UNIVERSIDADE DO ESTADO DE SANTA CATARINA
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT

LISTA DE EXERCÍCIOS 02

Let – In, Where, List Comprehension, Funções de Ordem Superior

DISCIPLINA: Programação Funcional
SIGLA: CCI192-01B
SEMESTRE: 2019-02

1. **reversoAninhado:** Escreva uma função que recebe uma lista de strings, e retorna uma lista de strings invertidas.
   Ex: reversoAninhado ["in", "the", "end"] => ["dne", "eht", "ni"]

2. **aFrente:** Escreva uma função que recebe um elemento e uma lista de listas, e retorna uma lista de listas cujo primeiro elemento seja o recebido como parâmetro.
   Ex: aFrente 7 [[1,2], [], [3]] => [[7,1,2], [7], [7,3]]

3. **semListaVazia:** Escreva uma função que recebe uma lista de listas, e retorna uma lista de listas removendo as listas internas vazias.
   Ex: semListaVazia [[1, 2], [], [1, 3]] => [[1,2], [1, 3]]

4. **cNroPar:** Escreva uma função que recebe uma lista de listas, e retorna apenas as listas internas que contenham ao menos um número par.
   Ex: cNroPar [[1,3], [2,1], [7,9], [2, 4, 8]] => [[2,1], [2, 4, 8]]

5. **aFrentePar:** Escreva uma função recebe um número e uma lista de listas de inteiros, e retorna uma lista removendo listas internas que tenham ao menos um número ímpar, e adiciona o número recebido como argumento como primeiro elemento das listas internas caso contrário.
   Ex: aFrentePar 7 [[2, 4], [2, 3], [3, 7], [3, 4], [6, 100]] => [[7, 2, 4], [7, 6, 100]].

6. **strPrimeiros3:** Usando foldl ou foldr, escreva uma função que recebe uma lista de cadeias de caracteres (valores do tipo String) e retorna uma cadeia de caracteres que contém os 3 primeiros caracteres de cada cadeia.
   Ex: ["Abcde", "1Abcde", "12Abcde", "123Abcde"] deve retornar "Abc1Ab12A123".

7. **somaIdadePessoas:** Usando foldr ou foldl, escreva uma função que recebe uma lista de tuplas com forma: (“Maria”, 24, “4.915.766-2”) – (nome, idade, rg), e retorna a soma das idades de todos os elementos da lista.

8. **pessoaMaisNova:** Usando foldr ou foldl, escreva uma função com mesma entrada do exercício anterior, e retorne o nome da pessoa mais nova.

9. **remdups:** Usando foldr ou foldl, escreva uma função que remova os elementos iguais adjacentes de uma lista, conservando só um dos elementos.
   Ex: remdups [1,2,2,3,3,3,1,1] => [1,2,3,1].

10. **contaAdjs:** Dada uma lista ordenada xs, escreva uma função usando foldr que retorne uma lista de valores na forma: (n, e), onde **n** é o número de ocorrências de **e** em xs.
    Ex: contaAdjs "aabbbccd" => [(2,'a'),(3,'b'),(2,'c'),(1,'d')]

11. **removeLetra:** Usando foldr, escreva uma função que recebe duas strings como argumento, e remove as letras da segunda palavra que ocorrem na primeira palavra.
    Ex: removeLetra "first" "second" => "econd".

12. **scanl':** Usando foldl, escreva a função scanl', que retorna os elementos avaliados a cada interação.
    Ex: scanl' "ate" = [[], "a", "at", "ate"]