**Aula Prática 1**

|  |
| --- |
| **Instruções:**  1 - Os exercícios práticos devem ser realizados individualmente e enviados por e-mail com o assunto [IF686EC] AP 1 para **monitoria-if686-ec-l@cin.ufpe.br** até as **23:59 do sábado (24.08.2019)**.  2 - As resoluções dos exercícios devem estar em arquivos diferentes, um arquivo por exercício com os nomes no formato Q[número da questão].hs. Nesse caso, são esperados 5 possíveis arquivos: Q1.hs, Q2.hs, Q3.hs, Q4.hs e Q5.hs.  3 - O arquivo com a resposta de cada questão deve conter a função solicitada no formato dado em negrito no enunciado da questão. Os tipos de entrada e saída explicitados, devem ser respeitados, assim como o nome da função. |

**[Q1]** Faça uma função em Haskell **isPalindromo :: String -> Bool** que dada uma String retorna se a mesma é palíndromo ou não. Considere que não haverá nenhuma letra maiúscula na string.

**isPalindromo** “asd”

False

**isPalindromo** “ana”

True

**isPalindromo** “Ana”

True

**[Q2]** Faça uma função em Haskell **btoi :: String -> Int** que, dada uma string representando um número binário, retorna um inteiro na base 10 dessa string.

**btoi “0011”**

**3**

**btoi “1100”**

**12**

**[Q3]** Faça uma função **metade :: [Int] -> ([Int], [Int])** que separa uma lista de inteiros em duas partes retornando-as em uma tupla. Caso essa string possua tamanho ímpar, a segunda lista da tupla retornada terá um elemento a mais que a primeira.

Exemplo:

**metade [1,2,3,4]**

**([1,2],[3,4])**

**metade [1,2,3]**

**([1],[2,3])**

**[Q4]** A string **"aaaaa"** repete um único caractere, o **'a'**, 5 vezes. Crie uma função **isReplica -> String -> Int -> Char -> Bool** que recebe uma string, um inteiro **x** e um char verifica se essa string é a repetição do char **x** vezes.

**isReplica "ee" 2 'e'**

**True**

**isReplica "uruu" 3 'u'**

**False**

**isReplica "xxx" 3 'y'**

**False**

**[Q5]** Crie uma função **decEnigma :: String -> [(Char, Char)] -> String** que decifra uma string da linguagem A para a linguagem B. Ela recebe a string que precisa ser decifrada e uma lista de tuplas contendo os dois alfabetos. Os primeiros caracteres da tupla, representam o alfabeto de A e, os segundos, de B.

**decEnigma "usr" [('u','j'), ('s','o'), ('r','b')]**

**"job"**

**decEnigma "msyc" [('m','e'), ('s','i'), ('y','t'), ('c','a')]**

**"eita"**

**decEnigma "qloz" [('q','h'), ('l','u'), ('o','g'), ('z','o')]**

**"hugo"**