D. Axel and Marston in Bitland

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A couple of friends, Axel and Marston are travelling across the country of Bitland. There are n towns in Bitland, with some pairs of towns connected by one-directional roads. Each road in Bitland is either a pedestrian road or a bike road. There can be multiple roads between any pair of towns, and may even be a road from a town to itself. However, no pair of roads shares the starting and the destination towns along with their types simultaneously.

The friends are now located in the town 1 and are planning the travel route. Axel enjoys walking, while Marston prefers biking. In order to choose a route diverse and equally interesting for both friends, they have agreed upon the following procedure for choosing the road types during the travel:

- The route starts with a pedestrian route.
- Suppose that a beginning of the route is written in a string s of letters P (pedestrain road) and B (biking road). Then, the string is appended to s, where stands for the string s with each character changed to opposite (that is, all pedestrian roads changed to bike roads, and vice versa).

In the first few steps the route will look as follows: P, PB, PBBP, PBBPBPPB, PBBPBPPBBPPBPBBP, and so on.

After that the friends start travelling from the town 1 via Bitlandian roads, choosing the next road according to the next character of their route type each time. If it is impossible to choose the next road, the friends terminate their travel and fly home instead.

Help the friends to find the longest possible route that can be travelled along roads of Bitland according to the road types choosing procedure described above. If there is such a route with more than 10^{18} roads in it, print -1 instead.

Input

The first line contains two integers n and m $(1 \le n \le 500,\ 0 \le m \le 2n^2)$ — the number of towns and roads in Bitland respectively.

Next m lines describe the roads. i-th of these lines contains three integers v_i, u_i and t_i $(1 \le v_i, u_i \le n,\ 0 \le t_i \le 1)$, where v_i and u_i denote start and destination towns indices of the i-th road, and t_i decribes the type of i-th road (0 for a pedestrian road, 1 for a bike road). It is guaranteed that for each pair of distinct indices i, j such that $1 \le i, j \le m$, either $v_i \ne v_j$, or $u_i \ne u_j$, or $t_i \ne t_j$ holds.

Output

If it is possible to find a route with length strictly greater than 10^{18}, print -1. Otherwise, print the maximum length of a suitable path.

Examples

input

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

output

```
3
```

input

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

output

```
-1
```

Note

In the first sample we can obtain a route of length 3 by travelling along the road 1 from town 1 to town 2, and then following the road 2 twice from town 2 to itself.

In the second sample we can obtain an arbitrarily long route by travelling the road 1 first, and then choosing road 2 or 3 depending on the necessary type.