

Tópico: Classes e Polimorfismo

Dica: Resolva todos os exercícios sem utilizar o computador. Uma vez que os exercícios estejam prontos, utilize o GHCi para conferir suas respostas.

1. Quais os tipos das seguintes funções? Inclua as devidas restrições de classe, se necessário.

segundo xs = head (tail xs)

trocar (x, y) = (y, x)

parear x y = (x, y)

dobro x = x * 2

palindromo xs = reverse xs == xs

2. Escreva uma função que forneça o terceiro elemento de uma tupla-3. Declare o tipo mais geral para essa função.

3. Fornecidos três valores, a, b e c, implemente uma função que retorne quantos desses três são iguais. A resposta deve ser 3, se todos são iguais; 2, se dois são iguais e um é distinto dos demais ou 0, se todos são distintos entre si. Declare o tipo mais geral para essa função.

4. Escreva uma função que receba uma tupla-3 e retorne uma tupla-2 com o maior e o menor elemento dentre os três. Declare o tipo mais geral para essa função.

5. Implemente as seguintes funções recursivas em Haskell definindo o tipo mais geral de cada função.

(a) Determinar o comprimento de uma lista. Exemplo de uso:

```
> comprimento [3,14,1,5,9]
```

```
5
```

(b) Determinar o somatório dos elementos de uma lista. Exemplo de uso:

```
> somatorio [3,14,1,5,9]
```

```
32
```

(c) Determinar o somatório dos elementos ímpares de uma lista. Exemplo de uso:

```
> somatorio_impares [3,14,1,5,9]
```

```
18
```

(d) Determinar a soma dos elementos de uma lista que são múltiplos de 3. Exemplo de uso:

```
> soma_mult_3 [3,14,1,5,9]
```

```
12
```

(e) Determinar o produto dos elementos de uma lista. Exemplo de uso:

```
> produtorio [3,14,1,5,9]
```

```
1890
```

(f) Determinar o n-ésimo elemento de uma lista. Exemplo de uso:

```
> n_esimo 3 [3,14,1,5,9]
```

```
5
```

Obs.: Considere que o primeiro elemento da lista está na posição 1.

(g) Determinar o último elemento de uma lista. Exemplo de uso:

```
> ultimo [3,14,1,5,9]
```

```
9
```

```
> substituir_todos 1 2 [3,14,1,5,1]
[3,14,2,5,2]
```

```
> maior [3,1,4,1,5,9]
```

(j) Escreva uma função que retorna verdadeiro se todos os elementos de uma lista de inteiros forem ímpares, ou falso, caso contrário. Exemplo de uso:

True

```
> insert 4 [1,3,5]
```

Se o elemento já existir, pode inseri-lo depois de sua ocorrência.

```
> pertence 5 [3,14,1,5,9]
```

(m) Dada uma lista de tuplas-2, criar uma nova lista com apenas os primeiros elementos de cada tupla. Exemplo de uso:

 $[3, 1, 9]$

```
> concatenar [[3,14,1,5],[9,1,2]]
```

[3,14,1,5,9,1,2]

[illegible]