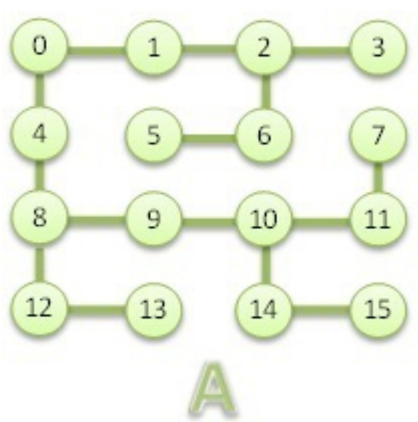
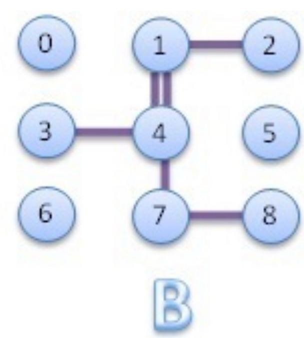
Desenhando Labirintos

Pedro gosta muito de desenhar labirintos, e teve uma ideia recentemente: quantos movimentos com a caneta ele precisa fazer, no mínimo, para desenhar um labirinto, saindo sempre da mesma posição e finalizando no mesmo ponto? Para a brincadeira ficar interessante, Pedro decidiu que não é permitido levantar a caneta do papel. Os modelos para construção do labirinto são sempre quadrados, ou seja, 3 x 3 , 4 x 4 , 5 x 5 até no máximo de 7 x 7.

Para cada exemplo, Pedro vai especificar de onde o desenho deve começar e é tarefa sua determinar a quantidade de movimentos necessários para desenhar o labirinto como Pedro deseja. Pedro ainda lembra que você não precisa se preocupar com ciclos, pois não haverá nenhum ciclo em nenhum dos casos de teste. Se existir uma entrada 4 1, isso não impede a existência de outra entrada 1 4 no mesmo caso de teste, ou seja, outra linha ligando estes mesmos dois nodos. De qualquer forma isto não fará diferença no desenho do labirinto, pois se Pedro traçar as duas linhas entre 1 e 4 ou apenas uma delas, a quantidade de movimentos deverá ser a mesma. Somente neste caso, a utilização da segunda linha por Pedro é opcional.

Veja os exemplos abaixo, no labirinto A (4 x 4), Pedro deseja sair do nodo 0, desenhar todas as linhas e retornar ao nodo 0. Para isso, o mínimo de movimentos possíveis é 30. No labirinto B (3 x 3), Pedro deseja sair do nodo 1, desenhar todas as linhas e retornar para o nodo 1. Neste caso, ele precisa de 10 movimentos para fazer este desenho.





Entrada

A primeira linha de entrada é um inteiro **T** (**T** < 100) que indica o número total de casos de teste. Cada caso inicia com uma linha contendo um inteiro **N** (**N** < **X**^{2}, onde **X** é a largura em nodos do labirinto, que pode variar de 3 até 7). Este **N** é o ponto (nodo) no qual o desenho deve ser iniciado e também é onde o desenho deve ser terminado. Na próxima linha há duas informações **V** e **A** que são respectivamente a quantidade de vértices e arestas do desenho. Uma quantidade **A** de linhas vem a seguir, cada uma descrevendo um segmento de linha que Pedro tem disponível para desenhar o labirinto desejado.

Saída

O arquivo de saída contém um valor para cada caso de teste de entrada. Este valor é a quantidade de movimentos de caneta que devem ser feitos para desenhar o labirinto do caso de teste, considerando que o início e o fim são sempre a partir do mesmo ponto (nodo) e que não é possível levantar a caneta do papel.

Exemplo de Entrada	Exemplo de Saída
2 0 16 15 0 4 2 3 6 2 8 9 10 9 8 12 14 15 14 10 6 5 10 11 11 7 4 8 0 1 1 2 12 13 1 9 6 1 2 1 4 4 7 7 8 4 1 4 3	30 10

Colônia de Formigas

Um grupo de formigas está muito orgulhoso pois construíram uma grande e magnífica colônia. No entanto, seu enorme tamanho tem se tornado um problema, pois muitas formigas não sabem o caminho entre algumas partes da colônia. Elas precisam de sua ajuda desesperadamente!

