

у2018-3-2. Кратчайшие пути. Игры

A. Floyd

2 seconds, 256 megabytes

You are given a fully-connected directed weighted graph. Please find the shortest path between each pair of nodes.

Input

The first line of the input contains one integer n ($1 \leq n \leq 100$) — the number of vertices. Next n lines contain n integers each and represent an adjacency matrix with weights. All integers do not exceed 100 in absolute value. The main diagonal contains only zeros.

Output

Output n lines with n integers each — the matrix of the distance between each pair of nodes, i.e., j -th number in i -th line represents the weight of the shortest path from vertex i to vertex j .

input
4 0 5 9 100 100 0 2 8 100 100 0 7 4 100 100 0
output
0 5 7 13 12 0 2 8 11 16 0 7 4 9 11 0

B. Shortest path 2

2 seconds, 256 megabytes

You are given an undirected connected weighted graph. Find the shortest distance from the first vertex to all others.

Input

The first line of the input contains two integers n and m ($2 \leq n \leq 30000, 1 \leq m \leq 400000$) — the number of vertices and the number of edges, correspondingly.

Next m lines contain the description of edges. Each edge is represented with three integers: the identifiers of its ends and the weight. The weight of each edge is a non-negative integer that does not exceed 10^4 .

Output

The sole line of the output should contain n integers — the distances from the first vertex to all others in the order of their identifiers.

input
4 5 1 2 1 1 3 5 2 4 8 3 4 1 2 3 3
output
0 1 4 5

C. Cycle of negative weight

2 seconds, 256 megabytes

You are given a directed weighted graph. Please find the negative cycle, or answer that it does not exist.

Input

The first line of the input contains one integer n ($1 \leq n \leq 100$) — the number of vertices. Next n lines contain n integers each and represent an adjacency matrix with weights. The weights do not exceed 10 000 in absolute value. If there is no edge, the corresponding weight is equal to 100 000.

Output

The first line of the output should contain "YES", if the cycle exists, or "NO", otherwise. If the cycle exists the second line should contain the number of vertices in the cycle, and the third line should contain the vertices in the order of the cycle.

input
2 0 -1 -1 0
output
YES 2 2 1

D. The shortest path of length K

4 seconds, 256 megabytes

You are given a directed graph. Please, find the shortest paths from S to all other vertices that consist of exactly K edges.

Input

The first line of the input contains four integers n, m, K and S ($1 \leq n \leq 10^4, 0 \leq m \leq 10^4, 0 \leq K \leq 100, 1 \leq S \leq n$) — the number of vertices, the number of edges, the total number of edges on paths, and the start vertex.

Next m lines contain the description of edges. Each edge is represented with three integers a_i, b_i, w ($1 \leq a_i, b_i \leq n, -10^5 \leq w \leq 10^5$) — the ends of the edge and its weight.

Output

The output should contain n integers on a separated line. i -th integer should represent the length of the minimal path between S and i with exactly K edges, or -1 if there is no such path.

input
3 3 1 1 1 2 100 2 3 300 1 3 2
output
-1 100 2

input
3 3 2 1 1 2 100 2 3 300 1 3 2
output
-1 -1 400

E. Shortest paths

2 seconds, 256 megabytes

You are given a weighted directed graph and vertex s . You should find the shortest distance from s to all the other vertices.

Input

The first line of the input contains three integers n, m, s ($2 \leq n \leq 2\,000, 1 \leq m \leq 5\,000$) — the number of vertices, the number of edges, and the identifier of the start vertex.

Next m lines contain the description of edges. Each edge is represented with three integers — the ends and the weight. The weight of a graph is an integer and does not exceed 10^{15} by the absolute value. The graph can contain loops and multiple edges between a pair of vertices.

