IF686 - 2020.3
Lista de exercícios
Map, Filter, Fold

1. Defina a função length usando map e sum

2. Usando a função map e funções da biblioteca, defina as seguintes funções
   (a) uppers :: String -> String que transforma um string para maiúsculas
   (b) doubles :: [Int] -> [Int] que dobra cada elemento uma lista
   (c) centavosReais :: [Int] -> [Float] converte preços em centavos para reais

3. Usando a função filter e funções da biblioteca, defina as seguintes funções
   (a) letras :: String -> String que remove todos os caracteres não alfabéticos do string dado como argumento
   (b) rmChar :: Char -> String -> String que remove do string todas as ocorrências do caractere dado como primeiro argumento
   (c) acima :: Int -> [Int] -> [Int] que remove da lista todos os números menores ou iguais ao número dado como primeiro argumento
   (d) desiguais :: [(Int, Int)] -> [(Int,Int)]que remove todos os pares (x,y) tais que x == y

4. Compreensão de listas processam uma lista de forma similar a map e filter . De forma geral, [f x | x <- xs, p x] é equivalente a map f (filter p xs). Escreva expressões equivalentes a seguir, usando map e filter
   (a) [toUpper c | c <- s, isAlpha c]
   (b) [2 *x | x <- xs, x > 3]
   (c) [reverse s | s <- strs, even (length s)]

5. Escreva funções usando recursão e também foldr
   (a) A função recursiva productRec :: [Int] -> Int que computa o produto de todos os elementos de uma lista. Escreva a equivalente productFold, usando a função foldr
   (b) A função recursiva andRec :: [Bool] -> Bool que verifica se todos os elementos de uma lista são verdadeiros. Escreva a equivalente andFold, usando foldr
   (c) A função recursiva concatRec :: [String] -> String que concatena todas os elementos de uma lista de strings em um único string. Escreva a função equivalente concatFold, usando foldr