. Remover um elemento da lista a partir de sua posição
3. Seja uma lista de tuplas cujo conteúdo é o nome e a idade de pessoas. Implemente funções que retornem:
 - (a) O nome da pessoa mais jovem
 - (b) A quantidade de pessoas com idade acima de 30
 - (c) A lista de pessoas com idade impar
4. Defina funções para implementar a união e interseção entre duas listas.
5. Construa uma função que verifica se uma lista numérica é um palíndromo.

Soluções:

Exercicio 1

```
-- Retorna a posição relativa de um dado elemento

posv::Int->[Int]->Int
posv x [] = 0
posv x (a:t) | x == a = 1
              | otherwise = 1 + posv x t

-- Retorna o maior elemento de uma lista

maior::[Int]->Int
maior[a] = a
maior(a:x) = if a > maior x then a
              else maior x

-- Cria uma tupla-2 com o maior elemento e sua posição

criatupla::[Int]->(Int,Int)
criatupla (a:x) = (maior(a:x), posv (maior (a:x)) (a:x))
```

Exercicio 2

```
-- Remove uma posição da lista

remove::Int->[Int]->[Int]
remove 1 (a:x) = x
remove n (a:x) = a: remove (n-1) x
```

Exercicio 3

```
type Pessoa = ([Char],Int)
type Nome = [Char]
type Cadastro = [Pessoa]
type Pessoas = [Nome]

lista::Cadastro
lista=[("Ana",40),("Joao",56),("Carla",7),("Paula",80)]

maisjovem::Cadastro->[Char]
maisjovem [(a,i)]=a
maisjovem ((a,x):((b,y):t)) | x < y      = maisjovem ((a,x):t)
                             | otherwise   = maisjovem ((b,y):t)

maioresTrinta::Cadastro->Int
maioresTrinta [] = 0
maioresTrinta ((_,b):t) | b > 30      = 1 + maioresTrinta t
                         | otherwise   = maioresTrinta t

listaImpar::Cadastro->Pessoas
listaImpar []=[]
listaImpar ((a,x):t) | mod x 2 /= 0 = a:listaImpar t
                     | otherwise   = listaImpar t
```

Exercicio 4

Exercicio 5