A colônia de formigas foi criada como uma série de **N** formigueiros conectados por túneis. As formigas, obssessivas como são, numeraram os formigueiros sequencialmente à medida que os construiam. O primeiro formigueiro, numerado 0, não necessitava nenhum túnel, mas para cada um dos formigueiros subsequentes, 1 até **N-1**, as formigas também construíram um único túnel que conectava o novo formigueiro a um dos formigueiros existentes. Certamente, esse túnel era suficiente para permitir que qualquer formiga visitasse qualquer formigueiro já construído, possivelmente passando através de outros formigueiros pelo percurso, portanto elas não se preocupavam em fazer novos túneis e continuavam construindo mais formigueiros.

O seu trabalho é: dada a estrutura de uma colônia e um conjunto de consultas, calcular, para cada uma das consultas, o menor caminho entre pares de formigueiros. O comprimento do caminho é a soma dos comprimentos de todos os túneis que necessitam ser visitados.

Entrada

Cada caso de teste se estende por várias linhas. A primeira linha contém um inteiro **N** representando a quantidade de formigueiros na colônia ($2 \leq \mathbf{N} \leq 10^5$). Cada uma das próximas **N-1** linhas contém dois inteiros que descrevem um túnel. A linha *i*, para $1 \leq i \leq$ **N-1**, contém $\mathbf{A}_i$ e $\mathbf{L}_i$, indicando que o formigueiro *i* foi conectado diretamente ao formigueiro $\mathbf{A}_i$ por um túnel de comprimento $\mathbf{L}_i$ ($0 \leq \mathbf{A}_i \leq \mathbf{i-1}$ e $1 \leq \mathbf{L}_i \leq 10^9$). A próxima linha contém um inteiro **Q** representando o número de consultas que seguem ($1 \leq \mathbf{Q} \leq 10^5$). Cada uma das **Q** linhas seguintes descreve uma consulta e contém dois inteiros distintos **S** e **T** ($0 \leq \mathbf{S},\mathbf{T} \leq \mathbf{N-1}$), representando, respectivamente, os formigueiros de origem e destino.

O último caso de teste é seguido por uma linha contendo apenas um zero.

Saída

Para cada caso de teste, imprima uma única linha com **Q** inteiros, os comprimentos do menor caminho entre os dois formigueiros de cada consulta. Escreva os resultados para cada consulta na mesma ordem em que aparecem na entrada.

<table>
<tr><th>Exemplo de Entrada</th><th>Exemplo de Saída</th></tr>
<tr><td><pre>6
0 8
1 7
1 9
0 3
4 2
4
2 3
5 2
1 4
0 3
2
0 1
2
1 0
0 1
6
0 1000000000
1 1000000000
2 1000000000
3 1000000000
4 1000000000
1
5 0
0</pre></td><td><pre>16 20 11 17
1 1
5000000000</pre></td></tr>
</table>

Escolhendo as Duplas

Uma professora quer dividir todos os alunos de uma sala em duplas para a realização de um trabalho em grupo. Nessas horas, há muita briga entre os alunos para escolher as duplas, porque muitos alunos querem fazer dupla com os melhores alunos da sala.

A professora decidiu dessa vez escolher as duplas de uma forma diferente. Cada aluno poderá dizer à professora um outro aluno com o qual queira fazer uma dupla. Após isso, a professora escolherá as duplas de forma com que todas as duplas satisfaçam pelo menos o desejo de um dos alunos da dupla.

Agora acho que você já sabe qual será seu trabalho neste problema. Dada a lista de desejos dos alunos, imprima a lista de duplas que a professora deve escolher.

Entrada

A entrada é composta por vários casos de teste. Cada caso de teste é composto por duas linhas. A primeira linha de um caso de teste contém um inteiro **N** (2 ≤ **N** ≤ 10000) igual ao número de alunos da sala de aula. A segunda linha contém os desejos de todos os alunos em ordem (a pessoa escolhida pelo aluno 1, pelo aluno 2, assim por diante). Nenhum aluno irá escolher a si próprio.

Saída

Para cada teste, a saída é composta por uma linha. Caso seja impossível formar as duplas do jeito que a professora quer, imprima "IMPOSSIBLE". Caso haja solução, imprima em ordem os parceiros de cada aluno (o parceiro do aluno 1, do aluno 2, assim por diante). Caso haja mais de uma solução, deve se priorizar o desejo dos alunos de menor índice, ou seja, sempre que possível deve-se atender o desejo do aluno 1, depois tentar atender o desejo do aluno 2, e assim por diante. Lembre-se que o problema pede para formar duplas, se o parceiro do aluno **X** é igual a **Y**, o parceiro do aluno **Y** deve ser igual a **X**.

