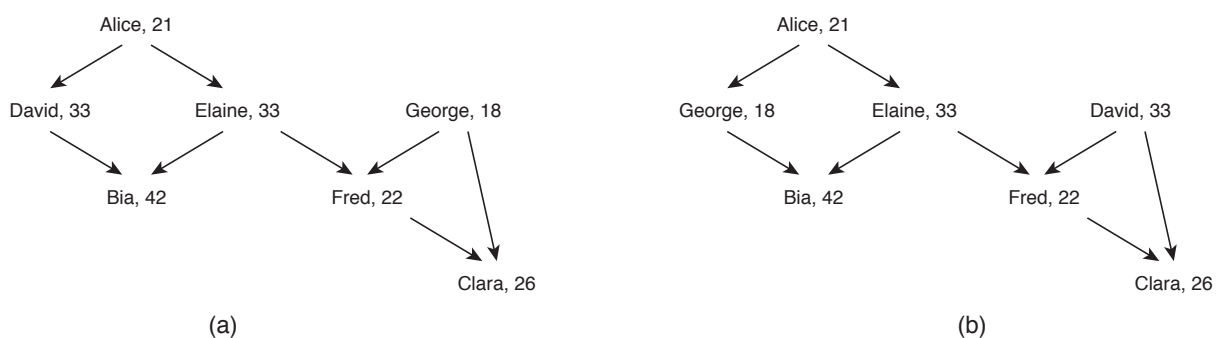


O Chefe

Nome do arquivo: `chefe.c`, `chefe.cpp`, `chefe.pas`, `chefe.java`, `chefe.js` ou `chefe.py`

Todos conhecem Iks, a última moda em redes sociais, que fez tanto sucesso que competidores como Facebook e Google+ estão começando a ter dificuldades financeiras. Assim como muitas companhias “.com”, Iks surgiu em uma pequena garagem, mas hoje emprega milhares de pessoas no mundo todo.

O sistema de gerência utilizado em Iks é bem diferente do padrão. Por exemplo, não há diretorias ou superintendências. No entanto, como é usual em outras companhias, há uma cadeia (ou melhor, várias cadeias) de comando: uma pessoa pode gerenciar outras pessoas, e pode ser gerenciada por outras pessoas. As figuras abaixo mostram a cadeia de comando para alguns empregados, junto com suas idades.



Uma pessoa P_1 pode gerenciar outra pessoa P_2 diretamente (quando P_1 é o superior imediato de P_2) ou indiretamente (quando P_1 gerencia diretamente uma pessoa P_3 que gerencia P_2 direta ou indiretamente). Por exemplo, na figura (a) acima, Alice gerencia David diretamente e Clara indiretamente. Uma pessoa não gerencia a si própria, nem direta nem indiretamente.

Um folclore que apareceu em Wall Street é que Iks é tão bem sucedido porque em sua rede de comando um(a) gerente é sempre mais jovem do que as pessoas que ele(a) gerencia. Como podemos ver na figura acima, isso não é verdade. Mas esse folclore incentivou Iks a desenvolver uma ferramenta para analisar o seu sistema de gerenciamento, e estudar se tem alguma influência no sucesso da empresa. Você foi contratado para trabalhar nessa ferramenta.

Dadas a descrição da cadeia de comando na Iks e as idades de seus empregados, escreva um programa que execute uma série de instruções. Instruções podem ser de dois tipos: trocas de gerência e perguntas. Uma instrução de troca de gerência faz dois empregados A e B trocarem suas posições na cadeia de comando. Como exemplo, a figura (b) acima mostra a cadeia de comando resultante quando David e George trocam suas respectivas posições na cadeia de comando. Uma instrução de pergunta identifica um empregado A e deseja saber a idade do mais jovem gerente (direto ou indireto) de A na cadeia de comando. Por exemplo, no cenário da figura (a) acima a idade do(a) gerente mais jovem de Clara é 18 anos; já no cenário da figura (b), a idade do(a) gerente mais jovem de Clara é 21 anos.

Entrada

A entrada é composta de várias linhas. A primeira linha contém três inteiros N , M e I , indicando respectivamente o número de empregados, o número de relações de gerência direta e o número de instruções. Empregados são identificados por números de 1 a N . A segunda linha contém N inteiros K_i , onde K_i indica a idade do empregado de número i .

Cada uma das M linhas seguintes contém dois inteiros X e Y , indicando que X gerencia Y diretamente. Seguem-se I linhas, cada uma descrevendo uma instrução. Uma instrução de troca de gerência é descrita em uma linha contendo o identificador T seguido de dois inteiros A e B , indicando os dois empregados que devem trocar seus lugares na cadeia de comando. Uma instrução de pergunta é descrita em uma linha contendo o identificador P seguido de um inteiro E , indicando um empregado. A última instrução será sempre do tipo pergunta.

Saída

Para cada instrução de pergunta seu programa deve imprimir uma linha contendo um único inteiro, a idade da pessoa mais jovem que gerencia (direta ou indiretamente) o empregado nomeado na pergunta. Se o empregado nomeado não possui um gerente, imprima o caractere ‘*’ (asterisco).

