ROBOT

Dave wishes to plunder an empty house. In order to discover the exact layout of the target house, he has teleported a remote-controlled robot inside.

The house is modeled as a grid composed of unit squares. Some squares are empty while others represent walls. The robot is initially **somewhere inside the house**, in an **empty square**.

Dave can **move the robot** by sending a simple command – the direction to move in (up, down, left or right). The robot can only move into an empty square; if it tries to move into a square occupied by a wall it will remain at its original location. After each command, it **reports back** whether move was successful or not.

TASK

Write a program that, given the ability to communicate with the robot, **determines the area of the room** into which the robot was teleported. The area of the room is the number of empty squares reachable from the robot's initial position, including the starting square. The area will be at most 1000 and you are allowed to use at most 5000 move commands.

INTERACTION

This is an interactive task. Your program sends commands to the robot using the standard output, and receives feedback from the robot by reading from the standard input.

To move the robot, you should output a single line containing one of the following commands: "`up`", "`down`", "`left`" or "`right`" (without the quotation marks), the direction in which the robot should move.

The robot will respond with a single line containing the word "`ok`" if the move was successful or the word "`fail`" otherwise.

When your program has found the solution, it should output a single line containing the area of the room and terminate its execution.

In order to interact properly with the grader, your program needs to flush the standard output after every write operation; the provided code samples show how to do this.

CODE SAMPLES

Code samples in all three languages are available for download on the "Tasks" page of the contest system. The code samples assume the room is rectangular and separately calculate the width and height of the room. This is not the correct solution and it will not score all points; the purpose of the samples is to show how to interact with the robot.

EXAMPLE

In the following example, commands are given in the left column, row by row. Feedback is given in the second column of the corresponding row.

output (command)	input (feedback)
up	fail
left	fail
right	ok
up	fail
right	ok
down	fail
up	fail
right	fail
left	ok
down	ok
down	fail
left	ok
left	fail
down	fail
5	

TESTING

You can use the TEST facility of the grading system to automatically run your solution with custom test cases. A single test case should be formatted as follows.

The first line of the test case should contain two integers R and C (1 ≤ R, C ≤ 500), the number of rows and columns, respectively.

The following R lines should contain C characters each. The character '.' represents an empty square, the character '#' represents a wall and the uppercase letter 'R' represents the starting location of the robot. There must be exactly one 'R' character on the map, and it needs to be inside of a room (i.e. it should be impossible for the robot to reach the edge of the map using valid moves).

Here is an example of a valid input file (corresponds to the example above). Notice that the robot is inside a room and that there is no way for it to move off the map.

```
6 8
........
.#####..
.#R..#..
.#..####
.###....
..#.....
```

Valid input for test facility

The grading system will provide you with a detailed log of the execution.