LISTA DE EXERCICIOS 03 – UNIDADE 2

PROGRAMACAO IMPERATIVA E FUNCIONAL – TURMA A E B – 2021.1

- **Deadline: Quarta-feira, dia 10/11/2021, as 23:59.**
- **Conteúdo: Listas e Tuplas.**
- **Linguagem: Haskell**
- Submeta todas as funções em um arquivo.hs único no formato L3_login.hs (por exemplo, minha submissão seria L3_ptlb.hs). Indique a qual questão cada função pertence através de comentários no código. **Qualquer arquivo enviado fora desse padrão será automaticamente descartado;**
- Não use funções padrões do Haskell (como length, tail, sum, reverse, etc), exceto quando exigido claramente pela pela questão. **O uso indevido de funções padrões resultara na anulação da questão.**
- O nome das funções implementadas **deve ser igual ao indicado pela questão**, caso contrário o script de correção não conseguira identificar sua questão e **ela será automaticamente zerada**.
- Se o tipo da função não está claramente definido na descrição da questão, considere uma função que recebe uma lista de inteiros e retorna um inteiro ou lista de inteiros;

1- Defina uma função recursiva ***elemNum :: Int -> [Int] -> Int*** que retorna o número de vezes que um elemento x aparece em uma lista xs. Se o elemento não se encontra na lista, a função deve retornar zero.

EX: elemNum 2 [1,2,3,4,2,5,2,5] => 3

EX: elemNum 5 [3,5,1,3,5,9,7] => 2

EX: elemNum 7 [1,2,3,3,4] => 0

2- Defina a mesma função anterior agora usando **list comprehension** e **funções padrões do Haskell**. *(OBS: renomeie a nova função para eleNum_2)*

3- Defina uma função recursiva ***unique :: [Int] -> [Int]*** que retorna a lista de elementos que aparece uma única vez na lista original. Se a lista apenas tiver elementos repetidos, a função deve retornar a lista vazia.

EX: unique [4,2,1,3,2,3] => [4,1]

EX: unique [1,1,1,1,1] => []

EX: unique [1,2,3,4,5] => [1,2,3,4,5]

4- Defina a mesma função agora usando **list comprehension** e **funções padrões do Haskell.** *(OBS: renomeie a nova função para unique_2)*

5- Defina a função ***orderTriple :: (Int,Int,Int) -> (Int,Int,Int)*** que coloca os elementos de uma tripla de três números inteiros em ordem ascendente. ***Dica***: use as funções **maxThree, minThree**, ou **maxAndmin** trabalhadas em exercícios anteriores para auxiliar na solução;

EX: orderTriple (30,11,45) => (11,30,45)

EX: orderTriple (5,1,7) => (1,5,7)