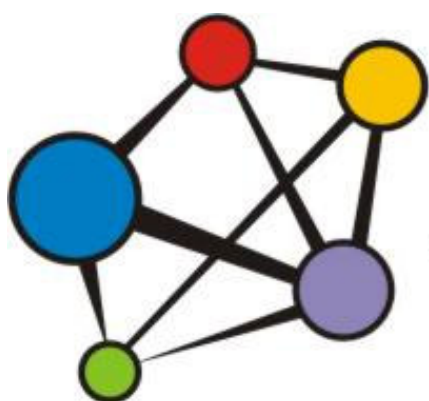




2^a Maratona de
Programação

FACENS 2006

PROBLEMA A: CUBO MÁGICO

Cor: Azul

Nomes dos Arquivos:

Arquivo-fonte: cubo.c, cubo.pas, etc

Descrição do problema:

Um brinquedo muito conhecido é o cubo mágico, que nada mais é que um cubo conforme a figura 1(a), com 9 quadrados coloridos em cada face. O objetivo do brinquedo é deixar cada face com apenas uma cor, através de rotações nas suas faces, conforme a figura 1(b).

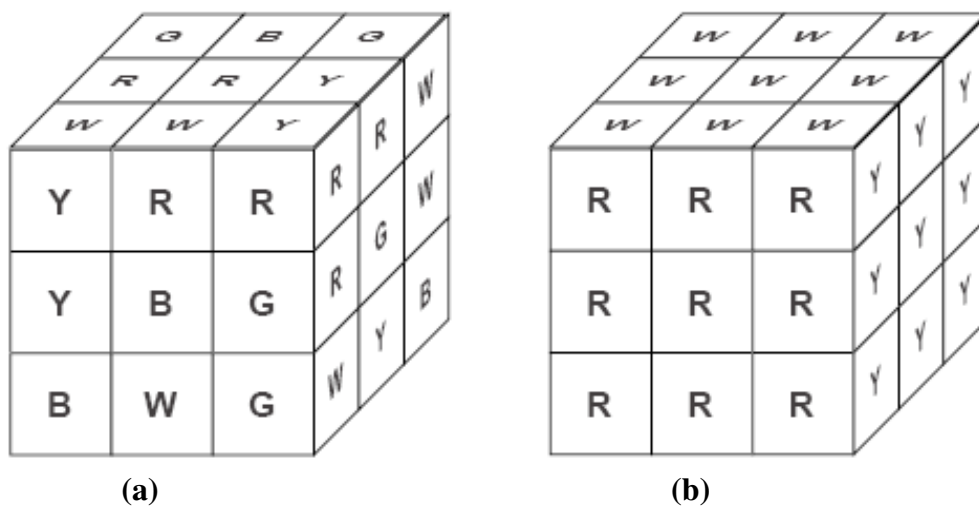


Figura 1: (a) configuração arbitrária (b) configuração final

Cada vez que uma face é rodada, todas as suas faces adjacentes são alteradas, conforme indicado na figura 2. Dada uma configuração qualquer do cubo, vencer o jogo pode ser bastante desafiador. Para auxiliar nesta tarefa, sua equipe deve construir um programa para testar se uma determinada seqüência de operações pode levar a configuração final desejada.

