



## Objective

Your two new robots, Algotron 2000 and Polynomialator-ZX, got lost in the New York subway. To make matters worse, the device that allows you to remotely control them from the top of your secret building has just broken down: each order sent will be transmitted indiscriminately to each of the two robots.

If only they were in the same place, you could get them back without any problems... If we could find a command sequence such that the two robots meet by receiving the same orders from their respective starting positions, everything would be arranged!

You decide to write a program: it will analyze the map of the tunnels and find a suitable order suite, if it exists!

## Data format

### Input

Row 1: two integers **N** and **M** separated by spaces between 1 and 100 and between 1 and 500 respectively. **N** and **M** represent the maximum station identifier (each station is identified by a number) and the number of tunnels on the map, respectively.

Row 2: two integers between 0 and **N** indicating the number of the station where Algotron 2000 and Polynomialator-ZX are located respectively.

Rows 3 to **M+2**: a row composed of 3 elements, separated by spaces representing a tunnel:

- An integer number indicating the number of a station (between 0 and **N**)
- The G, D or T character (for "Gauche"(left), "Droite" (right), or "Tout droit"(straight))
- An integer number, indicating the number of another station.

For example, a row with "4 G 3" indicates that turning left from Station 4, a tunnel is leading to Station 3. Some tunnels can be looping (e.g. "4 T 4").

Warning: Some tunnels are split, making it possible to have both "4 G 3" and "4 G 2". In such a case, if a robot is located for example in Station 4 receives the "G" command, it is considered that it can then in both stations 3 and 2. Your program doesn't have to indicate which station it's in: for the robots to meet, it's sufficient that at the end of all the commands there is a common station among the possible stations in which the robots can be found. In other words, if there is a path that corresponds to the directions indicated by your program allowing robots to meet then your answer is correct.

Note: it is dangerous to travel in the subway, and each tunnel is one way!

### Output

A series of G, D or T characters that, if tracked from the initial location of each robot, should take them to the same location, if it exists. If you can't find such a sequence, output the string the *fail*.

### **Examples**

#### Input

```
5 6
3 2
3 D 4
2G 1
3G 3
1 D 4
4 T 1
4 T 4
```

#### Ouput

```
GDDT
```

#### Input

```
5 12
3 2
1 T 2
1 G 4
```

2 D 1  
2 D 3  
3 D 2  
3G 4  
4 T 1  
4 D 4  
4G 5  
5 T 1  
5 T 3  
5 D 4

## Output

fail