



Linguagens de Programação - Turma B - Lista Haskell II

Prof. Marcos Fagundes Caetano - 2016/1

Matrícula	Nome

1. Fazendo o uso de bibliotecas, defina a função *metade* $:: [a] \rightarrow ([a], [a])$. Esta função divide uma lista de tamanho par em duas partes.

```
> metade [1,2,3,4,5,6]
([1,2,3],[4,5,6])
```

Obs: Fica a sua decisão o que fazer caso a lista tenha tamanho ímpar.

2. Implemente a função **retornaUltimo**, tal que: *retornaUltimo* $:: [a] \rightarrow a$. Esta função retorna o último elemento de uma lista. Caso a lista fornecida seja vazia, a função apresenta a seguinte mensagem de erro: **Esta lista está vazia!**. A função deve ter o seguinte comportamento:

```
> retornaUltimo []
Program error: Esta lista está vazia!
> retornaUltimo [1]
1
> retornaUltimo [1,2,3,4,5]
5
> retornaUltimo ['a','v','b']
'b'
```

3. Implemente a função **penúltimo**, tal que: *penultimo* $:: [a] \rightarrow a$. Esta função retorna o penúltimo elemento de uma lista. A função deve ter o seguinte comportamento:

```
> penultimo []
Program error: Esta lista está vazia!
> penultimo [1]
Program error: Lista só tem um elemento!
> penultimo [1,2,3,4]
3
```

4. Implemente a função **tamanho**, tal que: *tamanho* $:: [a] \rightarrow Int$. **Não utilize a função length**. Dica: esta função pode ser definida em função dela mesma. A função apresenta o seguinte comportamento.

```
> tamanho [1,2,3,4]
4
```

5. Implemente a função **reverte**, tal que: $reverte :: [a] \rightarrow [a]$. A função reverte apresenta o seguinte comportamento:

```
> reverte [1,2,3,4]
[4,3,2,1]
> reverte "A man, a plan, a canal, panama!"
"!amanap ,lanac a ,nalp a ,nam A"
```

6. Implemente a função **duplicar**, tal que: $duplicar :: [a] \rightarrow [a]$. A função duplicar apresenta o seguinte comportamento:

```
> duplicar [1,2,3,4]
[1,1,2,2,3,3,4,4]
> duplicar []
[]
```

7. Implemente a função **replicar**, tal que: $replicar :: [a] \rightarrow Int \rightarrow [a]$. **Dica:** verifique o comportamento das funções: **map**, **concatMap** e **replicate**.

```
> replicar [1,2,3] 4
[1,1,1,1,2,2,2,2,3,3,3,3]
> replicar [] 4
[]
```

8. Implemente a função **fatia**, tal que: $fatia :: Int \rightarrow Int \rightarrow [a] \rightarrow [a]$. A função fatia apresenta o seguinte comportamento:

```
> fatia 2 4 ['a','b','c','d','e','f']
"bcd"
> fatia 2 4 [1,2,3,4,5,6]
[2,3,4]
> fatia 2 6 [1,2,3,4,5,6]
[2,3,4,5,6]
> fatia 2 7 [1,2,3,4,5,6]
[2,3,4,5,6]
> fatia 0 7 [1,2,3,4,5,6]
[1,2,3,4,5,6]
```

Obs: A função **fatia** retornar uma faixa da lista definida pelos índices i e j , representados no exemplo acima por $i = 2$ e $j = 4$, por exemplo. Assuma que o usuário sempre irá informar posições corretas e válidas para a lista informada.