

Proj 3

ATE meia noite de 18/9

1) incrementar (função auxiliar)

Implemente um contador. Esse contador pode ser implementado como uma lista de tuplas onde cada tupla é no formado `(chave, contador)`. Ou o contador pode ser implementado usando um dicionario do `Data.Map.Strict`

Implemente a função:

```
incrementar :: Eq a => Conta -> a -> Conta
```

Onde `Conta` é o tipo do seu contador (nao precisa definir esse tipo usando o `data`). A funcao recebe um contador, um item e incrementa no contador a contagem do item, retornando o contador atualizado.

2) letra mais comum

usando o `incrementar` acima, escreva a função

```
letra_mais_comum :: [Char] -> Char
```

que recebe um string e retorna a letra mais comum no string. Use as seguintes regras

- letras sao apenas a..z (sem digitos, sem pontuação e sem brancos)
- letras maiusculas e minusculas são consideradas a mesma letra
- voce precisa usar a função `incrementar` acima

Assim

```
letra_mais_comum "77,88 a!? abc BB 8 8      8  fyt"
```

retorna `b` (ou `B`)

- `b` e `B` sao a mesma letra
- branco e `8` que aparecem mais vezes no string não são considerados letras.

Voce pode utilizar todas as funções do `Data.List` e `Data.Map.Strict`. Relevantes para o problema sao funcoes como `sort` e suas variações ou `maximum` e suas variacoes.

Para usar o `sort`, por exemplo, use

```
import Data.List sort
```

no comeco do seu programa.

Como sempre, voce pode definir quaisquer funções auxiliares que voce quiser.

A correção da 2a parte `letra_mais_comum` nao vai depender se sua implementacao do `incrementar` esta certa ou não.Ou seja, eu vou considerar que o `incrementar` funciona corretamente na correcao da parte 2.