

Liga da Justiça

Trinta e cinco anos atrás, um grupo de super-heróis foi escolhido para formar a Liga da Justiça, com o propósito de proteger o planeta Terra dos vilões. Depois de todos esses anos ajudando a humanidade, seus membros estão se aposentando e está na hora de escolher novos membros para a Liga da Justiça.

Para manter sua identidade secreta em segredo, super-heróis normalmente usam um número inteiro para se identificar. Existem H super-heróis na Terra, identificados por números inteiros de 1 até H . Com uma olhada rápida num jornal qualquer um pode descobrir se dois super-heróis já trabalharam juntos numa missão. Se isso aconteceu, dizemos que os dois super-heróis têm um relacionamento.

Só se pode ter uma Liga da Justiça no mundo, que pode ser formada por qualquer número de super-heróis (até mesmo apenas um). Além disso, para quaisquer dois heróis na nova Liga, eles têm que ter um relacionamento.

Considere também o grupo de heróis não escolhidos para entrar na Liga da Justiça. Para quaisquer dois heróis desse grupo, eles não podem ter um relacionamento. Isso previne a formação de outras Ligas da Justiça não oficiais.

Você trabalha para uma agência responsável por criar a nova Liga da Justiça. A agência não sabe se é ou não possível criar a Liga com todas as restrições dadas, e pediu ajuda às suas habilidades em programação. Dado um grupo de super-heróis e seus relacionamentos, determine se é possível selecionar um subgrupo para formar a Liga da Justiça conforme as restrições dadas acima.

Entrada

A entrada é composta por vários casos de teste. A primeira linha de cada caso de teste contém dois inteiros separados por um espaço, H ($2 \leq H \leq 50000$) e R ($1 \leq R \leq 100000$), indicando, respectivamente, o número de super-heróis e o número de relacionamentos. Cada uma das R linhas seguintes contém dois inteiros separados por um espaço, A e B ($1 \leq A < B \leq H$), indicando que o super-herói A tem um relacionamento com o super herói B . Note que se A tem um relacionamento com B , então B tem um relacionamento com A . Um relacionamento nunca é informado duas vezes num caso de teste.

O final da entrada é indicado por $H = R = 0$.

Saída

Para cada caso de teste da entrada imprima uma única linha, contendo a letra maiúscula "Y", caso seja possível selecionar um subgrupo de heróis para formar a Liga da Justiça de acordo com as restrições, ou a letra maiúscula "N", caso contrário.

Exemplo de Entrada	Exemplo de Saída
5 5	Y
1 2	N
2 3	Y
1 3	
1 4	
3 5	
9 8	
1 2	
2 3	
3 4	
4 5	
5 6	
6 7	
7 8	
8 9	
4 3	
1 2	
2 3	
3 4	
0 0	

Desvio de Rua

A prefeitura de uma grande cidade da Nlogônia iniciou um programa de recuperação do asfalto de suas ruas. Na Nlogônia, cada rua liga diretamente dois cruzamentos, e pode ter mão única ou mão dupla. Por determinação de um antigo decreto real, sempre existe ao menos um caminho entre dois pontos quaisquer da cidade.

No programa de recuperação, uma única rua será recuperada por vez, e para isso a rua será fechada para o tráfego. Esse fechamento pode causar caos no trânsito local ao violar o decreto real, impedindo vários cidadãos de voltarem para casa dos seus trabalhos e vice-versa. A prefeitura pode converter algumas das ruas de mão única em mão dupla, mas prefere evitá-lo pois ruas de mão dupla tendem a causar acidentes mais graves; a prefeitura prefere criar desvios apenas invertendo as mãos das ruas de mão única já existentes.

O Rei da Nlogônia solicitou seus préstimos para escrever um programa que, dada a descrição das ruas de uma cidade, determine se, quando uma dada rua é interditada para recuperação, continua existindo um caminho entre quaisquer dois pontos da cidade, mesmo que seja necessário alterar as mãos de direção de outras ruas.

Entrada

A entrada é composta por diversos casos de teste. A primeira linha de um caso de teste contém dois inteiros N ($1 \leq N \leq 10^3$) e M ($1 \leq M \leq 10^3$), representando respectivamente o número de cruzamentos e o número de ruas da cidade. Os cruzamentos são identificados por inteiros de 1 a N e as ruas são identificadas por números inteiros de 1 a M . Cada uma das M linhas seguintes descreve uma rua e contém três inteiros A ($1 \leq A$), B ($B \leq N$) e T ($1 \leq T \leq 2$), onde A e B são os cruzamentos que a rua liga diretamente, e T indica a mão de direção da rua: se $T = 1$ a rua tem mão única na direção de A para B ; se $T = 2$ a rua tem mão dupla. A primeira rua descrita será interditada para recuperação.

Saída

Para cada caso de teste seu programa deve imprimir uma linha contendo um caractere que descreve o que a prefeitura deve fazer para respeitar o decreto real após o fechamento da rua para reformas:

- '-': não é necessário qualquer tipo de alteração nas outras ruas.
- '*': é impossível respeitar o decreto real, independente de quaisquer mudanças nas outras ruas.
- '1': é possível cumprir o decreto real apenas invertendo as mãos de algumas das ruas de mão única.
- '2': é possível cumprir o decreto real, mas é necessário converter algumas ruas de mão única para mão dupla.

Exemplo de Entrada	Exemplo de Saída
3 3	-
1 2 2	2
2 3 2	*
3 1 2	1
3 3	
1 2 1	
2 3 1	
3 1 1	
2 1	
1 2 2	
5 6	
3 4 1	
1 3 2	
2 4 2	
3 5 2	
2 1 1	
4 1 1	

O País das Bicicletas

Como você já deve saber, a bicicleta é um dos meios de transportes mais populares da China. Quase todos os chineses possuem a sua, e utilizam-na para trabalho, recreação, e outras atividades.

