

Exercícios sobre Funções de Alta Ordem

prof. André Rauber Du Bois

Universidade Federal de Pelotas
dubois@inf.ufpel.edu.br

1 Questionário

1. Defina a função `aplicaDuasVezes` que recebe uma função e um valor e aplica essa função duas vezes no valor de entrada. Exemplo: supondo a existência de uma função `incrementa` que adiciona 1 a um inteiro:

```
*Main> aplicaDuasVezes incrementa 10
12
```

outroExemplo:

```
dobra :: Int -> Int
dobra x = x*x
```

```
*Main> aplicaDuasVezes dobra 10
10000
```

2. Defina novamente a função `vendaTotal`, que calcula o total de vendas de várias semanas. Agora a função `vendaTotal` deve receber como argumento a função que determina as vendas de uma semana:

```
vendaTotal :: (Int -> Int) -> Int -> Int
```

3. Definina a função `foldInt`

```
foldInt :: (Int -> Int -> Int) -> [Int] -> Int
(...)
```

```
soma :: Int -> Int -> Int
soma x y = x + y
```

```
mult :: Int -> Int -> Int
mult x y = x * y
```

```
*Main> foldInt soma [1,2,3]
```

```
*Main> foldInt soma []
erro

*Main> foldInt soma [1]
1

*Main> foldInt mult [2,3,4]
24
```

4. Defina a função

```
filterString :: (Char -> Bool) -> [Char] -> [Char]
```

```
naoEspaco :: Char -> Bool
naoEspaco x = x /= ' '
```

```
Hugs> filterString naoEspaco "Andre Du Bois "
"AndreDuBois"
```

5. Usando as funções `mapInt` e `foldInt`, defina: a função (`somaQuadrado :: [Int] -> Int`) que devolva a soma do quadrado dos números em uma lista
1
6. Defina a função `iter`, onde

```
iter 1 f x = f x
iter 2 f x = f (f x)
iter 3 f x = f (f (f x))
(...)
```