2000-2001 ACM North Central Regional Programming Contest
Problem 3 — Ro and Bot Meet

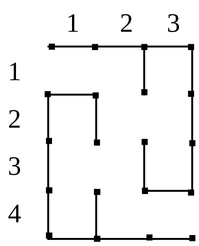
A flat rectangular maze is constructed from square cells. Each cell has one, two, or three open sides. The maze has two entry points from outside the maze, one in the cell at the upper left corner of the maze, and the other in the cell at the lower right corner of the maze. A maze may have dead ends, but it does not contain any loops. That is, there is no forward path through a maze that will lead back to a previously visited cell.

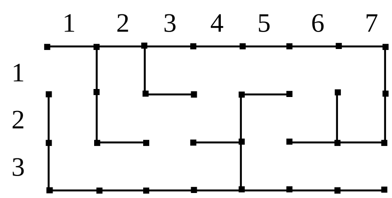
At the same time, two robots (named Ro and Bot) enter and begin traversing the maze. Ro enters at the upper left corner, and Bot enters at the lower right corner. Each robot takes the same amount of time to move from one cell to the next. The robots follow these rules in traversing the maze:

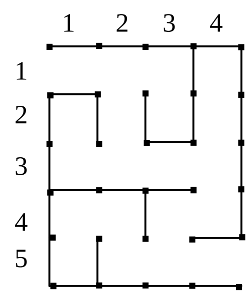
- If a robot enters a cell that has only one open side (a dead end), the robot turns around and leaves the cell.
- If a robot enters a cell that has two open sides, it leaves using the side through which it did not enter.
- If a robot enters a cell that has three open sides *A*, *B* and *C* (labeled in clockwise order), and was first entered through side *A*, the robot must choose to leave through side *B* or side *C*. Ro will choose to exit through side *B*, and Bot will choose to exit through side *C*. If a dead end later causes a robot to return to the cell, it will then leave through the last remaining open side. For example, if Ro enters a cell with three open sides through side *A*, it will leave through side *B*. If that route leads to dead ends, Ro eventually returns to the cell and leaves through side *C*. Naturally if all paths reached through sides *B* and *C* lead to dead ends, the robot eventually retreats back through side A (which it first used when it entered the cell).
- The robots stop if they meet in a cell (at the same time) or if they exit the maze.

You are to write a program that will determine from the description of a maze if the robots will stop inside the maze, and if they do, the cell at which they will stop.

A few examples will clarify these ideas. Shown below are three mazes. Below each maze is a table showing the cells through which the robots will pass while traversing the maze. Row and column numbers are used to identify the cells in each maze.







Ro		Bot	
Row	*Col*	*Row*	*Col*
1	1	4	3
1	2	4	2
2	2	3	2
2	3	2	2
1	3	2	3
2	3	3	3
3	3	2	3
2	3	1	3
2	2	2	3
3	2	2	2
4	2	1	2
4	3	1	1
exits		*exits*	

Ro		Bot	
Row	*Col*	*Row*	*Col*
1	1	3	7
2	1	3	6
3	1	3	5
3	2	2	5
3	3	2	6
2	3	1	6
2	2	1	7
1	2	2	7
2	2	1	7
2	3	1	6
2	4	1	5
1	4	1	4

Ro		Bot	
Row	*Col*	*Row*	*Col*
1	1	5	4
1	2	5	3
1	3	4	3
2	3	4	4
1	3	3	4
1	2	2	4
2	2	1	4
3	2	2	4
3	3	3	4
3	4	3	3
2	4	3	2
1	4	2	2
2	4	1	2
3	4	1	3
4	4	2	3
4	3	1	3
5	3	1	2
5	4	1	1
exits		*exits*	

Input

The input data will contain descriptions of multiple mazes. The description of each maze begins with integers giving the number of rows (*NR*) and number of columns (*NC*) in the maze. Neither *NR* nor *NC* will be larger than 20. Following these integers there will appear $NR \times NC$ hexadecimal digits, corresponding to the cells in the maze in row-major order. Blanks and end of line characters may be included at arbitrary places for readability. Hexadecimal digits include the decimal digits **`0`** through **`9`** (representing themselves), and the upper case letters **`A`** through **`F`** (representing the values 10 through 15 respectively). Each hexadecimal digit identifies the open sides of the corresponding cell in the maze, as follows. Each side of a cell has an associated number: top = 1, right = 2, bottom = 4, and left = 8. If the numbers corresponding to the open sides of a cell are totaled, they yield the corresponding hexadecimal digit that will appear in the input for that cell. For example, a cell that has only its left and right sides open would be specified in the input as the hexadecimal digit **`A`**, since 8 (left) + 2 (right) = 10. A cell with its right, bottom, and left sides open would be specified in the input as the hexadecimal digit **`E`**, since the value of **`E`** is 14, or 2 (right) + 4 (bottom) + 8 (left).

A pair of zeroes follows the data for the last case.

Output

The output for each maze must begin with the word **`Maze`** followed by the maze sequence number (they are numbered sequentially starting with 1), a colon, and a blank. This is then followed by the message

```
The robots do not meet.
```

or the message

```
The robots meet in row R, column C.
```

as appropriate (with ***`R`*** and ***`C`*** replaced by the row and column number at which the robots meet). Leave a single blank line between the output for each maze.

Sample Input

```
4 3 A C 4 4 7 D 7 D 1 1 3 A
3 7 C 4 2 E A E C 5 3 E 9 6 9 1 3 A B 8 3 A A
5 4 A E C 4 4 5 1 5 3 B A D 6 C 6 9 1 3 B A
0 0
```

Expected Output

```
Maze 1: The robots do not meet.

Maze 2: The robots meet in row 1, column 4.

Maze 3: The robots do not meet.
```