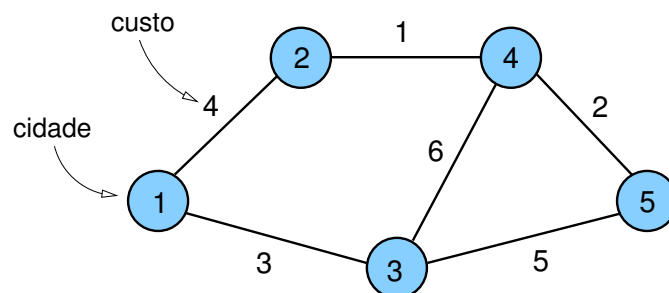


# Frete

Nome do arquivo: `frete.c`, `frete.cpp`, `frete.pas`, `frete.java`, `frete.js`, `frete.py2` ou `frete.py3`

O senhor Satoshi passou anos reclamando da empresa de correios do seu país, porque ela sempre transportava suas encomendas usando um caminho que passava pelo número mínimo de cidades entre a cidade onde o senhor Satoshi mora e a cidade destino da encomenda. A empresa alegava que essa estratégia levava ao menor tempo para a entrega final da encomenda. O problema é que ele notou que essa estratégia da empresa nem sempre levava ao menor preço para o frete total. Se ele pudesse escolher o caminho por onde a encomenda deveria passar para ir da sua cidade para a cidade destino, ele poderia economizar bastante com o frete, já que não havia muita urgência para a maioria de suas encomendas.

Depois de muita reclamação, a empresa finalmente está dando aos clientes a opção de determinar o caminho por onde a encomenda deve passar. O senhor Satoshi, feliz da vida, agora quer a sua ajuda para implementar um programa que, dado o custo de transporte de uma encomenda entre vários pares de cidades pelo país, para os quais a empresa realiza entregas diretas, determine qual é o preço total mínimo para o frete entre a cidade onde ele mora e a cidade destino da encomenda.



O país tem  $N$  cidades, identificadas pelos números de 1 a  $N$ . O senhor Satoshi mora na cidade 1 e o destino da encomenda será sempre a cidade  $N$ . É garantido que sempre haverá um caminho de 1 até  $N$ . No exemplo da figura, para  $N = 5$ , o custo mínimo será 7, para o caminho  $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$ .

## Entrada

A primeira linha da entrada contém dois números inteiros  $N$  e  $M$ , representando o número de cidades e quantos pares de cidades possuem entrega direta de encomenda pela empresa. As  $M$  linhas seguintes contêm, cada uma, três inteiros  $A$ ,  $B$  e  $C$ , indicando que a empresa realiza a entrega de uma encomenda diretamente entre as cidades  $A$  e  $B$ , cobrando o preço  $C$ .

## Saída

Seu programa deve imprimir uma linha contendo um inteiro representando o preço mínimo total para o frete entre a cidade onde o senhor Satoshi mora, a cidade 1, e a cidade destino da encomenda, a cidade  $N$ .

## Restrições

- $2 \leq N \leq 100$  e  $1 \leq M \leq 1000$
- $1 \leq A, B \leq N$  e  $A \neq B$
- $1 \leq C \leq 1000$

## Exemplos

Entrada	Saída
5 6 1 2 4 1 3 3 4 3 6 4 5 2 2 4 1 3 5 5	7

Entrada	Saída
7 10 1 2 5 3 1 32 1 4 3 2 3 4 2 6 20 6 3 1 6 4 9 6 5 6 3 7 18 5 7 2	18

## Chuva

Nome do arquivo fonte: `chuva.c`, `chuva.cpp` ou `chuva.pas`

A robótica causou uma grande revolução nos processos industriais no mundo todo; atualmente, vários tipos de robôs são usados na fabricação de carros, equipamentos eletrônicos e até mesmo utensílios domésticos.

Uma fábrica possui um robô de manutenção, que constantemente precisa ser deslocado entre setores diferentes para executar vários serviços. A movimentação do robô é feita por controle remoto: ele pode andar qualquer distância, mas apenas nas quatro direções cardeais (norte, sul, leste e oeste).

Robôs são feitos de metal, e por isso é ideal que eles evitem contato direto com a água. Assim, em dias chuvosos, é ideal que a trajetória do robô passe por dentro de galpões, debaixo de marquises e toldos, etc. para evitar sua exposição à chuva.

A sua tarefa é escrever um programa que, dadas as informações sobre as áreas cobertas e ponto inicial e final do robô, determine uma trajetória para o robô que minimize a porção do trajeto feita sob chuva.

### Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). A primeira linha da entrada contém quatro inteiros  $X_i$ ,  $Y_i$ ,  $X_f$  e  $Y_f$  ( $0 \leq X_i, Y_i, X_f, Y_f \leq 10^6$ ), indicando, respectivamente, a posição atual e a posição final do robô — o robô começa na posição  $(X_i, Y_i)$  e deve terminar na posição  $(X_f, Y_f)$ .

A linha seguinte da entrada contém um único inteiro  $N$  ( $0 \leq N \leq 1000$ ), indicando o número de áreas cobertas na fábrica. Cada uma das  $N$  linhas seguintes contém quatro inteiros  $X_1$ ,  $Y_1$ ,  $X_2$  e  $Y_2$  ( $0 \leq X_1 < X_2 \leq 10^6$ ,  $0 \leq Y_1 < Y_2 \leq 10^6$ ), indicando uma região coberta.

