

Este trabalho pode ser feito em duplas.

O objetivo desse trabalho é definir uma função que, dado um documento, gera um índice das palavras que ocorrem nesse documento. O programa deve ter como entrada um arquivo texto. Considere as seguintes definições:

```
type Doc = String
type Line = String
type Word = String
makeindex :: Doc → [(Int,Word)]
```

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

1. Separar o documento em linhas: `lines :: Doc → [Line]`
2. Numerar as linhas do documento: `numLines :: [Line] → [(Int,Line)]`
3. Associar a cada ocorrência de uma palavra do documento, o número da linha em que essa palavra ocorre: `allNumWords :: [(Int,Line)] → [(Int,Word)]`
4. Ordenar alfabeticamente as ocorrências de palavras no texto:

```
sortLs :: [(Int,Word)] → [(Int,Word)]
```

5. Juntar as várias ocorrências de cada palavra, produzindo, para cada palavra, a lista dos números das linhas em que a palavra ocorre:

```
almalgamate :: [(Int,Word)] → [(Int,Word)]
```

6. Eliminar, da lista de números de páginas em que cada palavra ocorre, as repetições de um mesmo número de linha:

```
shorten :: [(Int,Word)] → [(Int,Word)]
```

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
```

Antes de separar cada linha do seu texto em palavras, devem ser eliminados da linha os caracteres de pontuação, os quais não devem ser incluídos no índice (olhar as funções `isAlpha`, `isSpace`, `isDigit`, `zip`).