Restrições

- $1 \leq N \leq 500$
- $0 \leq M \leq 60 \times 10^3$
- $1 \leq I \leq 500$
- $1 \leq K_i \leq 100$, para $1 \leq i \leq N$
- $1 \leq X, Y \leq N$, $X \neq Y$
- $1 \leq A, B \leq N$
- $1 \leq E \leq N$

Exemplos

Entrada	Saída
7 8 9	18
21 33 33 18 42 22 26	21
1 2	18
1 3	18
2 5	*
3 5	26
3 6	
4 6	
4 7	
6 7	
P 7	
T 4 2	
P 7	
P 5	
T 1 4	
P 7	
T 4 7	
P 2	
P 6	

Entrada	Saída
6 5 6	*
10 20 30 40 50 60	10
1 5	30
1 4	30
3 6	60
2 5	
4 5	
P 1	
P 5	
P 6	
T 1 6	
P 1	
P 4	

Toca do Saci

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

Depois de muito procurar, Emília finalmente conseguiu encontrar a toca do Saci. A toca tem formato retangular, e é formada por um quadriculado de salas quadradas de mesmo tamanho, com N salas em uma dimensão e M salas na outra dimensão. A figura abaixo mostra um exemplo de mapa da toca, com cinco salas na dimensão horizontal e quatro salas na dimensão vertical. Há uma única entrada, pela sala marcada com o número 3 no mapa. As salas da toca são muito parecidas, para confundir quem tenta encontrar o Saci, e têm portas que comunicam-se apenas com salas vizinhas nas direções horizontal e vertical do mapa.

0	1	1	1	0
0	2	0	1	1
0	0	0	0	1
3	1	1	1	1

Emília entrou na toca seguindo o Saci com o objetivo de pegar o seu chapéu, e só vai devolvê-lo se o Saci prometer não fazer mais diabrites no Sítio. Muito esperta, ela foi deixando estrelinhas coloridas pelas salas que passou (marcadas com o número 1 no mapa), para saber o caminho de volta. Ela pegou o chapéu do Saci enquanto ele dormia, e começou o caminho de volta. Está muito escuro e ela precisa acender um fósforo em cada sala, para ver as estrelinhas que marcam o caminho. No meio do caminho, ela percebeu que seus fósforos estavam acabando e agora está com medo de não ter fósforos suficientes. Ela está na sala marcada com o número 2 no mapa. Você pode ajudá-la?

Dado o mapa da toca, como no exemplo acima, escreva um programa para saber por quantas salas Emília deve passar até encontrar a saída.

Entrada

A primeira linha da entrada contém dois inteiros N e M que indicam respectivamente os números de salas nas duas dimensões da toca. Cada uma das N linhas seguintes contém M números inteiros entre 0 e 3. O valor 0 indica uma sala sem estrelinhas; o valor 1 indica uma sala com estrelinhas deixadas por Emília; o valor 2 indica uma sala com estrelinhas que é a sala onde Emília está; finalmente, o valor 3 indica uma sala com estrelinhas que é a saída. Considere que, durante o trajeto da entrada até a sala marcada com o valor 2, Emília não passou mais do que uma vez por uma mesma sala, e não existe ambiguidade no caminho de volta (em outras palavras, a cada ponto do trajeto de volta, existe apenas uma sala marcada para Emília voltar).

Saída

Seu programa deve imprimir uma única linha, contendo o número de salas que Emília deve passar, seguindo as estrelinhas, até chegar à saída da toca.

Restrições

A entrada obedece às seguintes restrições:

- $1 \leq N \leq 1000$

- $1 \leq M \leq 1000$
- cada sala tem o valor 0, 1, 2 ou 3.
- apenas uma sala tem o valor 2.
- apenas uma sala tem o valor 3.

Informações sobre a pontuação

- Em um conjunto de casos de teste equivalente a 20 pontos, Emília está em uma sala que só possui uma sala vizinha com estrelinhas (como no exemplo 1).

Exemplos

Entrada	Saída
4 5 0 1 1 1 0 0 2 0 1 1 0 0 0 0 1 3 1 1 1 1	12

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

Entrada	Saída
4 5 0 1 2 1 0 0 1 0 1 1 0 0 0 0 1 3 1 1 1 1	10

Uiquipédia

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

A Uiquipédia (Wikipedia em inglês), fundada em 2001 por Jimmy Wales e Larry Sanger, é um site onde qualquer pessoa pode editar os artigos, fazendo correções ou ampliando seu conteúdo.

Uma das grandes vantagens da Uiquipédia sobre enciclopédias de papel é a facilidade de seguir **referências**; com um simples clique, é possível ir de um artigo para outro relacionado. Essas referências são chamadas de referências diretas. Também é possível navegar a Uiquipédia sequencialmente: cada artigo possui referência para o artigo anterior e para o posterior, na ordem alfabética. Essas referências são chamadas de referências sequenciais.