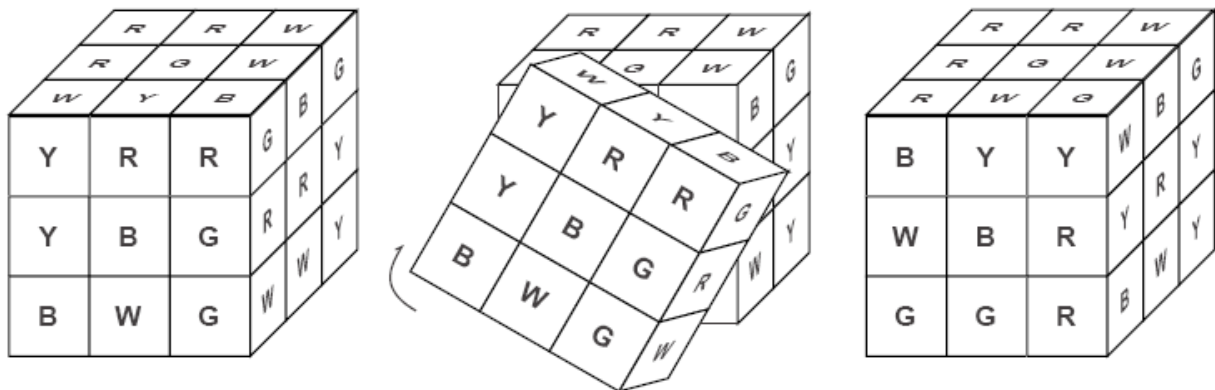


Figura 2: exemplo de rotação

Serão fornecidos para o seu programa uma configuração inicial e uma seqüência de rotações. O

Cubo será representado conforme a figura 3. As cores serão (Y-Amarela, B-Azul, G-Verde, R-Vermelho, W-Branco, M-Rosa). As rotações serão representadas como números inteiros. Valores positivos indicam rotações no sentido horário, e valores negativos indicam rotações no sentido anti-horário.

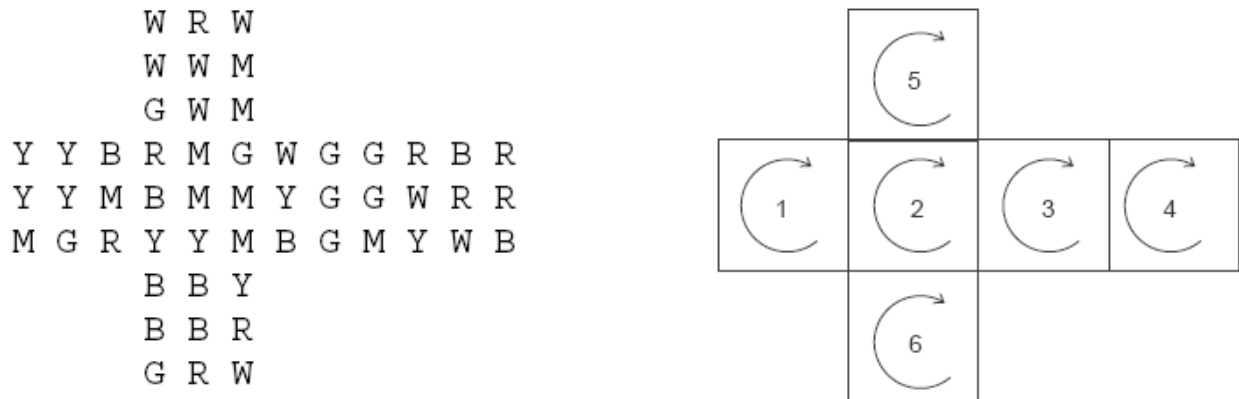


Figura 3: representação das faces do cubo e rotações.

Entrada:

O arquivo de entrada possui N casos de teste, sendo N indicado na primeira linha do arquivo. Cada caso de teste possui 10 linhas, sendo 9 linhas para representar o cubo. A última linha possui uma lista de rotações a realizar, sendo cada inteiro uma rotação na face correspondente. A lista termina com o valor zero.

Saída:

Cada caso de teste resultará em uma única saída, com a letra V de vencedor caso cada face do cubo fique com apenas uma cor ou P de perdedor caso contrário.

