Two robots are lost in a warehouse. The robots are numbered 1 and 2. The warehouse is a grid with R rows and C columns with each field either blocked or free. The robots are controlled through radio command. Each command consists of two pieces of data:

- *robot* - 1 or 2, the number of robot we are moving
- *direction* - character 'U', 'D', 'L' or R representing the direction (up, down, left, right) we want to move the robot in.

If the destination field is blocked, taken by another robot or outside the warehouse, nothing happens and the robot stays where he is. If the field is free, the robot moves in it.

The robots are equipped with GPS devices, however due to malfunctions during deployment we cannot receive the exact location of the robots, only their M**anhattan distance** to each other. If the robots are on fields (r_1, c_1) and (r_2, c_2), then their Manhattan distance is $|r_1 - r_2| + |c_1 - c_2|$.

After each command, unsuccessful or successful, the only information we know is the current distance.

Robots are currently in the warehouse on different, unoccupied locations. Write a program that will issue a series of command required to place both robots on two special extraction points in the warehouse. The commands are issued by writing on the standard output. After each command you must read the current distance on the standard input.

The warehouse will be such that all free fields are connected.

INTERACTION

Before interacting with the robots, a few input data is given.

The first line contains integers R and C ($2 \leq R, C \leq 200$), number of rows and columns in the warehouse.

The next R rows contain C characters each. Only characters '**.**', '#' and 'x' will appear. '.' represents a free field, '#' blocked field and 'x' one of the two extraction fields. There will be exactly two 'x' characters.

The next line contains the starting Manhattan distance.

After loading all input data, you may begin issuing command to the robots using standard input. Each command must be formatted "robot direction" followed by newline character. After each command you **must** flush the standard output. Use fflush(stdout) in C/C++ or flush(StdOut) in Pascal.

When you are done issuing commands, print "0" on a line by itself and **exit your program with return code 0.**

SCORING

Test cases worth 40 points do not contain blocked fields.

Test cases worth 80 points have R and $C \leq 50$.

TESTING

You may test your solutions using the TEST interface in the evaluation system. For each test a input file must be provided. The input file must follow the following formatting:

The first line of input must contain two integers R and C ($2 \leq R, C \leq 200$).

The following R lines must contain C characters each. The characters can be only '.', '#', 'x', '1' and '2'. There must be exactly two 'x' characters. Characters '1' and '2' denote the starting locations of the robots. There must be exactly one of each present in the input. Of course, the characters '1' and '2' will be converted to '.' when presented as input to your solution.

An example of such input file is:

```
4 5
##x1.
.##..
.....
2...x
```

To test your code on the evaluation system, you must first submit your solution using the SUBMIT interface and then use the TEST interface for testing.