

IF686 - 2020.3
Lista de exercícios
Funções

1. Definir a função `dobro` de tipo `Integer -> Integer`. A função deve receber um argumento e devolvê-lo multiplicado por dois.
2. Definir a função `quadruplo` que utiliza a função `dobro` do exercício anterior para devolver o seu argumento multiplicado por quatro
3. Definir a função `poli2`. Devem ser necessários quatro argumentos do tipo `Double` (`a`, `b`, `c`, e `x`) e devolver $a * x^2 + b * x + c$. Definir a assinatura do tipo `poli2` , ou seja, `poli2 :: alguma coisa`
4. Definir a função `parImpar` que devolve "par" (string) para entradas pares e "impar" (string) para entradas ímpares. Defina uma função para determinar se um número é par.
5. Defina a função `maxFour :: Integer -> Integer -> Integer -> Integer -> Integer` que retorna o máximo de quatro inteiros. Dê três definições dessa função: a primeira descrita com base em `maxThree`; a segunda deve usar a função `max` e a terceira deve usar as funções `max` e `maxThree`.
6. Defina a função `quantosIguais :: Integer -> Integer -> Integer -> Integer` que retorna quantos dos três argumentos são iguais. Por exemplo,
`quantosIguais 56 32 12 = 0`
`quantosIguais 12 12 43 = 2`
7. Usando casamento de padrão, definir a função `ehZero` que retorna verdadeiro se for dado como argumento um inteiro que seja 0, e falso, caso contrário. Definir o tipo da função `ehZero`
8. Usando recursão, implemente a função `sumTo`, de modo que `sumTo n` calcula o valor de $1 + 2 + \dots + n$
9. Defina a função `potencia`, de modo que `potencia n k` calcula `n` elevado a `k`. Use recursão.
10. Usando recursão, compute os coeficientes binomiais dados pelas seguintes equações
$$B(n, k) = B(n - 1, k) + B(n - 1, k - 1)$$
$$B(n, 0) = 1$$
$$B(0, k) = 0, \text{ quando } k > 0$$

Dica: usar casamento de padrão pode ser de grande ajuda.
11. Os números de Tribonacci são dados pelas seguintes equações
$$T(1) = 1$$
$$T(2) = 1$$

$$T(3) = 2$$

$$T(n+1) = T(n) + T(n-1) + T(n-2)$$

Implemente uma função recursiva eficiente que calcula $T\ n$. Considere o uso de uma função auxiliar.

12. Defina a função `addEspacos` que produz um string com uma quantidade `n` de espaços.

`addEspacos :: Int -> String`

13. Defina a função `paraDireita` utilizando a definição de `addEspacos` para adicionar uma quantidade `n` de espaços à esquerda de um dado String, movendo o mesmo para a direita.

`paraDireita :: Int -> String -> String`

14. Escreva uma função para retornar, em forma de tabela, todas as vendas da semana 0 até a semana `n`, incluindo o total e a média de vendas no período. Usem as funções definidas previamente e defina novas funções que achar necessário.

Semana	Venda
0	12
1	14
2	15
Total	41
Média	13.6667

`imprimeTabela :: Int -> String`

```
imprimeTabela n = cabecalho
                ++ imprimeSemanas n
                ++ imprimeTotal n
                ++ imprimeMedia n
```