In a certain city there are N intersections connected by one-way and two-way streets. It is a modern city, and several of the streets have tunnels or overpasses. Evidently it must be possible to travel between any two intersections. More precisely given two intersections V and W it must be possible to travel from V to W and from W to V.

Your task is to write a program that reads a description of the city street system and determines whether the requirement of connectedness is satisfied or not.

Input

The input contains several test cases. The first line of a test case contains two integers N and M, separated by a space, indicating the number of intersections ($2 \leq N \leq 2000$) and number of streets ($2 \leq M \leq N(N-1)/2$). The next M lines describe the city street system, with each line describing one street. A street description consists of three integers V, W and P, separated by a blank space, where V and W are distinct identifiers for intersections ($1 \leq V, W \leq N$, $V \neq W$) and P can be 1 or 2; if $P = 1$ the street is one-way, and traffic goes from V to W; if $P = 2$ then the street is two-way and links V and W. A pair of intersections is connected by at most one street.

The last test case is followed by a line that contains only two zero numbers separated by a blank space.

Output

For each test case your program should print a single line containing an integer G, where G is equal to one if the condition of connectedness is satisfied, and G is zero otherwise.

Sample Input

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

Sample Output

```
1
1
0
0
```