No último caso de entrada do exemplo, os pares são (1,3), respeitando o desejo do aluno 1, e também (2,4), respeitando o desejo do aluno 4.

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

Ir e Vir

Numa certa cidade há **N** intersecções ligadas por ruas de mão única e ruas com mão dupla de direção. É uma cidade moderna, de forma que muitas ruas atravessam túneis ou têm viadutos. Evidentemente é necessário que se possa viajar entre quaisquer duas intersecções, isto é, dadas duas intersecções **V** e **W**, deve ser possível viajar de **V** para **W** e de **W** para **V**.

Sua tarefa é escrever um programa que leia a descrição do sistema de tráfego de uma cidade e determine se o requisito de conexidade é satisfeito ou não.

Entrada

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém dois números inteiros **N** e **M**, separados por um espaço em branco, indicando respectivamente o número de intersecções ($2 \le \mathbf{N} \le 2000$) e o número de ruas ($2 \le \mathbf{M} \le \mathbf{N}(\mathbf{N}-1)/2$). O caso de teste tem ainda mais **M** linhas, que contêm, cada uma, uma descrição de cada uma das **M** ruas. A descrição consiste de três inteiros **V**, **W** e **P**, separados por um espaço em branco, onde **V** e **W** são identificadores distintos de intersecções ($1 \le \mathbf{V}, \mathbf{W} \le \mathbf{N}$, $\mathbf{V} \ne \mathbf{W}$) e **P** pode ser 1 ou 2; se **P**= 1 então a rua é de mão única, e vai de **V** para **W**; se **P** = 2 então a rua é de mão dupla, liga **V** e **W**. Não existe duas ruas ligando as mesmas intersecções.

O ultimo caso de teste é seguido por uma linha que contém apenas dois números zero separados por um espaço em branco.

Saída

Para cada caso de teste seu programa deve imprimir uma linha contendo um inteiro G, onde G é igual a 1 se o requisito de conexidade está satisfeito, ou G é igual a 0, caso contrário.

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

Labirintos de Cerca Viva

A Rainha da Nlogônia é uma fã de labirintos, e então os arquitetos do reino construíram vários labirintos em volta do palácio da Rainha. Todo labirinto construído para a Rainha é feito de salas conectadas por corredores. Cada corredor conecta um par diferente de salas distintas e pode ser atravessado em ambas as direções.

A Rainha ama passear pelas salas e corredores do labirinto nos finais de tarde. Seus serventes escolhem um desafio diferente todo dia, que consiste em encontrar um caminho simples de uma sala inicial até uma sala final no labirinto. Um caminho simples é uma sequência de salas distintas tal que cada par de salas consecutivas é conectado por um corredor. Neste caso a primeira sala da sequência deve ser a sala inicial, e a última sala deve ser a sala final. A Rainha acha que um desafio é bom quando, dentre as rotas da sala inicial até a sala final, exatamente uma delas é um caminho simples. Você pode ajudar os serventes da Rainha a escolher um desafio que agrada a Rainha?

Para tal, escreva um programa que dados a descrição de um labirinto e uma lista de consultas definindo a sala inicial e a sala final, determina para cada consulta se aquela escolha é um bom desafio ou não.

Entrada

Cada caso de teste é descrito usando várias linhas. A primeira linha contém três inteiros **R**, **C** e **Q** representando respectivamente o número de salas do labirinto ($2 \le \mathbf{R} \le 10^4$), o número de corredores ($1 \le \mathbf{C} \le 10^5$), e o número de consultas ($1 \le \mathbf{Q} \le 1000$). As salas são identificadas por inteiros de 1 até **R**. Cada uma das próximas **C** linhas descreve um corredor usando dois inteiros distintos **A** e **B**, indicando que existe um corredor conectando as salas **A**e **B** ($1 \le \mathbf{A}, \mathbf{B} \le \mathbf{R}$). Após isso, cada uma das próximas **Q** linhas descreve uma consulta usando dois inteiros distintos **S** e **T** indicando respectivamente as salas inicial e final do desafio ($1 \le \mathbf{S}, \mathbf{T} \le \mathbf{R}$). Você pode assumir que em cada caso de teste existe no máximo um corredor conectando cada par de salas, e não haverá duas consultas iguais.