Após muitos anos pedalando, Mr. Lee já não têm a mesma disposição para encarar as diversas subidas da cidade onde mora. E a cidade em que Mr. Lee vive é extremamente montanhosa. Por razões sentimentais, ele não quer mudar para uma cidade mais plana. Resolveu, então, que tentaria evitar grandes altitudes em seus caminhos mesmo que, para isso, tivesse que pedalar um pouco mais.

Mr. Lee obteve com o serviço topográfico chinês uma coleção de mapas de sua cidade, em que cada rua desses mapas possui a informação da maior altitude encontrada quando trafegada. Tudo que ele precisa fazer agora é determinar rotas que minimizem a altura percorrida entre pares (origem, destino).

Sabendo que você planeja visitar a cidade em que ele mora no próximo ano, Mr. Lee pediu sua ajuda. Em uma primeira etapa, ele deseja que você implemente um programa que receba mapas topográficos da cidade e uma coleção de pares (origem, destino). Para cada par, seu programa deve imprimir a maior altura encontrada em uma rota entre a origem e o destino. Lembre-se que as rotas devem minimizar tais alturas. As rotas propriamente ditas, serão determinadas em uma segunda etapa (quando você chegar à China para visitá-lo).

Como o transporte utilizado é uma bicicleta, você pode considerar que todas as ruas da cidade são de mão dupla. Não demore, pois Mr. Lee conta com você. :-)

Entrada

Seu programa deve estar preparado para trabalhar com diversos mapas, doravante denominados instâncias. Cada instância tem a estrutura que segue.

Na primeira linha são fornecidos dois inteiros n ($0 \leq n \leq 100$) e m ($0 \leq m \leq 4950$) que representam, respectivamente, os números de interseções e de ruas. Por razões de clareza, as interseções são numeradas de 1 a n , inclusive; toda rua começa e termina em uma interseção; e não existem interseções fora das extremidades de uma rua.

Nas próximas m linhas são fornecidos três inteiros: i e j ($1 \leq i, j \leq n$) que indicam a existência de uma rua entre as interseções i e j ; e h que representa a maior altitude encontrada quando a rua é trafegada. Esses inteiros estão separados por espaços em branco.

Na linha seguinte, é dado um inteiro k ($1 \leq k \leq 50$) que representa o número de pares (origem, destino) que serão especificados nas próximas k linhas. Cada par é formado por dois inteiros i e j como acima. Isto é, origem e destino são interseções de ruas, e também estão separados por espaços em branco.

Valores $n = m = 0$ indicam o final das instâncias e não devem ser processados.

Saída

Para cada instância solucionada, você deverá imprimir um identificador **Instancia** h em que h é um número inteiro, sequencial e crescente a partir de 1. Nas próximas k linhas, você deve imprimir as maiores alturas encontradas nas rotas entre os k pares (origem, destino) fornecidos, um valor por linha, na ordem da entrada.

Uma linha em branco deve ser impressa após cada instância.

Exemplo de Entrada	Exemplo de Saída
12 17 1 4 4 4 7 6 7 10 6 2 5 4 5 8 5 8 11 2 3 6 5 6 9 3 9 12 1 1 2 1 2 3 9 4 5 3 5 6 7 7 8 7 8 9 2 10 11 1 11 12 2 4 1 5 6 8 6 7 11 10 0 0	Instancia 1 4 3 6 1

Max, o Louco

No ano de 2042, após o surgimento da malévola União das Repúblicas Independentes (URI), a humanidade se deparou com uma grande escassez de recursos. Água e gasolina se tornaram bens muito valiosos, sendo que boa parte da tecnologia se perdeu após a URI tomar o poder mundial.

Você faz parte de um grupo da resistência, que tem o objetivo de tirar o poder da URI. Max, o herói da resistência, precisa realizar várias missões que envolvem viagens de carro entre cidades. Existem postos de gasolina em cada cidade, apesar dos altos e variados preços. Como os recursos financeiros da resistência são limitados, você foi convocado a escrever um programa que calcule qual a quantidade mínima de créditos da união necessários para completar cada uma das missões de Max.

Entrada

A entrada é composta por vários casos de teste. Cada caso de teste é iniciado por três inteiros, **N**, **M** e **T**, ($1 \leq N \leq 10$, $1 \leq M \leq 20$, $1 \leq T \leq 50$) correspondentes ao número de cidades na rota, o número de estradas e a capacidade do tanque do carro de Max, em litros. A entrada acaba quando **N** = **M** = **T** = 0.

As **M** linhas na sequência descrevem as ligações entre as cidades. Cada linha contém os inteiros **A**, **B** e **C**, ($1 \leq C \leq 1000$) que indicam a existência de uma rota (ida e volta) entre as cidades **A** e **B**, com um consumo de **C** litros de gasolina. Devido ao estado precário das estradas, é possível que determinadas cidades sejam inacessíveis. Não existe mais de uma rota direta entre qualquer par de cidades.

As próximas **N** linhas descrevem o custo, em créditos da união por litro, da gasolina em cada cidade. A primeira linha descreve o custo da gasolina na primeira cidade, a segunda linha descreve o custo na segunda cidade, e assim por diante.

Saída

Para cada caso de teste, seu programa deverá imprimir uma linha contendo o menor custo possível para viajar da cidade **1** até a cidade **N**. Caso não for possível viajar entre as cidades, imprima **-1**.

Exemplo de Entrada	Exemplo de Saída
3 2 50 1 2 30 2 3 30 1 1 1 3 1 10 1 2 20 1 1 1 0 0 0	10 -1