Output

The output should contain n lines — for each vertex, output the length of the shortest path from s . If there is no path then output "*", and if there does not exist a shortest path then output "-".

input
6 7 1 1 2 10 2 3 5 1 3 100 3 5 7 5 4 10 4 3 -18 6 1 -1
output
0 10 - - - *

Statement is not available on English language

F. В поисках утраченного кефира

2 секунды, 256 мегабайт

Школьник Вася хочет найти запасы спрятанного кефира. По легенде, кефир находится в домиках a , b или c . Вася хочет проверить каждый из этих трёх домиков, потратив на это минимальное количество времени.

Местность, в которой находится Вася представляет собой n домиков, пронумерованных числами от 1 до n . Некоторые из домиков соединены дорогами, по которым можно ходить в обе стороны. Время прохождения i -й дороги составляет w_i секунд. Путём в графе называется непустая последовательность вершин, такая что все соседние вершины соединены дорогой. Требуется помочь Васе найти путь, содержащий вершины a , b , c , такой что суммарное время прохождения всех дорог на пути минимально. При этом, если мы прошли по какой-то дороге дважды (или более), то и время её прохождения следует учитывать соответствующее количество раз. Начинать свой путь Вася может из любой вершины.

Гарантируется, что a , b , c — попарно различные домики.

Входные данные

В первой строке ввода записаны два числа n и m ($3 \leq n \leq 100\,000$, $0 \leq m \leq 200\,000$) — количество домиков в ЛКШ и дорог между ними соответственно.

Следующие m строк содержат описания дорог, по одному в строке. Каждая из дорог задаётся тройкой чисел u_i , v_i , w_i ($1 \leq u_i, v_i \leq n$, $1 \leq w_i \leq 10^9$) — номерами соединённых домиков и временем, затрачиваемым на прохождение данной дороги. По каждой дороге разрешено ходить в обе стороны. Гарантируется, что любая пара домиков соединена не более чем одной дорогой. Также гарантируется, что нет дороги, соединяющей домик с самим собой.

В последней строке записаны три попарно различных числа a , b , c ($1 \leq a, b, c \leq n$).

Выходные данные

Выведите одно целое число — минимальное возможное время, которое нужно затратить на прохождение пути, содержащего домики a , b и c . Если пути, содержащего все три домика не существует, то выведите -1 .

входные данные
4 4 1 2 3 2 3 1 3 4 7 4 2 10 1 4 3
выходные данные
11

входные данные
4 2 1 2 10 2 3 5 1 2 4
выходные данные
-1

В первом примере путь 1–2–3–4 является минимальным (11 секунд). Например, путь 1–2–4–3 не подходит, так как занимает больше времени (20 секунд), а путь 3–4–2 не подходит, так как домик a оказывается не посещенным.

Во втором примере не существует способа добраться от домика b до домика c , поэтому искомого пути не существует.

G. Long long trip

2 seconds, 256 megabytes

There is a country with n cities. The cities are numbered 1 through n . There are some bidirectional roads in the country. Each road connects a pair of cities. More precisely, for each i , road i connects the cities a_i and b_i .

Limit is a deer that likes to travel along the roads. Traveling along road i (in either direction) takes him exactly d_i minutes. Limit does not like cities, so he never waits in a city.

Limit is currently in the city 1, starting his travels. In exactly t minutes, he wants to be in the city n . You are to find, whether Limit can reach city n in exactly t minutes.

Input

First line contains integers n and m — the number of cities and the number of roads in the country, respectively ($1 \leq n \leq 50$). Next m lines describe the roads. Each line consists of integers a_i , b_i and d_i — the endpoints of the road and its length ($1 \leq a_i, b_i \leq n$; $1 \leq d_i \leq 10^4$).

The last line contains integer t — the number of minutes Limit wants to travel for ($1 \leq t \leq 10^{18}$).

Output

Output "Possible" if Limit can reach city n in exactly t minutes, output "Impossible" otherwise.

input
3 3 1 3 7 1 2 6 2 3 5 11
output
Possible

I. Retrograde analysis

2 seconds, 256 megabytes

You are given a directed graph with n vertices and m edges. Alice and Bob play a game. Initially vertex i contains a token. On each turn a player can move the token using one of the outgoing edges. The player that can't make a move, loses.

