

Universidade do Estado Santa Catarina – UDESC

Universidade Federal do Paraná – UFPr

UNICENTRO – Guarapuava – Pr.

Maratona Doméstica de Programação – 2008-2

(Warm up para Lurdes)

<http://maratona.c3sl.ufpr.br/boca/index.php>

Organizadores: Claudio (DCC), André Guedes (DInf)
e Eleandro (UNICENTRO)

3 de setembro de 2008



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.
- Nas implementações, as entradas e as saídas conhecidas serão padrão (ler e escrever na saída padrão).
- Utilize o *Clarification* para dúvidas da prova. Opcionalmente, os juízes respondem, contudo, os esclarecimentos são acessíveis a todos.
- A novidade é o nome dos programas que seguem um padrão no prefixo, definido por uma letra, relativo ao problema submetido. Exemplo: `a_warmup.c`, neste caso, refere-se ao problema 'a'.

Agradecimentos

- Ao pessoal do suporte da UDESC, Grupo Colméia, C^3 SL da UFPr, e a turma do André na UFPr.
- Pelas traduções: Fernando Sasse, Rosso, Marcelo e o Wanderley.
- Ao patrocínio da Datasul.
- Aos apoiadores: Kugel, Google, Rede Globo, ... e demais (caso tenhamos esquecido alguém).
- Ao DCC e demais anônimos que tornaram este evento uma realidade.

Problema A: Cara ou Coroa

Arquivo: a_warmup.[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
```

Problema B: Jack Compulsivo

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

Jack Compulsivo é um homem com muitos problemas. Ele não consegue viver sem ter sua vida organizada nos mínimos detalhes de acordo com suas manias. Uma de suas manias é ter os cabides de roupas em seu armário sempre em uma determinada ordem. Nesta ordem, bastante simples, todas as roupas já usadas devem estar sempre à esquerda de todas as roupas novas. Entretanto, enquanto ele trabalhava, sua diarista, Maria Limpeza, arrumou o seu armário e trocou todas as roupas de lugar. Quando Jack chegou quase teve um colapso.

Para evitar ter que fazer tirar e colocar muitos cabides, Jack pretende arrumar tudo fazendo algumas trocas de 2 cabides por vez. Ele pediu a sua ajuda para fazer um programa para determinar qual o menor número de trocas que é necessário para deixar o armário como ele quer. O número de cabides no armário é no máximo 10.000.

Entrada

A entrada é composta de vários casos de teste. Cada caso de teste é uma linha com '0's e '1's (sem espaços) onde os '0's representam as roupas já usadas e os '1's, as roupas novas.

Saída

A saída para cada caso de teste é uma linha com o número mínimo de trocas necessário para se fazer a arrumação.

Exemplo de Entrada

```
00101001001
10000000011111111110
0000100011010111101
```

Exemplo de Saída

```
2
1
3
```