Uma região coberta é um retângulo de lados paralelos aos eixos tal que  $(X_1, Y_1)$  e  $(X_2, Y_2)$  são vértices opostos do retângulo. Duas áreas cobertas podem ter regiões comuns. O robô pode entrar e sair de uma área coberta por qualquer ponto de seu perímetro, e pode trafegar livremente dentro da área coberta.

### Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha, contendo um número inteiro indicando a menor distância que o robô precisa percorrer sob chuva.

Exemplo de entrada	Exemplo de saída
0 0 4 3 0	7

Exemplo de entrada	Exemplo de saída
2 5 5 0 1 0 0 1 5	5

Exemplo de entrada	Exemplo de saída
4 5 5 0 2 0 0 1 5 0 0 3 2	5

### **Informações sobre a pontuação**

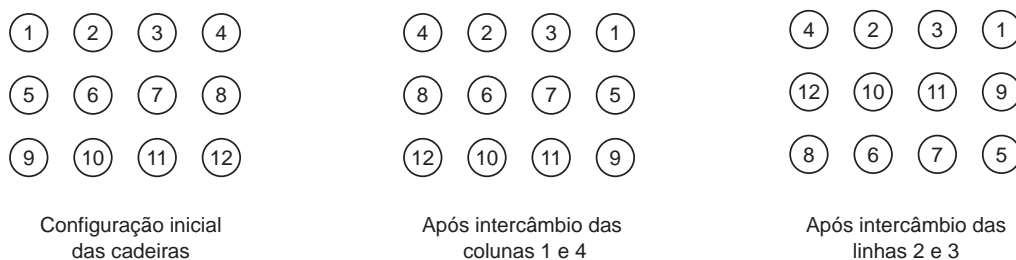
- Em um conjunto de casos de teste que totaliza 30 pontos,  $X_1, Y_1, X_2, Y_2, X_i, X_f, Y_i, Y_f \leq 10$  e  $N \leq 5$ .
- Em um conjunto de casos de teste que totaliza 55 pontos,  $X_1, Y_1, X_2, Y_2, X_i, X_f, Y_i, Y_f \leq 1000$  e  $N \leq 100$ .

# Cadeiras do auditório

Nome do arquivo fonte: `cadeiras.c`, `cadeiras.cpp`, ou `cadeiras.pas`

As cadeiras do auditório da escola são organizadas em um quadriculado com  $L$  linhas e  $C$  colunas. As linhas são numeradas de 1 a  $L$ , as colunas são numeradas de 1 a  $C$ , e as cadeiras são numeradas de 1 a  $L \times C$ , de tal modo que uma cadeira na linha  $i$  coluna  $j$  tem o número  $(i - 1) \times C + j$ .

Durante a aula de teatro, a professora fez com que os alunos executassem uma sequência de mudanças na configuração da sala. Cada uma dessas mudanças intercambiou ou duas colunas ou duas linhas. A figura abaixo ilustra uma configuração original com três linhas e quatro colunas, a posição das cadeiras após uma mudança (intercâmbio das colunas 1 e 4), e a posição das cadeiras após mais uma mudança (intercâmbio das linhas 2 e 3).



Ao final da aula, como era de se esperar, a numeração das cadeiras ficou bem bagunçada. O problema é que a próxima aula é de Matemática, e o professor é muito exigente, e quer começar a aula com as cadeiras perfeitamente posicionadas da maneira original.

## Tarefa

Sua tarefa é escrever um programa que, dada a posição de cada cadeira ao final da aula de teatro, determine qual é a menor sequência de mudanças que devem ser executadas para retornar as cadeiras aos seus devidos lugares, considerando que cada mudança faça o intercâmbio ou de duas linhas ou de duas colunas de cadeiras.

## Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado).

A primeira linha da entrada contém dois números inteiros  $L$  e  $C$ , representando respectivamente o número de linhas e o número de colunas de cadeiras do auditório ( $1 \leq L \leq 200$  e  $1 \leq C \leq 200$ ). Cada uma das  $L$  linhas seguintes contém  $C$  números inteiros entre 1 e  $L \times C$ , separados por um espaço em branco, indicando a posição das cadeiras ao final da aula de teatro. O  $j$ -ésimo número dado na linha  $i$  é o número da cadeira que se encontra na linha  $i$  e coluna  $j$ .

## Saída

Seu programa deve imprimir, na *saída padrão*, na primeira linha um inteiro  $K$  representando o número de mudanças necessárias para retornar as cadeiras para sua posição original. Cada uma das  $K$  linhas seguintes contém a descrição de uma mudança, na forma de um caractere  $M$  (que pode ser 'L' ou 'C'), seguido de um espaço em branco, seguido de um inteiro  $X$ , seguido de um espaço em branco, seguido de um inteiro  $Y$ . Se o caractere descrevendo a mudança é 'L',  $X$  e  $Y$  representam linhas que devem ser intercambiadas; se o caractere descrevendo a mudança é 'C',  $X$  e  $Y$  representam colunas que devem ser intercambiadas.

Para todos os casos testes existe solução com  $K \leq 1000$ . Se mais de uma solução existe com o mesmo número de mudanças, imprima qualquer uma delas.

## Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 30 pontos,  $L \leq 10$  e  $C \leq 10$ .
- Em um conjunto de casos de teste que totaliza 70 pontos,  $L \leq 100$  e  $C \leq 100$ .

## Exemplos

Entrada	Saída
2 2 4 3 2 1	2 L 1 2 C 1 2

Entrada	Saída
3 4 1 2 3 4 5 6 7 8 9 10 11 12	0