

1. Descreva a sequência de passos referente a execução das funções abaixo:
 - a. $fib\ 0 = 0$
 $fib\ 1 = 1$
 $fib\ n = fib\ (n-1) + fib\ (n-2)$

para >> fib 6
 - b. $len\ xs = foldr\ (\lambda x\ y \rightarrow 1 + y)\ 0\ xs$

para >> len [2,8,5,10,28]
 - c. $appFoldl\ f\ e\ xs = foldl\ f\ e\ xs$

para >> appFoldl (++) [] [[28,10,12], [1,4,6], [], [1,9]]
2. Definir uma função que recebe duas strings como parâmetro, e remove da segunda lista as letras que ocorrerem na primeira lista.
Ex: "Programacao" "Funcional" => "Funinl"
3. Definir uma função que recebe uma string, e retorna uma lista de todas as substrings possíveis que formam um prefixo da string.
Ex: "fim" => ["fim", "fi", "f"]
4. Definir uma função que recebe uma lista e um número inteiro, que indica o número de trocas de posições. Uma troca envolve capturar o primeiro elemento da lista e adicioná-lo ao fim da lista.
Ex: [1,2,3,4,5] 2 => [3,4,5,1,2]
5. Definir uma função que recebe uma lista de duplas, e retorne uma lista os elementos reposicionados. Reposicionar a lista envolve capturar o segundo elemento da cabeça da lista e o primeiro elemento da dupla seguinte, enquanto seja possível.
Ex: [(10,5),(6,22),(14,8)] => [(5,6),(22,14)]
6. Definir a função zip3, que recebe 3 listas [a] -> [b] -> [c], e retorna uma lista de tuplas de tamanho 3 [(a,b,c)].
Ex: zip3 [1,2,3] ['a','b'] [True, False] => [(1,'a',True),(2,'b',False)]
7. Definir uma função que recebe uma String, avalie cada letra e retorne True caso seja vogal, e False caso contrário. Devem ser removidos os espaços em branco antes. Utilizar função de ordem superior na implementação.
Ex: ehVogal "Fim da Aula" => [False,True,False,False,True,True,True,False,True]
8. Definir uma função que recebe uma lista de strings, e inverta cada elemento da lista, e então inverta a lista resultante. Não deve ser utilizada a função reverse do Prelude.
Ex: inversoExtInt ["Fim", "da", "Aula"] => ["aluA", "ad", "miF"]