

(BOCA: P2_2019_Q2_a) Problema: Imagine um cenário em que se tenha duas salas (1 e 2) com uma parede de separação entre elas. A sala 1 contém um conjunto de esferas que devem ser jogadas para a sala 2 através de buracos na parede que as separa. Faça um programa para contar a quantidade de esferas que não poderão ser jogadas para a sala 2 (ou seja, esferas maiores do que o buraco) quando considerada uma configuração de diâmetros de esferas e de buracos na parede. Considere que a parede poderá não ter buracos e a sala 1 poderá não conter esferas.

Modularize o código criando:

- a) Uma função que receba um número inteiro e retorna verdadeiro se representar o final da lista e falso caso contrário. Cabeçalho: *int EhFinalDaLista(int buraco);*.
 - b) Uma função que procure o maior buraco de uma lista de diâmetros de buracos. Ela deverá assumir que os valores da lista serão lidos da entrada padrão e deverá retornar após chegar no final da lista. Cabeçalho: *int EcontraMaiorBuraco();*.
 - c) Uma função que conte as esferas grandes de uma lista de esferas considerando o maior buraco da parede (recebido por parâmetro). Ela deverá assumir que os valores da lista serão lidos da entrada padrão e deverá retornar após chegar no final da lista. Cabeçalho: *int ContaEsferasGrandes(int maiorBuraco);*.
- Entrada: uma linha com uma lista de números inteiros positivos representando os diâmetros dos buracos na parede; e, uma linha com uma lista de números inteiros positivos representando os diâmetros das esferas na sala 1. O final de cada lista será representado por -1.
 - Saída: quantidade de esferas que não podem ser jogadas para a sala 2, sendo ela dada pela mensagem "QTD:" seguida da quantidade.

- Exemplo de Entrada:

5 15 8 -1
3 5 6 15 30 8 9 -1
5 15 7 7 8 -1
10 11 -1
-1
10 11 2 6 50 -1
10 11 -1
-1

- Exemplo de Saída:

QTD:2
QTD:0
QTD:5
QTD:0