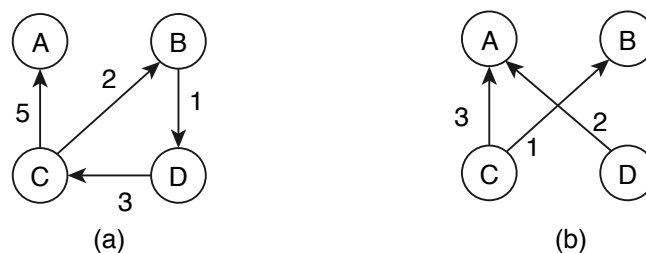


Câmara de Compensação

Nome do arquivo: “compensacao.x”, onde x deve ser `cpp`, `pas`, `java`, `js`, `py2` ou `py3`

Em uma cidade, muitas pessoas emprestam dinheiro para outras pessoas. A coisa chegou a um tal ponto que tem gente que é ao mesmo tempo devedor e credor. As pessoas resolveram então pagar suas dívidas e cada uma emitiu os cheques para pagar suas dívidas. Por exemplo, na figura, item (a), a pessoa C emitiu um cheque de 5 dinheiros para a pessoa A , e a pessoa D emitiu um cheque de 3 dinheiros para a pessoa C . Ou seja, a pessoa C recebeu da pessoa D e pagou a pessoa A . Pior ainda, existe um ciclo vicioso, em que a pessoa D emitiu um cheque de 3 dinheiros para a pessoa C , que por sua vez emitiu um cheque de 2 dinheiros para a pessoa B , que por sua vez emitiu um cheque de 1 dinheiro para a pessoa D . A situação mostrada no item (a) da Figura abaixo é descrita através de uma *lista de cheques*, com quatro triplas da forma (X, V, Y) , para indicar que X emitiu um cheque de V dinheiros para Y . Na mesma Figura, no item (b), a situação é descrita com uma lista de apenas três cheques.



Entretanto, as duas listas são equivalentes: o saldo na conta bancária de uma pessoa é o mesmo em ambas as listas de cheques. Em ambos os casos, completada a compensação de todos os cheques, a pessoa A terminará com 5 dinheiros a mais na sua conta, a pessoa B terminará com 1 dinheiro a mais na sua conta, a pessoa C terminará com 4 dinheiros a menos na sua conta e a pessoa D terminará com 2 dinheiros a menos na sua conta.

Vamos então definir equivalência de listas de cheques emitidos: duas listas de cheques são *equivalentes* se, ao final do processo de compensação de todos os cheques, o seguinte vale para cada pessoa: seu saldo bancário ao final da compensação de uma lista é o mesmo que o saldo bancário da pessoa ao final da compensação da outra lista.

O total de valores compensados no item (a) da figura é igual a 11 dinheiros ao passo que no item (b) o total é de apenas 6 dinheiros!

Este problema tem duas subtarefas:

- *Subtarefa A*: determinar, dada uma lista de cheques, se é possível ou não diminuir o total de valores compensados utilizando uma outra lista de cheques equivalente.
- *Subtarefa B*: determinar o total mínimo de valores compensados em uma lista de cheques equivalente.

Você deve escrever um programa que resolva apenas a Subtarefa A ou que resolva as duas subtarefas.

Entrada

A primeira linha contém dois inteiros, M e N , onde M é o número de cheques emitidos e N é o número de habitantes da cidade. Os habitantes são identificados por números inteiros de 1 a N . Cada uma das M linhas seguintes descreve um cheque da lista e contém três inteiros X, V e Y , que indica que X emitiu um cheque de V dinheiros a favor de Y . É possível que haja mais de um cheque de X a Y . Também é possível que haja cheques de X a Y e de Y a X , mas não de X a X .

Saída

Seu programa deve produzir duas linhas na saída. A primeira linha descreve a resposta para a Subtarefa A e deve conter um único caractere. O caractere deve ser S para indicar que é possível diminuir o total dos cheques compensados com uma lista de cheques equivalente, ou N para indicar que não é possível diminuir o total de cheques compensados.

