Aula Prática 3

Instruções:

1 - Os exercícios práticos devem ser realizados individualmente e enviados por e-mail com o assunto [IF686EC] AP 3 para monitoria-if686-ec-l@cin.ufpe.br até as **23:59 de segunda-feira (23.09.2019)**.

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: Q1.hs, Q2.hs e Q3.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, assim como o nome da função, devem ser respeitados.

[Q1] Dados os seguintes tipos algébricos

```
data Ops =
      SUM
    | MUL
    | SUB
    deriving ( Show, Eq )

data IntTree =
     Nilt Int
     | Node Ops IntTree IntTree
     deriving ( Show )
```

Escreva uma função **`evalTree :: IntTree -> Int`** que calcula o valor resultante das operações na árvore dada.

```
Exemplo:
evalTree (Node SUM (Node MUL (Nilt 5) (Nilt 3)) (Node SUB (Nilt 10)
(Nilt 5)))
20
```

[Q2] Dado o seguinte tipo algébrico

```
     data Tree t = Nilt |
                          Node t (Tree t) (Tree t)
```

Escreva uma função **`isBST :: Tree t -> Bool`** que checa se uma árvore é uma árvore de busca binária. Considerar que nenhum dos elementos se repetirá.

Exemplos:

```
isBST (Node 5 (Node 3 Nilt Nilt) (Node 7 Nilt Nilt))
True
isBST (Node 3 (Node 5 Nilt Nilt) (Node 7 Nilt Nilt))
False
```

[Q3] Dado o seguinte tipo algébrico

```
data Comando = ParaFrente Int
             | ParaTras Int
             | Escreva Char
             deriving (Show, Eq)
```

Escreva uma função **`interprete :: String -> [Comando] -> Char`** que, dada uma lista de comandos e uma string, retorne o caractere da posição final da cabeça de leitura, após realizar os comandos nessa string.

Exemplos:

```
interprete "abcdefghijklmno" [ParaFrente 5, Escreva 'x', ParaFrente
1, Escreva 'y', ParaTras 1]
'x'
interprete "abcdefghijklmno" [ParaFrente 5, Escreva 'x', ParaFrente
1, Escreva 'y']
'y'
interprete "abcdefghijklmno" [ParaFrente 5, Escreva 'x', ParaFrente
1]
'g'
interprete "abcdefghijklmno" []
'a'
```