

Bacharelado em Ciência da Computação
Programação Funcional
Trabalho Final - Linguagem Haskell (10 pontos)
Grupos de 2 alunos

1. Defina uma função que, dada uma lista numérica, retorna uma tupla-2, tal que contenha o maior valor da lista, bem como sua posição relativa.

Exemplo:

```
> maior [13,8,15,12,26,4]
(26,5)
```

2. Seja o código abaixo:

```
f1 char n | n <= 0 = putStr "Msg Erro"
          | otherwise = f2 1 n char

f2 x n char | x == n = imprime x char
            | otherwise = do imprime x char
                           putChar '\n'
                           f2 (+ x 1) n char

imprime x char | x <= 0 = putChar ' '
              | otherwise = do putChar char
                             imprime (x-1) char
```

a) Teste o código e explique a finalidade do programa

b) Mostre o traço da execução para a aplicação: `> f1 '#' 6`

3. Seja a definição abaixo para uma estrutura de dados em árvore binária de busca.

```
data ArvBinInt = Nil | NoInt Int ArvBinInt ArvBinInt
               deriving Show

arvDados::ArvBinInt
arvDados = NoInt 7 (NoInt 5 (NoInt 2 Nil Nil) (NoInt 6 Nil Nil))
           (NoInt 13 (NoInt 9 Nil Nil) (NoInt 20 Nil (NoInt 23
Nil Nil)))
```

a) Desenhe a árvore de dados armazenada em arvDados. Em seguida faça funções para:

a1) verificar se uma árvore é vazia

a2) retornar a sub-árvore à esquerda de um nó (raiz da árvore)

a3) inserir ordenadamente um novo elemento (repetições devem ser evitadas)

b) Seja a função abaixo de retorno e remoção do maior elemento. Verifique se está correta e caso necessário faça as devidas correções. Explique a estratégia a ser utilizada na remoção de um elemento qualquer em árvore binária de busca.

```
deletemax :: ArvBinInt -> (Int,ArvBinInt)
deletemax (NoInt y t1 Nil) = (y,t1)
deletemax (NoInt y t1 t2)  = (z, NoInt y t1 tz)
                             where (z,tz) = deletemax t2
```

4. Seja a definição abaixo para o algoritmo de ordenação segundo o método da Bolha. Explique a codificação do algoritmo, e mostre como funciona a classificação dos elemento de lista qualquer.

```
ordBolha [] = []
ordBolha xs =
  iterate troca xs !! (length xs - 1)
  where
    troca [x] = [x]
    troca (x:y:zs)
      | x > y      = y: troca (x:zs)
      | otherwise = x: troca (y:zs)

-- Definicao da funcao Iterate (módulo Prelude)
iterate f x = x: iterate f (f x)

> take 6 (iterate (\x -> (x+3)*2) 1)
[1,8,22,50,106,218]
```