Por exemplo, um artigo para o termo “Elefante” pode ter uma referência direta para “Mamíferos” em seu texto, desta forma pode-se chegar de “Elefante” a “Mamíferos” em um clique. Observe que pode não existir a referência direta contrária, ou seja, de “Mamíferos” para “Elefante”. Adicionalmente se “Elevador” é o próximo artigo depois de “Elefante”, na ordem alfabética, pode-se ir com um clique de “Elefante” para “Elevador” e de “Elevador” para “Elefante”, pois há uma referência sequencial entre eles.

Paulo e André são dois amigos que contribuem para a Uiquipédia. Muitas vezes, André edita um artigo e quer que Paulo o ajude a revisar a modificação. A conexão de Paulo à Internet é discada, e por isso ele quer chegar na página que André editou usando o menor número de cliques possível, começando do artigo em que está, e navegando apenas por referências, diretas ou sequenciais.

Tarefa

Escreva um programa que, dados todas as referências diretas existentes na Uiquipédia, a página onde Paulo está, e a página editada por André, determina de quantos cliques Paulo precisa, no mínimo, para ver a página que foi modificada por André, utilizando as referências diretas e sequenciais.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). A primeira linha contém um único inteiro, N , que é o número de referências da Uiquipédia ($1 \leq N \leq 1.000$). As N linhas contém cada uma duas strings X e Y , separadas por um espaço, que são os nomes de duas páginas da Uiquipédia conectadas por uma referência direta (de X para Y). Todo artigo existente na Uiquipédia aparece pelo menos uma vez na descrição das referências diretas, permitindo que as referências sequenciais sejam extraídas das informações dadas. Note que uma referência direta pode ligar duas páginas que estariam ligadas também por uma referência sequencial.

Depois da descrição das referências, há uma linha em branco, e a linha seguinte contém duas cadeias de caracteres, P e A , que são a página atual de Paulo e a página editada por André. O nome de cada página é limitado a 100 caracteres e contém somente letras maiúsculas, letras minúsculas e o símbolo ‘_’. Observe que na ordem alfabética o símbolo ‘_’ é anterior às letras maiúsculas, que por sua vez são anteriores às letras minúsculas.

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha, contendo um único inteiro, que diz o número mínimo de cliques que são necessários para ir da página atual de Paulo até a página editada por André. Sempre é possível navegar de um artigo a outro.

Entrada 3 Pink_Floyd O_Lado_Escuro_Da_Lua Pink_Floyd O_Muro O_Muro Muro_de_Berlim O_Muro O_Lado_Escuro_Da_Lua Saída 1	Entrada 4 Chaves Quico Quico Chiquinha Professor_Girafales Dona_Florinda Chaves Dona_Clotilde Chaves Chiquinha Saída 2
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Reunião

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

Todos os anos, a SBC (Sociedade Brasileira de Caminhoneiros) reúne seus membros em alguma cidade para discutir sobre a profissão. Nessas reuniões são discutidos os problemas da categoria e são apresentadas sugestões sobre como melhorar as condições de trabalho.

O grande problema desse tipo de encontro é que os membros estão espalhados pelo país, uma vez que a profissão exige que eles viajem para diversos lugares todos os dias. Por isso, a escolha da cidade onde será feita a reunião sempre é feita de modo que não prejudique demais nenhum dos caminhoneiros. O critério para tal é que a maior das distâncias percorridas pelos caminhoneiros para chegar ao local da reunião deve ser a menor possível. Ou seja, a distância percorrida pelo caminhoneiro que vai percorrer a maior distância entre todos os caminhoneiros para chegar à reunião deve ser a menor possível.

Tarefa

Dadas as cidades onde se encontram os caminhoneiros e a descrição das estradas que interligam essas cidades, escreva um programa que determina qual será a menor distância máxima percorrida por um caminhoneiro para chegar até o local da reunião. Os caminhoneiros conhecem bem as estradas, e portando sempre fazem o menor caminho possível até a cidade da reunião. Sempre existe um caminho ligando quaisquer duas cidades.

Entrada

A primeira linha da entrada possui dois números inteiros N ($2 \leq N \leq 100$) e M ($N - 1 \leq M \leq 10000$), que representam, respectivamente, o número de cidades e o número de estradas que as interligam. As cidades são identificadas por números inteiros entre 0 e $N - 1$. As próximas M linhas da entrada possuem, cada uma, a descrição de uma estrada. Cada descrição de entrada é composta por três números inteiros: U , V e W , onde U e V representam cidades ($0 \leq U \leq N - 1$ e $0 \leq V \leq N - 1$) e W representa o comprimento da estrada que une essas duas cidades (todas as estradas são mão dupla, $1 \leq W \leq 100$). É sempre possível viajar entre qualquer duas cidades com as estradas existentes, mas pode haver mais de uma estrada ligando o mesmo par de cidades.

Saída

Seu programa deve imprimir uma única linha contendo um número inteiro, a distância máxima percorrida por um caminhoneiro para ir até a reunião, obedecidas as restrições estabelecidas (ou seja, essa distância máxima deve ser a menor possível).

Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 30 pontos, $N \leq 4$.

Exemplos

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

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

Entrada	Saída
7 12 0 1 22 0 2 30 0 5 35 1 5 11 1 6 30 1 2 25 2 3 15 2 6 10 3 4 15 3 5 10 4 5 20 5 6 33	30