Universidade Federal de Uberlândia - UFU
Faculdade de Computação - FACOM
Lista de exercícios de Programação Funcional

Ordem superior - Mapear

1. Seja a função map implementada no módulo Prelude. Primeiramente vamos observar o uso da função map, que define a aplicação de uma outra função (que tenha a propriedade de transformação) a todos os elementos de uma lista. Execute os exemplos abaixo:

   > map (++"ana") ["mari", "juli", "lili"]
   ["mariana","juliana","liliana"]

   > map (^3) [4,5,3,6]
   [64,125,27,216]

2. Nos exercícios a seguir, **escreva primeiramente uma função recursiva**. Em seguida, reescreva essa função utilizando a função de ordem superior mapear, vista em aula e apresentada a seguir. **Não utilize a função nativa map.**

   ```
   mapear::(a->a)->[a]->[a]
   mapear f [] = []
   mapear f (a:ax) = f a : mapear f ax
   ```

   (a) Função **dobros :: Num a => [a] -> [a]** que dobra todos os elementos de uma lista.

   (b) Função **primeiros :: [(a,b)] -> [a]** que extrai o primeiro elemento de cada tupla-2 dentro de uma lista. **Dica:** utilize a função fst

   (c) Função **maiusculas :: String -> String** que converte uma string para outra string com letras maiúsculas. **Dica:** utilize a função toUpper da biblioteca Data.Char.

   (d) Função **hora_em_seg :: [Float] -> [Float]** que converte uma lista de horas em uma lista de segundos.

3. Faça a função **ellen**, que considera uma lista de strings e devolve uma lista dos tamanhos de cada string. Utilize ordem superior.

   *Main > ellen [“aula”,”de”,”programacao”]
   [4,2,11]