Exercícios de Programação Haskell I

prof. André Rauber Du Bois

Universidade Federal de Pelotas
http://minerva.ufpel.edu.br/~dubois/
dubois@ufpel.edu.br

1 Questionário

1. Escreva a função `osQuatroSaoIguais` que possui tipo

   ```
   Int -> Int -> Int -> Int -> Bool
   ```

   que retorna `True` se seus quatro argumentos são iguais
2. Defina a função `quantosSaoIguais :: Int -> Int -> Int -> Int` que conta quantos argumentos iguais a função recebeu
3. Defina a função

   ```
   todosDiferentes :: Int -> Int -> Int -> Bool
   ```

   que retorna `True` se todos os seus argumentos são diferentes. Obs: `m /= n` retorna `True` se `m` e `n` são diferentes
4. Defina um conjunto de testes para a função `todosDiferentes`
5. O que está errado com a seguinte definição de `todosDiferentes`:

   ```
   todosDiferentes n m p = ( ( n/=m ) && ( m/=p ) )
   ```

   O conjunto de testes que você definiu na questão anterior funciona com esta definição?
6. Defina a funcao

   ```
   todosIguais :: Int -> Int -> Int -> Bool
   ```

7. Escreva uma definição de `quantosSaoIguais` que use a função `todosDiferentes` e a função `todosIguais`
8. Defina a função `elevadoDois :: Int -> Int` que recebe um argumento `n` e devolve como resposta n^2
9. Defina a função `elevadoQuatro :: Int -> Int` que recebe um argumento `n` e devolve como resposta n^4. Use `elevadoDois` para definir `elevadoQuatro`
10. Supondo que exista uma função `vendas`:

    ```
    vendas :: Int -> Int
    ```

que devolve a venda semanal de uma loja (ex: `vendas 0` devolve as vendas na semana 0, `vendas 1` devolve as vendas na semana 1, etc. Implemente uma função chamada `vendaTotal`, que recebe um argumento `n` e calcula todas as vendas da semana `0` até a semana `n`. Observe que essa função deve ser recursiva. Exemplo de calculo: As vendas da semana 0 até a semana 2, podem ser calculados usando a seguinte formula:

```
vendas 0 + vendas 1 + vendas 2
```