

Conhecimento Geral

O problema da mochila binária (ou empacotamento binário), ou a colocação de objetos de certos pesos em diferentes locais, sujeitos a algumas restrições são problemas de interesse histórico. Alguns dos problemas de empacotamento são NP-completos, mas são amenizados por soluções de programação dinâmica ou por soluções heurísticas que garantem uma proximidade com a solução ótima.

Neste problema, você irá resolver um problema de empacotamento que lida com reciclagem de garrafas de vidro.

O Problema

A reciclagem de vidro exige que o vidro seja separado de acordo com sua cor em uma das três categorias: marrom (*B-brown*), verde (*G-green*) e transparente (*C-clean*). Neste problema você receberá três contêineres de reciclagem, cada um contendo um número específico de garrafas marrons, verdes e transparentes. Para poderem ser recicladas, as garrafas precisarão ser movidas de forma que cada contêiner fique com garrafas de apenas uma cor.

O problema consiste em minimizar o número de garrafas que serão movidas. Você pode pressupor que o único problema é minimizar o número de movimentos entre contêiner.

Para este problema, considere que cada contêiner tem capacidade infinita e a única restrição é que você deverá mover as garrafas de forma a cada contêiner ficar com garrafas de uma única cor. O número total de garrafas nunca será maior que 2^{31} (2.147.483.648).

A Entrada

A entrada consiste de uma série de linhas, cada uma contendo 9 números inteiros. Os três primeiros números representam o número de garrafas marrons, verdes e transparentes (respectivamente) do contêiner número 1, os três inteiros seguintes correspondem, respectivamente, ao número de garrafas marrons, verdes e transparentes do contêiner número 2 e os três últimos números correspondem, respectivamente, ao número de garrafas marrons, verdes e transparentes do contêiner número 3.

Por exemplo, a linha: **10 15 20 30 12 8 15 8 31** indica que há 20 garrafas transparentes no contêiner 1, 12 garrafas verdes no contêiner 2 e 15 garrafas marrons no contêiner 3.

Os números inteiros em uma linha serão separados por um ou mais espaços. Seu programa deverá processar todas as linhas do arquivo de entrada.

A Saída

Para cada linha de entrada haverá uma linha de saída indicando que cor de garrafas irá para cada contêiner de forma a minimizar o número de movimentos de garrafas. Você também deverá imprimir o número mínimo de movimentos.

A saída consistirá de uma *string* de três caracteres maiúsculos 'G', 'B', 'C' (representando as cores verde, marrom e transparente) e corresponderão as cores associadas a cada contêiner.

O primeiro caractere dessa *string* representará a cor associada ao primeiro contêiner, o segundo caractere representará a cor associada ao segundo contêiner e o terceiro caractere corresponderá a cor associada ao terceiro contêiner.

O inteiro indicando o número mínimo de movimentos deverá aparecer após a *string*.

Se mais de uma ordem de contêineres marrom, verde e transparente atingir o número mínimo de movimentos então a representação com menor “ordem alfabética” deverá ser apresentada (por exemplo CBG tem menor ordem alfabética do que GBC).

Exemplo de Entrada

```
1 2 3 4 5 6 7 8 9
5 10 5 20 10 5 10 20 10
```

Exemplo de Saída

```
BCG 30
CBG 50
```