

Warmup – Inconfidente

1^a Seletiva Interna – 2012/1

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), Alexandre Gonçalves Silva, Roberto Silvio Ubertino Rosso Jr., André Luiz Guedes (revisão técnica – Dinf/UFPr), Gilmário, Gian Ricardo Berkenbrok, Rogério Eduardo da Silva

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.

Warmup – A Inconfidente

1^a Seletiva Interna da UDESC

20 de abril de 2012

Conteúdo

1	Problema A: Ajude um Candidato ao Doutorado!	4
2	Problema B: Insensibilidade	5
3	Problema C: Controle de Natalidade	7

1 Problema A: Ajude um Candidato ao Doutorado!

Arquivo: doutorado.[c|cpp|java]

Jon Marius esqueceu como somar dois números enquanto pesquisava em seu doutorado. E agora ele tem uma longa lista de problemas de adição que precisa resolver, além daqueles inerentes à ciência da computação! Você pode ajudá-lo?

Em sua lista atual, Jon Marius tem dois tipos de problemas: adição na forma “ $a + b$ ” e o recorrente problema “P=NP”. Jon Marius é uma pessoa muito distraída, então ele deve resolver este último problema diversas vezes, pois continua esquecendo a solução. Além disto, ele gostaria de resolver sozinho, então você deve ignorá-los.

Especificação da Entrada

A primeira linha de entrada consiste em um único inteiro N ($1 \leq N \leq 1000$), denotando o número de casos de teste. Então seguem N linhas com “P=NP” ou um problema de adição na forma “ $a + b$ ”, onde $a, b \in [0, 1000]$ são inteiros.

Especificação da Saída

Exiba o resultado de cada adição. Para linhas contendo “P=NP”, apresente “skipped”.

Exemplo de entrada

```
4
2+2
1+2
P=NP
0+0
```

Exemplo de saída

```
4
3
skipped
0
```

2 Problema B: Insensibilidade

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

O planeta Bizz fica a 133 upals de distância do planeta Terra (onde “upals” é uma unidade de medida dada por “um monte de anos-luz”), e parece ser o único planeta com vida além do nosso. Este planeta é muito interessante, pois, em cada país, seus habitantes têm uma característica diferente. Um desses países é a Cegônia, que tem como característica o fato de que todos os seus habitantes são cegos. Em compensação, todos possuem um “sexto sentido” acentuado, podendo perceber o que está à sua volta mesmo sem enxergar. Este ano, o governo da Cegônia fará um censo, e dentre os dados de seus habitantes, quer saber o quanto de insensibilidade cada pessoa possui.

A insensibilidade indica quão ruim é a capacidade das pessoas de perceber os objetos à sua volta sem precisar enxergar. Tal teste é feito da seguinte maneira: a pessoa é colocada em uma sala onde se encontram vários objetos em posições pré-determinadas. A pessoa deve, então, dizer quais são as coordenadas de cada objeto dentro da sala. Para cada objeto, calcula-se o quadrado da distância entre a posição adivinhada pela pessoa e a posição real do objeto. Este valor é chamado de D.

O nível de insensibilidade da pessoa é dado pela soma de todos os D. Por exemplo, suponha que na sala existam 4 objetos, nas coordenadas (1, 1), (3, 4), (5, 7) e (10, 10). Se a pessoa então disser que os objetos estão, respectivamente, nas posições (1, 2), (5, 4), (5, 7) e (19, 10), o valor de D para cada objeto será 1, 4, 0 e 81 e, portanto o nível de insensibilidade da pessoa é $1 + 4 + 0 + 81 = 86$.

Você precisa fazer um programa que, dadas as coordenadas verdadeiras dos objetos e as coordenadas indicadas por uma pessoa, diga qual é o nível de insensibilidade dessa pessoa.

Especificação da Entrada

A entrada é composta de vários casos de teste. A primeira linha da entrada contém um único inteiro N ($1 \leq N \leq 1000$), indicando quantos objetos estão no quarto. As N linhas seguintes contêm cada uma quatro inteiros $X1, Y1, X2, Y2$ ($0 \leq Xi \leq 1000$). Cada linha representa um objeto: a posição real do objeto é $(X1, Y1)$, e a posição onde a pessoa disse estar tal objeto é $(X2, Y2)$. O final da entrada é indicado por $N = 0$. A entrada deve ser lida da entrada padrão.

Especificação da Saída

Seu programa deve imprimir na saída padrão, uma única linha, contendo um único inteiro, indicando o nível de insensibilidade da pessoa estudada. A saída deve ser escrita na saída padrão.

Exemplo de Entrada

```
4
1 1 1 2
3 4 5 4
5 7 5 7
10 10 19 10
5
0 0 0 0
1 3 1 3
4 10 11 10
2 2 3 3
0 1 0 1
0
```

Exemplo de Saída

```
86
51
```

3 Problema C: Controle de Natalidade

Arquivo: uga.[c|cpp|java]

Na república de Uga-Uga institui-se o controle de natalidade. A regra para o contador Og, barnabé do ministério da saúde, era simples: no máximo, cada casal poderia ter até 10 filhos, distribuídos no máximo em 5 meninas e 5 meninos.

Sim, uma regra simples em face de cada mão ter 5 dedos. A mão esquerda é usada para contar as meninas, e a mão direita os meninos. Og estava feliz com isto.

Sua missão: escrever um programa para ajudar o Og a ler os dados de cada mão e imprimir o total de filhos de cada casa.

Especificação da Entrada

A entrada contém vários casos de teste. Cada caso de teste consiste de uma única linha contendo dois inteiros **E** e **D**, separadas por um espaço simples, indicando respectivamente o número de filhas e filhos. Para o caso valores válidos a serem somados: $0 \leq E \leq 5$ e $0 \leq D \leq 5$.

Caso algum destes valores sejam negativos, algo inviável para o homem das cavernas, pois os negativos foram descobertos após esta época, trata-se aí de um problema NP (leia-se: “**Nem a Pedra**”).

O final da entrada é indicado por $E = D = 0$. A entrada deve ser lida da entrada padrão.

Especificação da Saída

Para cada caso de teste na entrada imprima uma linha contendo um único inteiro indicando quantos filhos tem caverna. Caso algum destes valores sejam negativos, seu programa deve imprimir a indignação de Og: Og diz: Eh um NP! (o símbolo ‘!’ ao final da frase é de indignação.).

A saída deve ser escrita na saída padrão.

Exemplo de Entrada

```
3 4
5 -5
5 5
-1 -1
0 5
5 0
0 0
```

Exemplo de Saída

```
7
Og diz: Eh um NP!
10
Og diz: Eh um NP!
5
5
```