Exemplo:

Entrada:	Saída:
3 G Y Y G Y Y G Y Y W W W Y R R M M M G G B W W W Y R R M M M G G B W W W Y R R M M M G G B R B B R B B R B B -1 0 G Y Y G Y Y G Y Y W W W Y R R M M M G G B W M W Y R R M W M G G B W W W Y R R M M M G G B R B B R B B R B B -1 0 M W M W W G W W Y G Y Y M M B M B G W R B B Y Y M M B M G G W R R Y M G W B B R R G R R W R Y Y G B Y R G B +4 +6 -2 +3 -4 +2 -3 -6 0	V P V

PROBLEMA B: FÓRMULA 1

Cor: Amarelo

Nomes dos Arquivos:

Arquivo-fonte: fl.c, fl.pas, etc

Descrição do problema:

Os maiores desejos de um piloto de Fórmula 1 é que seu carro acelere o mais rápido possível para atingir a maior velocidade e o outro é que seja possível desacelerar de maneira sutil, o que nem sempre é uma tarefa simples. Por exemplo, há um trecho na pista do Canadá no qual, em apenas 2,5 segundos, o carro vai de 300 km/h para apenas 60 km/h.

No mundo das altas velocidades qualquer movimento mínimo incorreto, pode se tornar o fator crucial para decisão de vitória. Imagine estar em uma estrada, voltando do trabalho para sua casa (já fez aquele percurso repetidas vezes e conhece cada detalhe do trajeto), mas numa curva você pisa um pouquinho mais forte no freio, o que faz seu carro rodar sem controle. É exatamente isso o que acontece na Fórmula 1, qualquer variação, mesmo que mínima, pode levar ao desastre.

Para estudar com precisão as variações de velocidade de um carro em uma pista de fórmula 1, uma empresa coletou amostras de velocidade durante uma corrida de 1 em 1 segundo. Uma informação importante é que os pesquisadores desta empresa desejam descobrir como se comporta a média de velocidade, considerada em intervalos de tempo de uma dada duração (5 segundos, 1 minuto, etc.).

Exemplo, para uma seqüência de medições 30, 50, 90, 120, 160, 210, 270, 320 e intervalos de quatro segundos, as velocidades médias são respectivamente: $290 / 4 = 72.5$, $420 / 4 = 105$, $580 / 4 = 145$, $760 / 4 = 190$, $960 / 4 = 240$.

Sua equipe foi contratada para desenvolver um programa que conhecida às velocidades e o tamanho do intervalo desejado, informe quais são as velocidades médias e a maior e menor velocidade média obtida no intervalo dado. Lembre-se que não temos velocidades negativas.

Entrada

A entrada é constituída de vários conjuntos de testes, onde na primeira linha de um conjunto de teste contém dois números inteiros positivos P e Q, que indicam respectivamente o número total de medições de velocidade de uma seqüência obtida, e o tamanho dos intervalos (em segundos), em que as médias devem ser calculadas. As P linhas seguintes contém cada uma das seqüências de velocidade medidas. Os valores de P ou Q iguais a zero indicam o fim dos casos de testes.

O número de velocidades medidas, assim como o intervalo o qual se deseja obter as médias devem ser inteiros positivos e as velocidades devem ser positivas e maior que zero, caso contrário, deverá sinalizar que a entrada é inválida.

Saída

Para cada conjunto de teste de entrada, o programa deverá imprimir “Velocidades Medias: n”, onde n é numerado a partir de 1. Também deve imprimir todas as médias obtidas para o teste em questão, e a posição da menor e maior velocidades médias obtidas.

Exemplos

Entrada:	Saída
4 2 10.00 50.00 70.00 20.00 2 1 -20.00 50.00 0 0	Velocidades Medias: 1 30.0 60.0 45.0 Menor: 1 Maior: 2 Velocidades Medias: 2 Dados invalidos

PROBLEMA C: DAMAS

Cor: Vermelho

Nomes dos Arquivos:

Arquivo-fonte: damas.c, damas.pas, etc

Descrição do problema:

Sua equipe faz parte de uma multinacional que trabalha em desenvolvimento de jogos. Um dos projetos da empresa contempla o desenvolvimento de uma série de jogos de tabuleiro para celulares, sendo que o usuário, de seu celular, jogará com um oponente humano em outro celular.

Dentro deste projeto, uma das funcionalidades será o modo tutorial. Neste modo, ao escolher uma pedra do tabuleiro, o usuário poderá ver todas as casas que esta pedra poderia ocupar, seja através do movimento simples, através da captura de uma peça ou através da captura de várias peças, conforme indicado na figura 1.

