

Exercícios de Programação Funcional em Haskell

Entrega:

Coloque os exercícios num repositório versionado, num arquivo denominado t3.hs

Informe a URL do repositório em <http://bit.do/entrega-paradigmas>

Prazo: quarta-feira, 26/08, 23:55

1. Copie a função abaixo para um arquivo .hs e carregue-o no interpretador Haskell. Depois teste essa função e descubra o que faz a função mod. Entregue esse exercício em forma de comentário.

```
isEven :: Int -> Bool
isEven n = mod n 2 == 0
```

2. Crie uma função `somaquad :: Int -> Int -> Int` que calcule a soma dos quadrados de dois números x e y.
3. Crie uma função `doubleFirst :: [Int] -> Int` que calcule o quadrado do primeiro elemento de uma lista.
4. Crie uma função `hasEqHeads :: [Int] -> [Int] -> Bool` que verifique se 2 listas possuem o mesmo primeiro elemento.
5. Escreva uma função que receba uma lista de nomes e adicione a string "Mr. " no início de cada nome.
6. Crie uma função que conte quantos espaços existem numa dada string.
7. Escreva uma função que, dada uma lista de números, calcule $3*n^2 + 2/n + 1$ para cada número n da lista.
8. Escreva uma função que, dada uma lista de idades de pessoas no ano atual, retorne uma lista somente com as idades de quem nasceu depois de 1970.
9. Escreva uma função `serie :: Double -> [Double] -> Double` que receba um número m e uma lista de n números, e calcule o resultado da seguinte série:

$$\frac{x_1}{m} + \frac{x_2}{m} + \frac{x_3}{m} + \dots + \frac{x_n}{m}$$

10. Crie uma função `charFound :: Char -> String -> Bool` que verifique se o caracter (primeiro argumento) está contido na string (segundo argumento).

Exemplos de uso da função:

```
> charFound 'a' ""
False
> charFound 'a' "uau"
True
```

11. Crie uma função `htmlListItems :: [String] -> [String]`, que receba uma lista de strings e retorne outra lista contendo as strings formatadas como itens de lista em HTML.

Exemplo de uso da função:

```
> htmlListItems ["abra", "ca", "dabra"]
["<LI>abra</LI>", "<LI>ca</LI>", "<LI>dabra</LI>"]
```

12. A função `takeWhile :: (a -> Bool) -> [a] -> [a]` é uma função de alta ordem. Ela recebe uma função condicional e uma lista, retornando o "menor prefixo" (= porção inicial) da lista que satisfaça a condição.

Teste os exemplos abaixo no GHCi e depois crie um novo exemplo:

```
> takeWhile (< 5) [1,2,3,4,5]
> takeWhile (/=' ') "Fulana de Tal"
```

Obs.: O exercício 12 deve ser entregue em forma de comentário.

13. Seu amigo Lázaro trabalha com organização de eventos e tem uma longa lista de participantes de uma feira. Os organizadores da feira querem usar uma estratégia de marketing com as participantes do sexo feminino, mas infelizmente a lista só tem os nomes e os dados para contato, sem informação do sexo. Como marketing não é uma ciência exata, já seria suficiente identificar uma boa parte dos nomes femininos, mesmo cometendo alguns erros. Para ajudar o Lázaro rapidamente, você tem a ideia de fazer um programa para identificar os participantes cujo primeiro nome termina com 'a', pois há grande probabilidade de que sejam do sexo feminino. Resolva isso em Haskell, com uma função que receba a lista do Lázaro e retorne a lista