

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

Salas Separadas

Luiz é um coordenador muito dedicado do curso de ciência da computação. Ele sabe a importância de incentivar seus alunos a esforçarem-se em algoritmos, por isso decidiu realizar uma competição de programação individual entre estes, com direito a premiação.

Luiz gostaria de evitar ao máximo a possibilidade de fraude na competição, e por isso não gostaria que houvessem dois candidatos amigos na mesma sala durante a realização da prova.

Infelizmente Luiz só dispõe de duas salas para a realização da prova, mas felizmente conhece muito bem seus alunos, e sabe dizer com facilidade quem é amigo de quem. Entretanto, ele precisa saber se será possível, apenas com as duas salas disponíveis, combinar os alunos da forma que deseja e para isso solicitou a você, aluno dedicado do curso, que crie um programa para ajudá-lo.

Entrada

A entrada é composta de diversos casos de teste. A primeira linha de cada caso de teste consiste em um inteiro N ($2 \leq N \leq 100$) indicando o número de alunos que irão realizar a prova. Cada N par de linhas seguintes descreve as relações de amizade de cada participante, de forma que a primeira linha consiste no identificador do participante, e a linha seguinte consiste em uma lista descrevendo uma quantidade M de alunos ($1 \leq M < N$) com os quais aquele participante possui uma relação de amizade.

Considere que não é relevante o número de pessoas dispostas em cada sala, e que, se existe uma relação de amizade entre alunos x e y , existe uma relação de amizade entre y e x . A entrada termina quando $N = 0$, e não deve ser processada.

Saída

Para cada caso de teste, deverá ser impressa uma linha contendo a resposta "SIM", caso seja possível dispor os alunos de forma que não hajam dois amigos realizando a prova na mesma sala, e "NAO", caso contrário.

Exemplo de Entrada	Exemplo de Saída
6	SIM
1	SIM
2 3	NAO
2	
1 4	
3	
1 6	
4	
2 5	
5	
4 6	
6	
3 5	
4	
1	

Exemplo de Entrada	Exemplo de Saída
2 3 4 2 1 3 1 4 1 4 1 2 4 2 4 3 4 4 1 2 3 0	