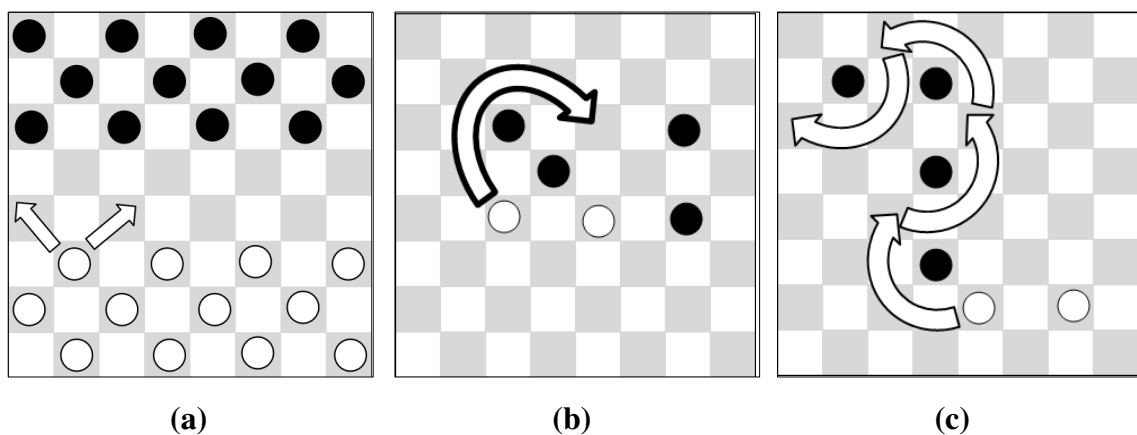


Figura 1: (a) movimentação simples de uma peça, (b) captura de uma peça e (c) captura de várias peças.

Sua equipe ficou responsável pelo modo tutorial. Você deve criar um programa que, dada uma configuração de tabuleiro e uma determinada peça, seja capaz de marcar todas as casas alcançáveis por aquela determinada peça.

Entrada

A entrada é composta de vários conjuntos de teste. O primeiro inteiro representa o número de casos de teste. Em seguida, uma matriz 8x8 representa uma configuração de tabuleiro. As brancas sempre começam embaixo. Peças brancas são representadas pela letra "B", pretas pela letra "P" e as casas livres pelo caractere "_". Depois da matriz, um par de inteiros indica a peça que se deseja movimentar. A coordenada é no formato (linha, coluna), começando da diagonal superior esquerda do tabuleiro com (0,0).

Saída

Para cada caso de teste deve ser colocado na saída uma matriz 8x8 marcando "X" nas posições que

Entrada:

3	
B	
4 3	
P P P	P P P
P P	P P
P	P
P	X B
B	B X
B	B B
B B	B
B	
3 2	
P P	P X P
P P	P P
P	X P X
P P	P P
B	B
B	B
B B	B B
B	B
4 3	

PROBLEMA D: LABIRINTO – O RETORNO

Cor: Verde

Nomes dos Arquivos:

Arquivo-fonte: lab.c, lab.pas, etc

Descrição do problema:

Na maratona de programação do ano passado foi proposto um problema de labirinto, que seria aplicado na construção de um jogo on-line. Embora interessante, o problema tinha uma séria restrição: o personagem, para achar a saída, precisava conhecer o mapa completo. Uma situação mais comum é que o personagem esteja em um labirinto e não possa ver os caminhos nem a saída, e tenha que achá-la. Na verdade, tipicamente, o personagem só pode ver a situação ao seu redor ou próxima a ele.

Nesta nova versão do problema, o personagem começará em uma posição qualquer de um labirinto, sem poder ver absolutamente nada a não ser as quatro posições ao seu redor: norte, leste, sul e oeste. O personagem caminhará de forma arbitrária pelo labirinto procurando pela saída e por um tesouro. Enquanto caminha, suas novas observações do ambiente permitem que ele crie uma representação interna de como é o labirinto. Uma vez encontrada a saída e o tesouro, o personagem deve traçar a menor rota entre o tesouro e a saída.

