

Warmup – Iracema

1ª Seletiva Interna – 2013/1

Sevidor BOCA:

<http://10.20.107.205/boca/>
(acesso interno)

<http://200.19.107.205/boca/>
(acesso externo)



Organizadores:

Claudio Cesar de Sá (coordenação geral), Lucas Negri (coordenação técnica), Yuri Kaszubowski Lopes (coordenação técnica), Alexandre Gonçalves Silva (revisão técnica), Roberto Silvio Ubertino Rosso Jr., André Luiz Guedes (Dinf/UFPr)

Lembretes:

- Aos *javaneiros*: o nome da classe deve ser o mesmo nome do arquivo a ser submetido. Ex: classe `petrus`, nome do arquivo `petrus.java`;
- É permitido consultar livros, anotações ou qualquer outro material impresso durante a prova;
- A correção é automatizada, portanto, siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa. Deve-se considerar entradas e saídas padrão;
- Procure resolver o problema de maneira eficiente. Se o tempo superar o limite pré-definido, a solução não é aceita. As soluções são testadas com outras entradas além das apresentadas como exemplo dos problemas;
- Teste seu programa antes de submetê-lo. A cada problema detectado (erro de compilação, erro em tempo de execução, solução incorreta, formatação imprecisa, tempo excedido ...), há penalização de 20 minutos. O tempo é critério de desempate entre duas ou mais equipes com a mesma quantidade de problemas resolvidos;
- Utilize o *clarification* para dúvidas da prova. Os juízes podem opcionalmente atendê-lo com respostas acessíveis a todos;
- A interface KDE está disponível nas máquinas Linux, que pode ser utilizada ao invés da Unity. Para isto, basta dar *logout*, e selecionar a interface KDE. Usuário e senha: *udesc*;
- O nome *Iracema* é uma homenagem a personagem indígena de José de Alencar. Hoje, 19 de abril, dia do índio.

1 Problema A: Área dos Retângulos

Arquivo: `retangulos.[c|cpp|java]`

Meu filho está com dificuldades no trabalho de casa de matemática. Por favor, vocês poderiam ajudá-lo? Ele está trabalhando com a área do retângulo vocês sabem que, $area = comprimento \times largura$. O professor dele, forneceu uma tabela de comprimentos, larguras e áreas. Em cada linha da tabela falta um dos 3 valores; meu filho tem que calcular o valor que falta em cada linha e escrevê-lo na linha de forma que os valores em cada linha representem o comprimento, largura e área de um retângulo.

A entrada é uma série de linhas, cada uma contendo 3 inteiros, l (comprimento), w (largura) e a (área) (sendo $0 \leq l$ e $w \leq 100$, $0 \leq a \leq 10.000$) representando o comprimento, a largura e a área do retângulo, nesta ordem. Os inteiros são separados por um espaço. Em cada linha apenas um valor é substituído por um zero. A última linha contém 0 0 0 e não deve ser processada.

A saída é a mesma série de linhas mas com o zero de cada uma substituído pelo valor correto para o comprimento, largura ou a área, o que for apropriado. O novo valor será sempre um inteiro.

Exemplo de Entrada

```
2 0 6
6 5 0
0 8 80
9 0 45
0 0 0
```

Exemplo de Saída

```
2 3 6
6 5 30
10 8 80
9 5 45
```

2 Problema B: *Bug* de Memória dos Smarthphones

Arquivo: `memoria.[c|cpp|java]`

Esta história de *smarthphone* virando computador de bolso *virou a cabeça* dos experientes na área. Mas seguindo o inevitável *curso d'aguas*, o professor Claudius Virux Linux teve que comprar o seu.

Claro, já que são computadores, tem todos os componentes esperados como: cpu, RAM, memória secundária, I/O, etc. O professor imaginou que poderia ter todos aplicativos que desejasse, já que a memória secundária iria na ordem de Gigabytes, e seus aplicativos eram da ordem de Megabytes.

Contudo, ele descobriu que apenas uma fração desta memória poderia ser utilizada para armazenar aplicativos, sendo que no restante está separado e destinado somente a fotos, vídeos e músicas. Assim o professor fez um levantamento de quantos Megabytes precisavam cada uma de suas aplicações, que ele precisaria utilizar em seu *smartphone*.

O professor quer saber se os aplicativos desejados podem ser instalados no seu *smartphone*. Dada uma lista de tamanhos de aplicativos, determine se eles cabem ou não na memória reservada para aplicativos.

Entrada

A entrada contém vários casos de teste. A primeira linha de cada caso de teste contém dois inteiros, N e M , separados por um espaço, onde N é o número de aplicativos a serem instalados (entre 0 e 100) e M é a memória disponível para aplicativos (entre 0 e 100.000). A segunda linha de cada caso contém N inteiros separados por um espaço, mostrando a memória utilizada por cada aplicativo (entre 1 e 1.000).

Todos os valores de memória são expressos em Megabytes. O final da entrada é indicado por $M = N = 0$ e não deve ser processado. A entrada deve ser lida da entrada padrão.

Saída

Para cada caso de teste na entrada imprima uma linha com a mensagem:

- OK: Caso todos os aplicativos caibam na memória (somatório dos tamanhos menor ou igual a M);
- `memoria cheia`: Caso contrário.

Exemplo de Entrada

```
3 500
50 150 200
4 600
500 150 200 30
0 0
```

Exemplo de Saída

```
OK
memoria cheia
```