Lista de Exercícios Laboratório – 1
Programação Funcional – Haskell

*Resolver em dupla ou trio.

1) Apresente uma função recuriva que recebe uma lista de Inteiros e retorna o par (Tupla) (Primeiro, Último) contendo o primeiro e o último elemento dessa lista.

2) Implemente uma função chamada “palindrome” que recebe como parâmetros dois inteiros N e M e retorna um gerador de palídromos para os números de N até M (Lista).
Ex:
Entrada: 1 3 | Saída : [1,2,3,3,2,1]
Entrada: 3 6 | Saída : [3,4,5,6,6,5,4,3,2,1]

3) Escreva uma função “getElement” recebe uma lista e um número inteiro positivo para retornar o n-ésimo elemento da lista.
ex.: getElement 1 [3, 7, 4, 2] = 3

4) Escreva uma função “contais” recebe uma lista e um elemento qualquer e verifica se o elemento pertence à lista.
ex.: contais 1 [3,7,4,2] = False

5) Escreva uma função “sizeList” que recebe uma lista qualquer e retorna o número de elementos na lista.
obs.: não usar a função length

6) Escreva uma função chamada “nMedia” que recebe um inteiro, uma lista de inteiros e retorna a média dos n-primeiros elementos dessa lista.
Ex: nMedia 3 [1,2,3,4,5] = 2
nMedia 4 [1,4,1,2,4,3] = 2

7) Elabore uma função “mult” (recursiva) que recebe dois parâmetros inteiros (x) (y) e retorna (x * y) inteiro. Obs: O programador não deve usar os operadores de + ou -, EXCETO para montar duas funções de apoio :

suc::Int→Int e ant::Int→Int

8) Elabore outra função “dobroList” (Recursiva) que para cada elemento de uma lista (como entrada) multiplica por 2. (Utilize a estratégia “Currying” de acordo com a função do exercício 7).
Ex. dobroList [1, 4 , 5] = [2, 8 ,10]