Entrada

A entrada possui vários casos de teste. Cada caso começa com um número inteiro ($0 < N < 300$) que representa o número de movimentos do personagem. Número de movimentos zero representa o final dos casos de teste. Em seguida, cada linha representa um movimento, sendo a primeira letra a direção (N-Norte, S-Sul, L-Leste, O-Oeste), e as outras quatro o que o personagem observa olhando para as quatro direções a sua volta, na sequência norte, leste, sul e oeste. As observações possíveis são a entrada (E), a saída (S), o tesouro (T), corredores vazios () e obstáculos (O). O início do centro de coordenadas, (0,0) deve ser considerado na posição em que o personagem começa.

Saída

Para cada caso de teste na saída devem ser indicadas as coordenadas, desde o tesouro até a saída do labirinto, pelo caminho mínimo. A linha de saída de um caso de teste deve apresentar, primeiramente, o número de passos a serem executados. Em seguida, em pares separados por vírgula, são indicadas as coordenadas, separadas por espaços, para se sair do labirinto.

Exemplos

Entrada:	Saída
9	7 -3,-3 -3,-2 -3,-1, -3,0, -
N _OEO	3,1 -3,2 -3,3
N _O_O	12 -4,0 -3,0 -3,1 -3,2 -3,3 -
N O__	2,3 -1,3 0,3 1,3 2,3 3,3 3,2
L O_O_	
L OSO_	
O O_O_	
O O__	
O O_O_	
O O_OT	
26	
O OEO_	
O O_O_	
O __O	
S _O_O	
S _O_O	
S __OO	
L O_O_	
L O_O_	
L __O_	
L OSO_	
L O_O_	
L _OOS	
N _O_O	
N _O_O	
N _O_O	
N _O_O	
N _O_O	
N OO__	
O O_O_	
O O_O_	
O T_O_	
O O_O_	
O O_O_	
O O__O	
S _O_O	
S _O_O	
S __O	
0	

PROBLEMA E: URNA ELETRÔNICA

Cor: Laranja

Nomes dos Arquivos:

Arquivo-fonte: urna.c, urna.pas, etc

Descrição do problema:

O processo eleitoral brasileiro é conhecido internacionalmente, pela sua rapidez, fruto da utilização da tecnologia. Para auxiliar neste exercício da democracia, sua equipe de programadores foi recrutada. Vocês devem conhecer um pouco mais sobre este processo para tal fim.

Para que um candidato vença as eleições no primeiro turno é necessário que ele tenha 50% mais um voto do total de votos válidos. A regra começou a ser usada na eleição presidencial de 1989. Vale para cargos do Poder Executivo (presidente da República, governador de Estado e prefeito). Os cargos para o Poder Legislativo (deputados e senadores) são preenchidos em turno único, por maioria simples (ganha quem tem mais votos). Os votos válidos são aqueles dados aos candidatos. Não valem para a conta, portanto, os votos brancos e nulos.

Por exemplo, digamos que no dia da eleição, 100 eleitores compareçam para a votação. Seriam então apurados 100 votos. Esses são os votos totais. Na hora da apuração, contou-se o seguinte:

- candidato A - 43 votos
- candidato B - 41 votos
- brancos - 9 votos
- nulos - 7 votos

Como se observa, a soma dos votos apurados é 100, porém, os válidos são apenas 84, ou seja, aqueles recebidos pelos candidatos A e B. Para vencer essa eleição no primeiro turno, um candidato teria ter 42 votos mais 1, em outras palavra 50% do votos válidos mais um voto. É o que recebeu o candidato A que foi eleito no primeiro turno.