Se o seu programa resolve também a Subtarefa B, a segunda linha descreve a resposta para essa subtarefa e deve conter um número inteiro, o valor mínimo do total de cheques compensados, em uma lista equivalente. Se o seu programa não resolve a Subtarefa B, você pode deixar a linha em branco ou colocar um valor inteiro arbitrário.

Restrições

- $1 \leq M \leq 10^6$
- $2 \leq N \leq 10^3$
- $1 \leq X \leq N, 1 \leq Y \leq N, X \neq Y$
- $1 \leq V \leq 10^2$

Informações sobre a pontuação

- Subtarefa A: 20 pontos.
- Subtarefa B: em um conjunto de casos de testes que vale 20 pontos $1 \leq N \leq 10$.

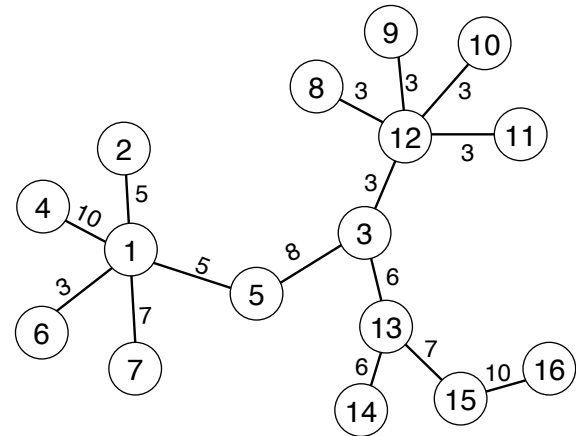
Exemplo de entrada 1 4 4 2 1 4 3 5 1 3 2 2 4 3 3	Exemplo de saída 1 S 6
Exemplo de entrada 2 5 4 4 50 3 2 25 1 3 10 2 2 100 1 4 50 3	Exemplo de saída 2 S 215
Exemplo de entrada 3 4 4 3 10 1 2 40 1 2 30 4 2 20 4	Exemplo de saída 3 N 100

Visita entre cidades

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

A Linearlândia é composta de N cidades, numeradas de 1 até N . Para alguns pares de cidades existe exatamente uma estrada bidirecional entre as duas cidades do par. Os pares de cidades ligados diretamente por uma estrada são escolhidos de forma que sempre é possível ir de qualquer cidade para qualquer outra cidade por um, e somente um, caminho (um caminho é uma sequência de estradas, sem repetição).

Dada a lista de pares de cidades ligados diretamente por estradas, as distâncias entre os pares de cidades, uma cidade origem e uma cidade destino, seu programa deve computar qual a distância entre a cidade de origem e a cidade destino, usando as estradas. Por exemplo, na figura, a distância para ir da cidade 12 para a cidade 7 é 23; a distância da cidade 15 para a cidade 12 é 16; e a distância da cidade 7 para a cidade 15 é 33.



Entrada

A primeira linha da entrada contém três inteiros N , A e B , representando o número de cidades na Linearlândia, a cidade origem e a cidade destino, respectivamente. As cidades são identificadas por inteiros de 1 a N . As $N - 1$ linhas seguintes contém, cada uma, três inteiros P , Q e D , indicando que existe uma estrada ligando diretamente as cidades P e Q , com distância D .

Saída

Seu programa deve imprimir uma linha contendo um inteiro representando a distância para ir de A até B pelas estradas de Linearlândia.

