Problem A
Crashing Robots

source: `robots.c` *or* `robots.cpp` *or* `robots.java`

Description

In a modernized warehouse, robots are used to fetch the goods. Careful planning is needed to ensure that the robots reach their destinations without crashing into each other. Of course, all warehouses are rectangular, and all robots occupy a circular floor space with a diameter of1 meter. Assume there are N robots, numbered from 1 through N. You will get to know the position and orientation of each robot, and all the instructions, which are carefully (and mindlessly) followed by the robots. Instructions are processed in the order they come. No two robots move simultaneously; a robot always completes its move before the next one starts moving.

A robot crashes with a wall if it attempts to move outside the area of the warehouse, and two robots crash with each other if they ever try to occupy the same spot.

Input

The first line of input is K, the number of test cases. Each test case starts with one line consisting of two integers, $1 \leq A, B \leq 100$, giving the size of the warehouse in meters. A is the length in the EW-direction, and B in the NS-direction.

The second line contains two integers, $1 \leq N, M \leq 100$, denoting the numbers of robots and instructions respectively.

Then follow N lines with two integers, $1 \leq X_i \leq A$, $1 \leq Y_i \leq B$ and one letter (`N`, `S`, `E` or `W`), giving the starting position and direction of each robot, in order from1 through N. No two robots start at the same position.

Finally there are M lines, giving the instructions in sequential order. An instruction has the following format:

```
<robot #> <action> <repeat>
```

Where `<action>` is one of

- `L`: turn left 90 degrees,
- `R`: turn right 90 degrees, or
- `F`: move forward one meter,

and $1 \leq$ `<repeat>` ≤ 100 is the number of times the robot should perform this single move.

Output

Output one line for each test case:

- "`Robot i crashes into the wall`", if robot `i` crashes into a wall. (A robot crashes into a wall if $X_i = 0$, $X_i = A + 1$, $Y_i = 0$ or $Y_i = B + 1$.)
- "`Robot i crashes into robot j`", if robots `i` and `j` crash, and `i` is the moving robot.
- "`OK`", if no crashing occurs.

Only the first crash is to be reported.

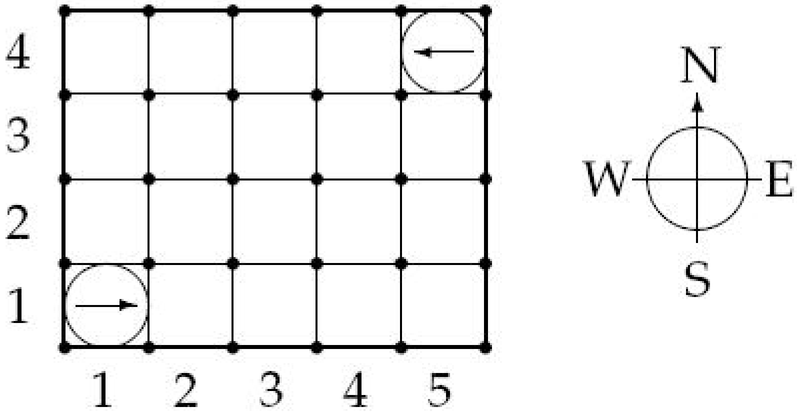

Figure 1: The starting positions of the robots in the sample warehouse.

Sample

Input

```
4
5 4
2 2
1 1 E
5 4 W
1 F 7
2 F 7
5 4
2 4
1 1 E
5 4 W
1 F 3
2 F 1
1 L 1
1 F 3
5 4
2 2
1 1 E
5 4 W
1 L 96
1 F 2
5 4
2 3
1 1 E
5 4 W
1 F 4
1 L 1
1 F 20
```

Output

```
Robot 1 crashes into the wall
Robot 1 crashes into robot 2
OK
Robot 1 crashes into robot 2
```