



acm International Collegiate
Programming Contest

2006



event
sponsor

Maratona de Programação da SBC 2006

Sub-Regional Brasil do ACM ICPC

9 de Setembro de 2006

Sessão de Aquecimento

(Este caderno contém 2 problemas; as páginas estão numeradas de 1 a 2, não contando esta página de rosto)

Sedes Regionais

Região Centro-Oeste

- Universidade de Brasília, Brasília, DF
- UNAES, Campo Grande, MS
- Universidade Estadual do Mato Grosso do Sul, Dourados, MS

Região Nordeste

- Unifor, Fortaleza, CE
- Universidade Federal do Rio Grande do Norte, Natal, RN
- AESO, Olinda, PE
- Unime, Salvador, BA
- Universidade Federal do Maranhão, São Luís, MA

Região Sul

- Unioeste, Cascavel, PR
- URI, Erechim, RS
- Unisul, Florianópolis, SC
- Universidade Federal do Rio Grande do Sul, Porto Alegre, RG
- FURG, Rio Grande, RS

Região Sudeste

- Universidade Federal de Minas Gerais, Belo Horizonte, MG
- Metrocamp, Campinas, SP
- Faculdades Módulo, Caraguatatuba, SP
- Universo, Juiz de Fora, MG
- PUC Poços de Caldas, Poços de Caldas, MG
- Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ
- Uninove, São Paulo, SP
- Unitau, Taubaté, SP
- Unitri, Ubertlândia, MG
- Centro Universitário de Vila Velha, Vila Velha, ES

Região Norte

- Centro Universitário do Pará, Belém, PA
- Faculdade SEAMA, Macapá, AP
- Universidade Federal de Tocantins, Palmas, TO

Promoção: Sociedade Brasileira de Computação
Patrocínio: Fundação Carlos Chagas – IBM – Ci&T – Microsoft – UOL

Problema A

Fechem as portas!

Nome do arquivo fonte: portas.c, portas.cpp, portas.java ou portas.pas

Madame Beauvoir possui uma mansão onde ela recebe todos os seus descendentes (netos e bisnetos) durante as férias. Sua mansão possui exatamente N quartos (cada quarto é numerado de 1 a N), onde N é também a quantidade de netos e bisnetos (cada descendente é também numerado de 1 a N).

Como toda criança, os descendentes de Mme. Beauvoir são bastante travessos. Todo dia é a mesma confusão: eles acordam de manhã cedo antes dela e se encontram no grande jardim. Cada descendente, um de cada vez, entra na mansão e troca o estado das portas dos quartos cujos números são múltiplos do seu identificador. Trocar o estado de uma porta significa fechar uma porta que estava aberta ou abrir uma porta que estava fechada. Por exemplo, o descendente cujo identificador é igual a 15 vai trocar o estado das portas 15, 30, 45, etc.

Considerando que todas as portas estão inicialmente fechadas (todos os descendentes fecham as portas antes de descer para o jardim) e que cada descendente entra exatamente uma vez na mansão (a confusão é tão grande que não sabemos em que ordem), quais portas estarão abertas após a entrada de todos os descendentes na mansão?

Entrada

A entrada contém vários casos de teste. Cada caso de teste consiste em uma linha que contém um inteiro N ($0 \leq N \leq 25000000$), indicando o número de portas e descendentes. O final da entrada é indicado por $N = 0$.

A entrada deve ser lida da entrada padrão.

Saída

Para cada caso de teste da entrada seu programa deve produzir uma linha na saída, contendo a seqüência crescente de números correspondente aos identificadores dos quartos cujas portas estarão abertas. Ao imprimir a seqüência, deixe um espaço em branco entre dois elementos consecutivos.

A saída deve ser escrita na saída padrão.

| Exemplo de entrada | Saída para o exemplo de entrada |
|--------------------|---------------------------------|
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |
| 4 | 1 4 |
| 0 | |

Problema B

Esquerda, Volver!

Nome do arquivo fonte: esquerda.c, esquerda.cpp, esquerda.java ou esquerda.pas

Este ano o sargento está tendo mais trabalho do que de costume para treinar os recrutas. Um deles é muito atrapalhado, e de vez em quando faz tudo errado – por exemplo, ao invés de virar à direita quando comandado, vira à esquerda, causando grande confusão no batalhão.

O sargento tem fama de durão e não vai deixar o recruta em paz enquanto este não aprender a executar corretamente os comandos. No sábado à tarde, enquanto todos os outros recrutas estão de folga, ele obrigou o recruta a fazer um treinamento extra. Com o recruta marchando parado no mesmo lugar, o sargento emitiu uma série de comandos “esquerda volver!” e “direita volver!”. A cada comando, o recruta deve girar sobre o mesmo ponto e dar um quarto de volta na direção correspondente ao comando. Por exemplo, se o recruta está inicialmente com o rosto voltado para a direção norte, após um comando de “esquerda volver!” ele deve ficar com o rosto voltado para a direção oeste. Se o recruta está inicialmente com o rosto voltado para o leste, após um comando “direita, volver!” ele deve ter o rosto voltado para o sul.

No entanto, durante o treinamento, em que o recruta tinha inicialmente o rosto voltado para o norte, o sargento emitiu uma série tão extensa de comandos, e tão rapidamente, que até ele ficou confuso, e não sabe mais para qual direção o recruta deve ter seu rosto voltado após executar todos os comandos. Você pode ajudar o sargento?

Entrada

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém um inteiro N que indica o número de comandos emitidos pelo sargento ($1 \leq N \leq 1000$). A segunda linha contém N caracteres, descrevendo a série de comandos emitidos pelo sargento. Cada comando é representado por uma letra: ‘E’ (para “esquerda, volver!”) e ‘D’ (para “direita, volver!”). O final da entrada é indicado por $N = 0$.

A entrada deve ser lida da entrada padrão.

Saída

Para cada caso de teste da entrada seu programa deve produzir uma única linha da saída, indicando a direção para a qual o recruta deve ter sua face voltada após executar a série de comandos, considerando que no início o recruta tem a face voltada para o norte. A linha deve conter uma letra entre ‘N’, ‘L’, ‘S’ e ‘O’, representando respectivamente as direções norte, leste, sul e oeste.

A saída deve ser escrita na saída padrão.

| Exemplo de entrada | Saída para o exemplo de entrada |
|--------------------|---------------------------------|
| 3 | L |
| DDE | S |
| 2 | |
| EE | |
| 0 | |