

O objetivo deste trabalho é definir uma função que, dado um documento, gera um índice com as linhas em que cada palavra ocorre. O programa deve ter como entrada um arquivo texto e deve imprimir uma palavra e seus índices por linha. 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:

- Separar o documento em linhas: `lines :: Doc → [Line]`
- Numerar as linhas do documento: `numLines :: [Line] → [(Int, Line)]`
- Associar a cada ocorrência de uma palavra do documento, o número da linha em que essa palavra ocorre: `allNumWords :: [(Int, Line)] → [(Int, Word')]`
- Ordenar alfabeticamente as ocorrências de palavras no texto:
`sortLs :: [(Int, Word')] → [(Int, Word')]`
- 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')]`
- Eliminar, da lista de números de linhas 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
```

Devem ser removidas as palavras com três ou menos caracteres.

Exemplo, para um arquivo com o texto:

```
Departamento de Ciencia da Computacao
Curso de Ciencia da Computacao
Programacao Funcional
```

A saída deve ser:

```
Ciencia – [1,2]
Computacao – [1,2]
Curso – [2]
Departamento – [1]
Funcional – [3]
Programacao – [3]
```