O objetivo deste trabalho é definir uma função que, dado um documento (arquivo .txt), gere um índice de linhas em que as palavras ocorrem nesse documento, as informações (palavras e lista de linhas em que ocorrem) devem ser armazenada em uma árvore binária de pesquisa declarada como:

```
type Word'  = String
type Line   = String
type Doc    = String

data Tree   = Node Word' [Int] Tree Tree | Leaf deriving Show
```

A função makeIndexTree deve combinar todas as funções definidas nos itens abaixo retornando a árvore de índices.

makeIndexTree :: Doc → Tree

O problema de gerar os índices pode ser dividido nos seguintes subproblemas:

a) Separar o documento em linhas: lines :: Doc → [Line]

b) Numerar as linhas do documento: numLines :: [Line] → [(Int,Line)]

c) Associar a cada ocorrência de uma palavra do documento, o número da linha em que essa palavra ocorre: allNumWords :: [(Int,Line)] → [(Int,Word')]

d) Inserir elementos em uma lista ordenada, o elemento deve ser inserido em uma posição que mantenha a lista resultante ordenada. Caso a lista já contenha o elemento não deve ocorrer a inserção:

   insOrd :: a → [a] → [a]

e) Inserir uma palavra e linha de ocorrência na árvore, caso a palavra já tenha sido inserida apenas a linha deve ser adicionada a lista de linhas relacionadas com a palavra. Deve ser usada a função definida no item anterior para essa tarefa:

   ins :: Word' → Int → Tree → Tree

f) Percorrer a lista com as duplas de palavras e linhas inserido cada uma delas na árvore:

   mIndexTree :: [(Int,Word')] → Tree

A impressão das palavras e índices deve ser feita percorrendo a árvore em ordem.

Observações:
As seguintes funções são definidas na biblioteca padrão de *Haskell*:

```
lines :: String → [String] -- Divide um texto em linhas
words :: String → [String] -- Divide uma linha em palavras
```