Exercícios – Programação Funcional
(Linguagens de Programação)

01) Escreva uma denição recursiva da função prefixo :: String -> String -> Bool que verica se uma cadeia de caracteres é prefixo de outra.

Exemplos :
prefixo "abra" "abra" = True
prefixo "abra" "abracadabra" = True
prefixo "braca" "abracadabra = False
prefixo "abracadabra" "abra" = False

02) Considere uma função count que conta o número de elementos de uma lista para os quais uma função com retorno Bool passada por parâmetro retorna TRUE.

Exemplos:
> count (>2) [0,1,2,3]
1
> count (/='a') "banana"
3

a) Escreva uma denição recursiva da função count.
b) Escreva uma denição não-recursiva de count.
c) Usando count e a função isLetter :: Char → Bool escreva uma definição da função
extras :: String → Int que conta o número de caracteres em uma cadeia que **NÃO** são letras.

d) Elabore uma função main que recebe como entrada uma função F com retorno Bool (Assumindo uma propriedade) e várias palavras do teclado. O programa PARA quando a função F retorna verdade para uma palavra lida do teclado e retorna uma lista com as palavras recentemente digitadas.

03) Implemente um algoritmo de ordenação com as características apresentadas em sala de aula.

a) Apresente a execução do algoritmo para a lista [0,2,1] e [0,2,1,0,1].
b) Quais modificações podem ser implementadas para que o algoritmo possa repetir as chaves idênticas.

04) Escreva uma definição recursiva da função rle :: String → [(Int, Char)] que comprime uma cadeia usando run-length encoding : cada sequência de caracteres repetidos é representada por um par com o comprimento e o caracter correspondente.

Exemplo: rle "aaabbcbbbb" = [(3,'a'),(2,'b'),(1,'c'),(4,'b')]