

Universidade do Estado Santa Catarina (UDESC)

Universidade Federal do Paraná (UFPR)

Maratona Doméstica de Programação – 2009 – 1

(*Warm up* – Dona Vanda)

`http://200.18.7.16:8001/boca`

Organizadores: Claudio (DCC), Rosso (DCC), André Guedes (DInf)

23 de maio de 2009



Lembrete:

- É permitido consultar livros, anotações ou qualquer outro material impresso cópia durante a prova.
- A correção é automatizada, portanto siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa.
- Todos os problemas têm o mesmo valor na correção. Logo, leia e comece com os mais fáceis.
- Procure resolver o problema de maneira eficiente. Na correção, a eficiência também é levada em conta. As soluções serão testadas com outras entradas além das apresentadas como exemplo dos problemas.
- Utilize o *Clarification* para dúvidas da prova. Opcionalmente, os juízes respondem, contudo, os esclarecimentos são acessíveis a todos.
- Responsável pela construção da prova: Claudio – UDESC
- Revisão Técnica: André – UFPR

Agradecimentos

- Ao pessoal do suporte da UDESC, do DCC, Grupo Colméia, ...
- Pelas traduções: Fernando Deeke Sasse (DMAT), Alexandre (DCC), Adriano (DCC) e o Dênio (DCC).
- Ao patrocínio da TOTVS.
- Ao DCC e demais anônimos que tornaram este evento uma realidade.

Sumário

1	Problema A: Cara ou Coroa	3
2	Problema B: Retângulos	4

1 Problema A: Cara ou Coroa

Arquivo: cara_coroa.[c|cpp|java]

Quando eu estava na escola, muitos, muitos anos atrás, nós costumávamos jogar um jogo simples que envolvia jogar uma moeda. O primeiro jogador podia dizer a chamada de "cara" ou "coroa", o segundo jogava a moeda. O primeiro jogador ganhava um ponto por cada acerto, o segundo jogador ganhava um ponto por cada chamada errada. Quando nos aborrecíamos com o jogo somávamos os escores! A entrada deste problema consiste num certo número de jogos. Cada jogo inicia com uma linha contendo dois nomes de não mais que 20 letras cada e separados por um espaço em branco. A segunda linha de cada jogo contém apenas um número inteiro positivo n , especificando o número de vezes que a moeda foi jogada ($0 < n \leq 1000$). As n jogadas seguem, cada uma em uma linha contendo 2 caracteres ['H' ou 'T' "Head"= Cara em inglês; Tail = "Coroa" em inglês], separados por um espaço. O primeiro caractere representa a chamada e o segundo o resultado. A entrada será finalizada por uma linha contendo apenas # #. A saída será uma linha para cada jogo. A linha conterá o nome do primeiro jogador(como gravado na Entrada), seguida pelo escore do primeiro jogador, seguida pelo nome do segundo jogador, seguido pelo escore do segundo jogador. Entradas em uma linha são separadas por um único espaço.

Exemplo de Entrada

```
Sally John
5
H H
H T
H H
T T
T H
Eloise Ahmed
4
H T
T T
H H
T H
# #
```

Exemplo de Saída

```
Sally 3 John 2
Eloise 2 Ahmed 2
```

2 Problema B: 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
```