Codeforces β Sponsored by Telegram

Shortest Paths

A. Dijkstra?

time limit per test: 1 second
memory limit per test: 1024 MB
input: standard input
output: standard output

You are given a weighted undirected graph. The vertices are enumerated from 1 to n. Your task is to find the shortest path between the vertex 1 and the vertex n.

Input

The first line contains two integers n and m $(2 \le n \le 10^5, 0 \le m \le 10^5)$, where n is the number of vertices and m is the number of edges. Following m lines contain one edge each in form a_i, b_i and w_i $(1 \le a_i, b_i \le n, 1 \le w_i \le 10^6)$, where a_i, b_i are edge endpoints and w_i is the length of the edge.

It is possible that the graph has loops and multiple edges between pair of vertices.

Output

Write the only integer `-1` in case of no path. Write the shortest path in opposite case. If there are many solutions, print any of them.

Examples

input

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

output

```
1 4 3 5
```

input

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

output

```
1 4 3 5
```

B. Two Buttons

time limit per test: 2 seconds
memory limit per test: 1024 MB
input: standard input
output: standard output

Vasya has found a strange device. On the front panel of a device there are: a red button, a blue button and a display showing some positive integer. After clicking the red button, device multiplies the displayed number by two. After clicking the blue button, device subtracts one from the number on the display. If at some point the number stops being positive, the device breaks down. The display can show arbitrarily large numbers. Initially, the display shows number n.

Bob wants to get number m on the display. What minimum number of clicks he has to make in order to achieve this result?

Input

The first and the only line of the input contains two distinct integers n and m $(1 \le n, m \le 10^4)$, separated by a space .

Output

Print a single number — the minimum number of times one needs to push the button required to get the number m out of number n.

Examples

input

```
4 6
```

output

```
2
```

input

```
10 1
```

output

```
9
```

Note

In the first example you need to push the blue button once, and then push the red button once.

In the second example, doubling the number is unnecessary, so we need to push the blue button nine times.

C. The Two Routes

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In Absurdistan, there are n towns (numbered 1 through n) and m bidirectional railways. There is also an absurdly simple road network — for each pair of different towns x and y, there is a bidirectional road between towns x and y **if and only if** there is no railway between them. Travelling to a different town using one railway or one road always takes exactly one hour.

A train and a bus leave town 1 at the same time. They both have the same destination, town n, and don't make any stops on the way (but they can wait in town n). The train can move only along railways and the bus can move only along roads.

You've been asked to plan out routes for the vehicles; each route can use any road/railway multiple times. One of the most important aspects to consider is safety — in order to avoid accidents at railway crossings, the train and the bus must not arrive at the same town (except town n) simultaneously.

Under these constraints, what is the minimum number of hours needed for both vehicles to reach town n (the maximum of arrival times of the bus and the train)? Note, that bus and train are not required to arrive to the town n at the same moment of time, but are allowed to do so.

Input

The first line of the input contains two integers n and m $(2 \le n \le 400, 0 \le m \le n(n-1)/2)$ — 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 $(1 \le u, v \le n, u \ne v)$.

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

Output

Output one integer — the smallest possible time of the later vehicle's arrival in town n. If it's impossible for at least one of the vehicles to reach town n, output -1.

Examples

input

```
4 2
1 3
3 4
```

output

```
2
```

input

```
4 6
1 2
1 3
1 4
2 3
2 4
3 4
```

output

```
-1
```

input

```
5 5
4 2
3 5
4 5
5 1
1 2
```

output

```
3
```

Note

In the first sample, the train can take the route $1 \rightarrow 3 \rightarrow 4$ and the bus can take the route $1 \rightarrow 2 \rightarrow 4$. Note that they can arrive at town 4 at the same time.

In the second sample, Absurdistan is ruled by railwaymen. There are no roads, so there's no way for the bus to reach town 4.

D. Complete The Graph

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

ZS the Coder has drawn an undirected graph of n vertices numbered from 0 to $n - 1$ and m edges between them. Each edge of the graph is weighted, each weight is a **positive integer**.

The next day, ZS the Coder realized that some of the weights were erased! So he wants to reassign **positive integer** weight to each of the edges which weights were erased, so that the length of the shortest path between vertices s and t in the resulting graph is exactly L. Can you help him?

Input

The first line contains five integers n, m, L, s, t $(2 \le n \le 1000,\ 1 \le m \le 10\,000,\ 1 \le L \le 10^9,\ 0 \le s, t \le n - 1,\ s \ne t)$ — the number of vertices, number of edges, the desired length of shortest path, starting vertex and ending vertex respectively.

Then, m lines describing the edges of the graph follow. i-th of them contains three integers, u_i, v_i, w_i $(0 \le u_i, v_i \le n - 1,\ u_i \ne v_i,\ 0 \le w_i \le 10^9)$. u_i and v_i denote the endpoints of the edge and w_i denotes its weight. If w_i is equal to 0 then the weight of the corresponding edge was erased.

It is guaranteed that there is at most one edge between any pair of vertices.

Output

Print "NO" (without quotes) in the only line if it's not possible to assign the weights in a required way.

Otherwise, print "YES" in the first line. Next m lines should contain the edges of the resulting graph, with weights assigned to edges which weights were erased. i-th of them should contain three integers u_i, v_i and w_i, denoting an edge between vertices u_i and v_i of weight w_i. The edges of the new graph must coincide with the ones in the graph from the input. The weights that were not erased must remain unchanged whereas the new weights can be any **positive integer** not exceeding 10^{18}.

The order of the edges in the output doesn't matter. The length of the shortest path between s and t must be equal to L.

If there are multiple solutions, print any of them.

Examples

input

```
5 5 13 0 4
0 1 5
2 1 2
3 2 3
1 4 0
4 3 4
```

output

```
YES
0 1 5
2 1 2
3 2 3
1 4 8
4 3 4
```

input

```
2 1 123456789 0 1
0 1 0
```

output

```
YES
0 1 123456789
```

input

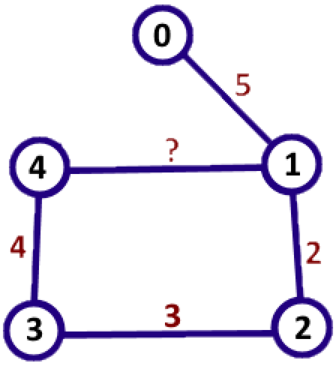
```
2 1 999999999 1 0
0 1 1000000000
```

output

```
NO
```

Note

Here's how the graph in the first sample case looks like :



In the first sample case, there is only one missing edge weight. Placing the weight of 8 gives a shortest path from 0 to 4 of length 13.

In the second sample case, there is only a single edge. Clearly, the only way is to replace the missing weight with 123456789.

In the last sample case, there is no weights to assign but the length of the shortest path doesn't match the required value, so the answer is "`NO`".