O último caso de teste será seguido por uma linha contendo três zeros.

Saída

Para cada caso de teste imprima **Q** + 1 linhas. Na i-ésima linha escreva a resposta para a i-ésima consulta. Se as salas formam um bom desafio, então escreva o caractere 'Y' (maiúsculo). Caso contrário escreva o caractere 'N' (maiúsculo). Imprima uma linha contendo um único caractere '-' (hífen) depois de cada caso de teste.

Exemplo de Entrada	Exemplo de Saída
6 5 3 1 2 2 3 2 4 2 5 4 5 1 3 1 5 2 6 4 2 3 1 2 2 3 1 4 1 3 1 2 0 0 0	Y N N - N Y Y -

Dona Minhoca

Dona Minhoca fica furiosa quando ouve as pessoas dizerem que minhocas são bichos palíndromes, nos quais não é possível distinguir a cabeça do rabo. Que infâmia!

Dona Minhoca vive em uma linda caverna, composta de salões e túneis. Cada túnel liga dois salões distintos e pode ser usado nas duas direções. Um "ciclo" na caverna é uma sequência de salões $s_1, s_2, \ldots, s_n, s_{n+1} = s_1$, tais que $s_i \neq s_{i+1}$ e (s_i, s_{i+1}) é um túnel, para $1 \leq i \leq n$. A caverna de Dona Minhoca pode conter ciclos, mas cada salão faz parte de no máximo um ciclo da caverna. Os túneis e salões são estreitos, de forma que se uma parte do corpo de Dona Minhoca ocupa um túnel ou salão, não há espaço para Dona Minhoca entrar novamente por esse túnel ou salão.

Alguns salões da caverna têm acesso a partir da superfície. Dona Minhoca tem um mapa que descreve a caverna, informando para cada túnel o seu comprimento e quais dois salões o túnel liga. Dona Minhoca também é vaidosa e conhece o seu próprio comprimento.

Dona Minhoca quer saber, para os salões que têm acesso à superfície, se é possível entrar na caverna pelo salão, percorrer a menor distância possível dentro da caverna, e sair novamente pelo mesmo salão que entrou, sempre andando para a frente, sem nunca dar marcha-a-ré. Você pode ajudá-la?

Entrada

A primeira linha contém dois inteiros **S** (**$2 \leq S \leq 10^4$**) e **T** (**$1 \leq T \leq 2S$**) representando respectivamente o número de salões e o número de túneis da caverna. Os salões são identificados por inteiros de **1** a **S**. Cada uma das **T** linhas seguintes descreve um túnel e contém três inteiros **A**, **B** e **C** (**$1 \leq A < B \leq S$**; **$1 \leq C \leq 100$**), onde **A** e **B** representam os salões ligados pelo túnel, e **C** representa o comprimento do túnel. Um salão é ligado por túneis a no máximo outros 100 salões e cada dois salões são ligados por no máximo um túnel. A próxima linha contém um inteiro **Q** (**$1 \leq Q \leq 100$**), que indica o número de consultas. Cada uma das **Q** linhas seguintes descreve uma consulta, e contém dois inteiros **X** (**$1 \leq X \leq S$**) e **M** (**$1 \leq M \leq 10^5$**), que indicam respectivamente o salão pelo qual Dona Minhoca quer entrar e o comprimento de Dona Minhoca.

Saída

Para cada consulta da entrada seu programa deve produzir apenas uma linha, contendo apenas um número inteiro, o comprimento do percurso mínimo que Dona Minhoca deve percorrer dentro da caverna para entrar e sair pelo salão indicado na consulta, sem dar marcha-a-ré. Se não for possível para Dona Minhoca entrar e sair sem dar marcha-a-ré, a linha deve conter o valor −1.

Exemplo de Entrada	Exemplo de Saída
4 4 1 2 12 2 3 10 3 4 8 2 4 5 3 1 23 4 10 1 24 8 9 1 2 1 2 3 1 3 4 1 2 5 10 5 6 25 2 6 20 3 7 9 7 8 3 3 8 4 4 1 10 4 60 8 5 7 55	47 23 -1 20 -1 16 71