Problem A. 196769. Nugman and Graph

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Nugman has a weighted directed graph, consisting of n vertices. In this graph, any vertex has a directional edge to any other vertex. Nugman loves playing with the graph and now he has invented a new game

- The game consists of n steps.
- On the i-th step Nugman will add vertex x_i to the graph. As Nugman add a vertex, he also will add all the edges that go in and out of this vertex.

Nugman wants to know the **longest shortest path** in the graph after adding each vertex.

Input

The first line contains an integer n ($1 \le n \le 500$) the number of vertices in the graph. Next n lines contain n integers each — the graph adjacency matrix: the j-th number in the i-th line a_{ij} ($1 \le a_{ij} \le 100000$) represents the weight of the edge that goes from vertex i to vertex j. It is guaranteed that $a_{ii} = 0$.

The next line contains n distinct integers: $x_1, x_2, \ldots, x_n$ ($1 \le x_i \le n$) the vertices that Nugman deletes.

Output

Print n integers — the i-th longest shortest path after adding vertex x_i.

Examples

standard input	standard output

2
0 5
4 0
1 2
	0
	5

4
0 3 1 1
6 0 400 1
2 4 0 1
1 1 1 0
4 1 2 3
	0
	1
	2
	2

Problem B. 194603. Red Riding Hood and Long Journey

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	256 megabytes

Red Riding Hood lives in a small village. Today she received a letter from her grandmother, in which it is written she is ill. Therefore, Little red Riding hood, being a caring girl, wants to bring a basket of fruits and medicine for her grandmother who lives in another village. She knows a village that has medicines and a merchant who sells fresh fruits. The path can be long and therefore, she wants to come as quickly as possible. Help Red Riding Hood find the shortest way to her grandmother so that she can buy all the necessary presents along the way.

Input

In the first line you will be given two numbers N, M ($1 \leq N \leq 100000, 0 \leq M \leq \min(2 \cdot 10^5, N(N-1)/2)$), denoting the number of villages and number of paths between two neighboring villages. Next M lines contains three numbers: v, u ($1 \leq v, u \leq N$) and c ($1 \leq c \leq 1000$). First two numbers represents the path between villages v and u, the third is the distance between them. Last line contains four numbers: s, a, b, f ($1 \leq s, a, b, f \leq N$), denoting the Red Riding Hood, merchant, pharmacy, and grandmother's villages.

Output

Print one single number - the shortest distance to grandmother's village so that Red Riding Hood can buy all the necessary presents along the way. If there is no such path then print -1.

Examples

standard input	standard output

`4 5`
`1 2 3`
`3 4 7`
`1 3 1`
`3 2 1`
`2 4 6`
`1 2 3 4`
	`8`

`4 2`
`1 3 10`
`2 4 15`
`2 2 3 4`
	`-1`

`6 9`
`1 2 5`
`1 3 6`
`2 4 4`
`2 5 1`
`3 4 4`
`3 6 6`
`4 5 3`
`4 6 2`
`5 6 3`
`1 3 3 3`
	`6`

Note

Everyone who presented in the task are not necessarily located in different villages. It is also worth to clarify that path between two neighboring villages is available for both directions and guaranteed that there is no more than one direct path between any two villages.

Problem C. 195790. Nugman and Graph 3

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Nugman and Olzhas are stuck in a magical world called ADS, and they want to return to KBTU. To return, they must walk together to their destination. There are n cities (numbered from 1 to n) and m two-way airlines for each pair of different cities x and y, there is a two-way road between cities x and y if and only if there is no airline between them. The trip between the cities takes exactly one hour.

Nugman is afraid of heights and therefore walks, and Olzhas knows how to fly in this world. They start their journey in city 1 and end in city n. Nugman and Olzhas **cannot intersect**, with the exception of cities 1 and n.

In what is the smallest number of hours both will get to the city n?

Input

The first line of the input contains two integers n and m ($2 \le n \le 400$, $0 \le m \le 79800$) — the number of towns and the number of railways respectively.

Each of the next m lines contains two integers u and v, denoting a railway between towns u and v ($1 \le u, v \le n$).

You may assume that there is at most one railway connecting any two towns.

Subtasks

Output

In a single line print a single integer — the answer to the problem. If it's impossible print -1.

Examples

standard input	standard output

`3 1`
`1 3`
	`2`

`3 3`
`1 2`
`2 3`
`3 1`
	`-1`

Problem D. Element of SPADA

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

There are n cities located on a 2d plane, city i has coordinates (x_i, y_i). To travel between city i and j, you need to be at least $|x_i - x_j| + |y_i - y_j|$ years old. You're currently located at city with index 1, and decided to visit city n, but you're not old enough to do it, so you want to fabricate your documents. To avoid suspicion, you want to find the minimum age required to travel between city 1 and n.

Input

First line contains integer n ($1 \leq n \leq 1000$), a number of cities. The next n lines contain two integers x_i, y_i ($1 \leq x_i, y_i \leq 10^9$), coordinate of city i.

Output

Output single integer - minimum age required to travel between city 1 and city n.

Examples

standard input	standard output

4
1 1
3 3
5 2
5 7
	5

1 1 1	0

Problem E. 252592. Cycle.

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Given a directed graph. Determine if it contains a cycle of negative weight, and if so, print it.

Input

The first line contains the number $N(1 \leq N \leq 100)$ – the number of graph vertices. The next N lines contain N numbers each – the adjacency matrix of the graph. The absolute value of edge weights is less than 100000. If there is no edge, the corresponding value is 100000.

Output

In the first line print `YES` if the cycle exists, or `NO` otherwise. If there is a cycle, print in the second line the number of vertices in it (starting path from the same vertex as in the end of path), and in the third line print the vertices included in this cycle (path) in the order of traversal. If there are several cycles, then print any of them.

Example

standard input	standard output

`3`
`100000 100000 -51`
`100 100000 100000`
`100000 -50 100000`
	`YES`
	`4`
	`3 2 1 3`