

Exercícios – Programação Funcional (Linguagens de Programação)

01) Escreva uma definição recursiva da função `| intersperse :: a -> [a] -> [a] |` que intercala um elemento entre valores consecutivos em uma lista. Se a lista tiver menos de dois valores deve ficar inalterada.

```
> intersperse 0 [1 .. 4] | Saída : [1,0,2,0,3,0,4]
> intersperse ',' "abcd" | Saída : "a,b,c,d"
> intersperse ',' "a" | Saída : "a"
```

02) Escreva uma definição da função `insert` que insere um valor numa lista cujas entradas estão em ordem crescente; se o valor já ocorrer na lista, então deve ser repetido; a lista resultante deve manter a ordenação. Exemplos:

```
> insert 3 [1,2,4] | Saída : [1,2,3,4]
> insert 'd' "abcde" | Saída : "abcdde"
```

* Altere a função `insert` para receber outra lista em qualquer ordem e concatenar com a primeira mantendo a ordem.

```
> insertAlt [3,2,4,7] [1,2,4] | Saída : [1,2,2,3,4,4,7]
> insertAlt [0,3,2,4,7] [1,2,4] | Saída : [0,1,2,2,3,4,4,7]
```

03) Vamos representar as temperaturas médias de vários dias consecutivos por uma lista de valores `Float`. Escreva uma definição da função `| subidas :: [Float] -> Int |` que calcula quantas vezes a temperatura subiu (isto é, a temperatura do dia anterior foi estritamente inferior à do dia atual). No caso de a lista de temperaturas possuir menos que dois valores, o resultado deverá ser 0.

```
> subidas [19,20,21,22] | Saída : 3
> subidas [19,20,20,22] | Saída : 2
> subidas [20,19,18,19] | Saída : 1
> subidas [20,20] | Saída : 0
> subidas [20] | Saída : 0
```

04) Defina uma função `unzip::[(Int , Int)]→ ([Int] , [Int])` que transforma uma lista de pares em um par de listas.

05) Escreva uma função “`applyPairFunc`” com características de Alta Ordem para computar $f(x,y)$ entre duas listas “A” e “B” de inteiros e retornar outra lista de inteiros “C”, contendo o resultado $f(x,y) \mid x \in A \text{ e } y \in B$.

Ex1: Suponha a $f(x,y) = x*y + 5$ Entrada: lista A [1,2] lista B [2,3] Saída: [7, 11]	Ex2: Suponha a $f(x,y) = x*y + 5$ Entrada: lista A [1,2 ,3] lista B [2,3] Saída: [7, 11]
ghci: applyPairFunc (\x y→x*y+5) [1,2] [2,3] [7 , 11]	

*Note que (Ex2) a computação da função passada por parâmetro deve ser realizada apenas nos primeiros elementos do menor vetor.

- a) Implemente usando lista de compressão.
- b) Implemente usando recursão.