Além de saber o resultado das eleições, se algum candidato ganhou no primeiro turno ou se haverá segundo turno e quem são os candidatos que participarão, é extremamente importante para os partidos conhecerem detalhes das eleições. Dentre os detalhes estão: Total de votos válidos, total de votos por candidatos e porcentagem de votos válidos recebidos por candidato. São justamente estas informações que sua equipe de desenvolvimento deve fornecer, através do processamento dos dados consolidados das urnas.

Entrada

A entrada possui vários casos de teste. Cada caso é iniciado com um número da zona eleitoral ($100 < Z < 350$) seguido por um número inteiro que representa a quantidade de seções ($0 < Q < 100$) dessa zona. A linha seguinte conterá o número da seção ($0 < S < 130$). Cada linha seguinte representará os votos de um candidato, sendo que o primeiro número representa o candidato e o segundo a quantidade de votos obtidos na seção. Caso um candidato não possua votos na seção, na

posição que indica a quantidade de votos recebidos aparecerá o número zero. Para a eleição a presidência da república existem 4 candidatos cujos números são: 11, 21, 23 e 35. Após receber os votos dos candidatos, serão informados os votos Brancos (BR) e Nulos (NU) seguindo o mesmo padrão de entrada: sigla e quantidade.

Ao receber zona eleitoral zero e quantidade de seções também zero indica que um determinado caso de teste foi finalizado e a totalização deve ser realizada. Caso zona eleitoral seja -1 e quantidade de seções também -1, indica o final dos casos de testes.

Saída

Para cada caso de teste devem ser exibidos na saída o total de votos válidos, o total de votos por candidato e a porcentagem de votos recebidos por candidato. Além disso, deve informar se existe um vencedor no primeiro turno ou caso exista segundo turno, quais serão os candidatos que participarão. Caso o número de votos de um candidato seja um valor inválido (negativo), deve-se informar que a seção foi impugnada e os votos dessa seção serão descartados, ou seja, não serão computados nos totais de votos.

Exemplos

Entrada:	Saída
100 2	Zona Eleitoral: 100
12	Total de votos validos: 1444
11 253	
21 158	Zona Eleitoral: 231
23 200	Secao 23 impugnada
35 12	Total de votos validos: 211
BR 64	
NU 178	Candidato 11: 536 (32.39%)
13	Candidato 21: 205 (12.39%)
11 258	Candidato 23: 320 (19.34%)
21 0	Candidato 35: 594 (35.89%)
23 105	BR: 289 (17.46%)
35 458	NU: 532 (32.15%)
BR 25	Havera segundo turno:
NU 204	Candidato 35
231 2	Candidato 11
23	
11 -1	Zona Eleitoral: 120
21 100	Total de votos validos: 186
23 120	
35 470	Candidato 11: 154 (82.80%)
BR 2	Candidato 21: 30 (16.13%)
NU 18	Candidato 23: 1 (0.54%)
15	Candidato 35: 1 (0.54%)
11 25	BR: 0 (0.00%)
21 47	NU: 100 (53.76%)
23 15	Candidato 11 vence no primeiro turno
35 124	
BR 200	
NU 150	
0 0	
120 1	

45	
11	154
21	30
23	1
35	1
BR	0
NU	100
0	0
-1	-1

PROBLEMA F: DISCOS

Cor: Roxo

Nomes dos Arquivos:

Arquivo-fonte: discos.c, discos.pas, etc

Descrição do problema:

Os nove números da figura 1 (a) representam as posições de nove discos, onde cada disco só pode assumir uma das quatro posições: norte ou 12 horas (representado por 0), leste ou 3 horas (representado por 1), sul ou 6 horas (representado por 2) e oeste ou 9 horas (representado por 3), conforme demonstrado na figura 1(b).

