QUESTÕES

1. Considere as seguintes entrada e saída abaixo:

L1Q1.in

```
start 3 -1 2 0 start 4 2 5 1 0 start 2 1 -3
start 6 -8 15 0 start 5 -1 3 2
```

L1Q1.out

```
start -3 1 2 start -1 0 2 3 start 0 1 2 4 5
start -1 2 3 5 start -8 0 6 15
```

A cada linha um `start` marca o início de uma lista de números naturais que acaba na ocorrência do próximo `start` ou na quebra de linha. No arquivo de saída devem aparecer todas as listas em ordem crescente. As listas deverão também ser ordenadas entre si conforme a soma de seus elementos.

2. Implemente uma pilha de texto todos os elementos estão sempre ordenados alfabeticamente sem ferir a politica LIFO (las-in first-out).

Walter
Daniele
Carla
Bruno

$->$ (*push Antônia*)

Walter
Dani
Carla
Bruno
Antônia

Assim para push Antônia é preciso executar: pop, pop, pop, pop, push Antônia, push Bruno, push Dani, push Walter.

L1Q2.in

```
Bruno Dani Carla Antônia Walter
Maria João
```

L1Q2.out

```
push-Bruno push-Dani 1x-pop push-Carla push-Dani 3x-pop push-Antônia push-Bruno push-Carla push-Dani push-Walter
push-Maria 1x-pop push-João push-Maria
```

jose.dihego@ifba.edu.br

3. Considere uma lista duplamente ligada não-circular onde cada elemento possui uma chave inteira. Cada elemento dessa lista pode ter associado a ele uma outra lista simplesmente ligada circular também de valores reais que diferem deste por no máximo 0,99; Todas as listas devem estar ordenadas.

L1Q3.in

LE 10 9 6 4 LI 4.11 10.1 6.88 4.99 9.3 9.2 6.15 4.33

L1Q3.out

[4(4.11->4.33->4.99)->6(6.15->6.88)->9(9.2->9.3)->10(10.1)]