

## EXERCÍCIOS EM HASKELL

### Lista 4

1) Dada uma tupla do tipo (String, String, Char), escreva três funções diferentes para retornar cada elemento da tupla.

2) Dada a base abaixo e as funções criadas para tuplas na questão 1

```
base :: Int -> (String, String, Char)
base x
  | x == 1 = ("joao", "mestre", 'm')
  | x == 2 = ("jonas", "doutor", 'm')
  | x == 3 = ("joice", "mestre", 'f')
  | x == 4 = ("janete", "doutor", 'f')
  | x == 5 = ("jocileide", "doutor", 'f')
  | otherwise = ("ninguem", 'x')
```

- a) Escreva uma função recursiva *contMestre* que retorne o número de mestres.
- b) Escreva uma função recursiva *contDoc* que retorne o número de doutores.
- c) Transforme as funções desenvolvidas nos itens a) e b) em uma única função *contMD*, passando a titulação como parâmetro.
- d) Crie uma nova função *cont* adicionando a função desenvolvida no item c) o gênero como parâmetro (f: feminino ou m: masculino).

3) Dada a base abaixo

```
base :: Int -> (String, String, Char)
base x
  | x == 1 = ("joao", "mestre", 'm')
  | x == 2 = ("jonas", "doutor", 'm')
  | x == 3 = ("joice", "mestre", 'f')
  | x == 4 = ("janete", "doutor", 'f')
  | x == 5 = ("jocileide", "doutor", 'f')
  | otherwise = ("ninguem", "", 'x')
```

- a) Crie um apelido para o tipo de elementos da tupla (nome, titulo, gênero), para a tupla (pesquisador) e para a lista de pesquisadores (grupo). Para isso, utilize a palavra-chave *type*.
- b) Crie uma função recursiva para formar uma lista de pesquisadores com os elementos da função base.
- c) Escreva uma função que receba uma lista de pesquisadores e retorne outra lista apenas com o nome de todos os pesquisadores que possuem o título de doutor. Utilize o gerador de lista.

4) Crie um módulo para armazenar as funções de tratamento abaixo:

- a) Uma função que receba um nome e adicione no início o termo “Sr.”
- b) Uma função que receba um nome e adicione no início o termo “Sra.”
- c) Uma função que receba um nome e adicione no início o termo “Srta.”
- d) Uma função recursiva de alta ordem (do tipo mapeamento) que receba uma lista de nomes e uma das funções definidas em (a), (b) ou (c). Essa função deve retornar uma lista de nome com a adição do termo de tratamento no início.

Em seguida, importe o módulo que você criou no módulo Main e imprima a função de alta ordem definida na letra (d) passando como argumento as funções definidas nas letras (a), (b) e (c). Dica: defina listas de nomes para adequadas para cada caso.

5) Crie um módulo para armazenar as funções abaixo:

- a) Uma função que receba um valor real e verifique se é positivo.
- b) Uma função que receba um valor real e verifique se é negativo.
- c) Uma função que receba um valor real e verifique se é igual a zero.
- d) Uma função recursiva de alta ordem (do tipo filtro) que receba uma lista de valores e permita selecionar os elementos que satisfazem as condições (a), (b) ou (c).

Em seguida, importe o módulo que você criou no módulo Main e imprima a função de alta ordem definida na letra (d) passando como argumento as funções definidas nas letras (a), (b) e (c). Dica: crie uma lista formada por números positivos, negativos e nulos.