Restrições

- $2 \leq N \leq 10000$
- $1 \leq A \leq N$, $1 \leq B \leq N$, $A \neq B$
- $1 \leq P \leq N$, $1 \leq Q \leq N$
- $1 \leq D \leq 100$

Exemplos

Entrada	Saída
4 2 4 1 2 10 2 3 11 3 4 12	23

Entrada	Saída
16 15 7 3 5 8 12 3 3 5 1 5 2 1 5 4 1 10 6 1 3 7 1 7 12 8 3 12 9 3 12 10 3 12 11 3 3 13 6 13 14 6 15 13 7 15 16 10	33

Castelos da Nlogônia

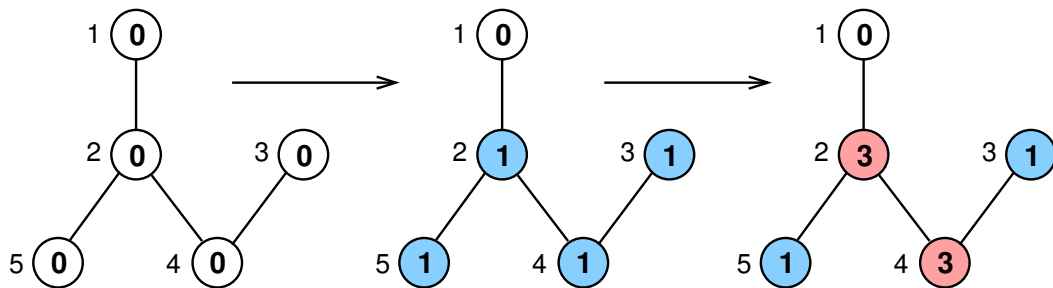
Nome do arquivo: `cores.c`, `cores.cpp`, `cores.pas`, `cores.java`, `cores.js`, `cores.py2` ou `cores.py3`

O rei da Nlogônia não consegue decidir de que cor ele vai mandar pintar os castelos do reino. Nos últimos tempos ele tem dado ordens bastante extravagantes do tipo: “pintem todos os castelos no caminho entre o castelo A e o castelo B, inclusive eles, da cor C”. Ele pode falar “no” caminho, porque os castelos da Nlogônia estão ligados por estradas entre eles de modo que existe exatamente um caminho entre quaisquer dois castelos, possivelmente passando por outros castelos, sem repetir castelos. De outra forma, sempre é possível ir de qualquer castelo para qualquer outro castelo e por apenas um caminho, sem repetir castelos.

A Nlogônia tem N castelos, identificados por números de 1 a N . A figura ilustra uma sequência de duas operações de colorir sobre cinco castelos, numerados de 1 a 5, com cores identificadas por inteiros de 0 a 3:

- colorir os castelos de 5 até 3 com a cor 1;
- colorir os castelos de 2 até 4 com a cor 3.

Ao final, os castelos de 1 a 5 terão as cores 0, 3, 1, 3 e 1, respectivamente.



Neste problema, considerando que os N castelos na Nlogônia inicialmente estão pintados da cor zero, dados os pares de castelos que estão ligados por uma estrada e uma sequência de M ordens de pintura, seu programa deve imprimir a cor que cada castelo vai ter ao final, depois que todas as ordens de pintura forem executadas em sequência.

Entrada

A primeira linha da entrada contém dois inteiros N e M , respectivamente o número de castelos e o número de ordens de pintura. Os castelos são indexados de 1 a N . As $N - 1$ linhas seguintes contêm, cada uma, dois inteiros U e V distintos, indicando que existe uma estrada entre os castelos U e V diretamente. Nas M linhas seguintes, cada linha contém três inteiros P , Q e C , representando uma ordem de pintura entre os castelos P e Q , não necessariamente distintos, com a cor C .

Saída

Seu programa deve imprimir uma única linha, contendo a sequência de cores dos castelos de 1 a N , após todas as M ordens de pinturas terem sido executadas em sequência.

Restrições

- $2 \leq N \leq 100$ e $1 \leq M \leq 100$
- $1 \leq U, V, P, Q \leq N$
- $0 \leq C \leq 100$

Exemplos

Entrada	Saída
5 2 1 2 2 5 2 4 4 3 5 3 1 2 4 3	0 3 1 3 1

Entrada	Saída
9 4 7 1 1 2 6 2 9 6 2 5 8 6 4 5 3 4 1 4 2 7 8 4 3 4 1 9 2 8	4 8 1 1 2 8 4 4 8