For every vertex i determine, who wins if both players play optimally.

Input

Input consists of one or more testcases. Every testcase contains a description of directed graph. Graph is described as follows.

First line contains two integers n ($1 \leq n \leq 300\,000$) and m ($1 \leq m \leq 300\,000$). Next m lines contain edges of the graph, each one is described by two integers from 1 to n — starting and ending vertices of an edge. Graph can contain loops and multiple edges.

The sum of n over all testcases in an input doesn't exceed 300 000.

The sum of m over all testcases in an input doesn't exceed 300 000.

Output

For each testcase output n lines. For each vertex i output FIRST, SECOND or DRAW, depending on who wins, if both players play optimally starting the game from vertex i .

Output an empty line after each testcase.

input
5 5 1 2 2 3 3 1 1 4 4 5 2 1 1 2 4 4 1 2 2 3 3 1 1 4
output
DRAW DRAW DRAW FIRST SECOND FIRST SECOND FIRST FIRST SECOND SECOND

J. Nimbers

2 seconds, 256 megabytes

You are given a directed acyclic graph. Find a Nimber (Grundy function) for each vertex.

Input

The first line contains two integers n, m : the number of vertices and edges in the graph ($1 \leq n, m \leq 100\,000$). Each of the next m lines contains two integers x, y ($1 \leq x, y \leq n$).

Note that the given graph may contain multiple edges.

Output

Print n integers: the values of Grundy functions for all starting vertices.

input
3 3 1 3 7 1 2 6 2 3 5 25
output
Possible

input
2 1 1 2 1 9
output
Possible

input
2 1 2 1 1 1000000000000000000
output
Impossible

input
4 3 1 3 10 1 2 10 2 3 10 1000
output
Impossible

H. Game

2 seconds, 256 megabytes

You are a given directed acyclic graph. One of the graph vertices contains a token. Two players play a game, alternating turns. On each turn, player can move the token from vertex u to vertex v , if there is an edge uv in the graph. The player that can't make a move loses.

Determine the winner if both players play optimally.

Input

The first line contains three integers n, m and s ($1 \leq n, m \leq 10^5$; $1 \leq s \leq n$) — the number of vertices, edges and the vertex, where token is located initially. The next m lines contain graph edges. Each edge is described by two integers v and u — starting and ending vertices ($1 \leq v, u \leq n$).

The graph doesn't contain cycles.

Output

Output First player wins if the first player wins, otherwise, print Second player wins.

input
3 3 1 1 2 2 3 1 3
output
First player wins

input
3 2 1 1 2 2 3
output
Second player wins

input
3 3 1 2 2 3 1 3
output
2 1 0

input
2 1 2 1
output
0 1

Statement
is not
available
on
English
language

К. Дровосек

2 секунды, 256 мегабайт

Двое играют в следующую игру: имеется дерево с отмеченной вершиной (корнем). Игроки ходят по очереди. За ход игрок рузрубает ветку (стирает ребро), причем из двух получившихся компонент связности остается только та, которая содержит корень — остальная отваливается и больше в игре не участвует. Проигрывает тот, кто не может сделать ход. Определите, может ли выиграть первый игрок, и если да, то укажите любой из его выигрышных ходов.

Входные данные

В первой строке входного файла находятся 2 числа, N и R — количество вершин дерева и номер корня ($1 < N \leq 100\,000$, $1 \leq R \leq N$). Далее следуют $N - 1$ строка, в каждой из которых находятся два числа — номера вершин, которые соединяет очередное ребро.

Выходные данные

Выведите в выходной файл одно число 1 или 2 — номер игрока, который выигрывает при правильной игре. Если выигрывает первый игрок, то выведите также любой его выигрышный ход, т.е. порядковый номер ребра во входном файле, которое ему достаточно разрубить первым ходом (число от 1 до $N - 1$).

входные данные
1 1
выходные данные
2

входные данные
2 2 1 2
выходные данные
1 1