3 3 0
2 2 2
2 1 2

(a)

0
3 + 1
2

(b)

Figura 1: (a) posições dos discos e (b) possíveis posições dos discos

Existem nove maneiras diferentes de movimentar os discos, cada maneira é chamada de movimento e cada movimento é selecionado por um número de 1 à 9. Os locais no movimento (que representam cada um dos discos) marcados com “x” devem rotacionar o disco em noventa graus no sentido horário de acordo com a figura 2 (exceto para os discos que estão na posição 0). Já os locais no movimento marcados com “z” não devem alterar os discos. Os movimentos são apresentados na figura 2.

x x z	x x x	z x x
x x z	z z z	z x x
z z z	z z z	z z z
Movimento 1	Movimento 2	Movimento 3
x z z	z x z	z z x
x z z	x x x	z z x
x z z	z x z	z z x
Movimento 4	Movimento 5	Movimento 6
z z z	z z z	z z z
x x z	z z z	z x x
x x z	x x x	z x x
Movimento 7	Movimento 8	Movimento 9

Figura 2: movimentos possíveis

O objetivo é deixar todos os discos alinhados em 0, ou seja, na posição norte, mas para isso é preciso seguir algumas regras:

A prioridade na escolha do movimento deve ser a seguinte: o movimento deve ser aquele que movimenta o maior número de discos em posições mais baixas (exceto a posição 0). Exemplo: se houver discos na posição um primeiramente deve ser escolhido o movimento que altere o maior número de discos que estão na posição 1, depois o maior número de discos que estão na posição 2 e assim sucessivamente. Caso dois movimentos satisfaçam essa condição, deverá ser selecionado o movimento de índice menor.

Por exemplo, dados os discos nas seguintes posições:

3 3 0
2 2 2
2 1 2

O primeiro movimento a ser selecionado será o Movimento 5, pois embora existam outros movimentos que alterem o disco que está na posição 1, esse é o movimento de menor índice. Após o movimento os discos devem estar na seguinte posição:

3 0 0
3 3 3
2 2 2

O segundo movimento a ser selecionado deverá ser o Movimento 8, pois altera o maior número de discos na posição 2. Após o movimento os discos deverão estar na seguinte posição:

3 0 0
3 3 3
3 3 3

O terceiro movimento a ser selecionado será o Movimento 5 novamente, pois embora existam outros movimentos que alterem o disco que está na posição 3, esse é o movimento de menor índice. Após o movimento os discos devem estar na seguinte posição:

3 0 0
0 0 0
3 0 3

O quarto movimento a ser selecionado será o Movimento 4, pois altera o maior número de discos na posição 3. Após o movimento os discos deverão estar na seguinte posição:

0 0 0
0 0 0
0 0 3

O quinto e último movimento a ser selecionado será o Movimento 6, pois embora existam outros movimentos que alterem o disco que está na posição 3, esse é o movimento de menor índice. Após o movimento os discos devem estar na seguinte posição:

0 0 0
0 0 0
0 0 0

O problema consiste em desenvolver um programa que dada uma entrada de nove discos em qualquer uma das quatro posições encontre a sequência de movimentos para que todos os discos sejam colocados na posição 0, seguindo as regras propostas.

Entrada

Como entrada o programa deverá receber vários conjuntos de testes, sendo que a primeira linha de um conjunto de teste contém uma letra. Caso essa letra seja diferente de 'E' ou 'e' o programa deverá ser encerrado. Se a letra contida na primeira linha for 'E' ou 'e', deverão ser recebidas três linhas cada uma com três posições de discos que são valores inteiro no intervalo de 0 à 3 separados por um espaço.

Saída

Para cada conjunto de teste de entrada, o programa deverá imprimir o índice dos movimentos realizados separados por “;” (ponto-e-vírgula), para se atingir a posição 0 em todos os discos. Caso seja atribuído um valor para o disco diferente do intervalo de 0 à 3 uma mensagem “Erro” deve ser impressa. Se não houver movimentos a serem realizados mostrar a mensagem “Movimentos Inexistentes”.

Exemplos

Entrada:	Saída
E	5;8;5;4;6;
3 3 0	5;8;1;5;8;
2 2 2	Erro
2 1 2	1;8;5;1;6;8;1;
e	
3 1 0	
0 3 2	
2 1 2	
E	
3 1 0	
0 4 2	
2 1 2	
e	
1 1 0	
0 3 2	
2 1 1	
S	