Buggy Robot

There is a robot in a 2D grid. The grid consists of empty cells and obstacles, and there is exactly one cell that is the exit. The robot will exit the grid if it ever reaches the exit cell. Empty cells are denoted as '**.**', the robot's initial position is denoted as '**R**', obstacles are denoted as '**#**', and the exit is denoted as '**E**'.

You can program the robot by sending it a series of commands. The series of commands is a string consisting of characters: '**L**' (move one square left), '**U**' (move one square up), '**R**' (move one square right), or '**D**' (move one square down). If the robot would run into an obstacle or off the edge of the grid, it will ignore the command (but it will continue on to future commands, if there are any).

Your friend sent a series of commands to the robot, but unfortunately, the commands do not necessarily take the robot to the exit.

You would like to fix the string so that the robot will reach the exit square (note once the robot reaches the exit, it stops, even if there are more commands in the string). You can fix the string with a sequence of operations. There are two operations: inserting a command anywhere in the string, or deleting a command anywhere in the string. What is the minimum number of operations you would need to fix the program?

Input
Each input will consist of a single test case. Note that your program may be run multiple times on different inputs. The first line of input contains two integers, *r* and *c* (2≤*r*,*c*≤50) which are the number of rows and number of columns of the grid. The next *r* lines will each contain a string with exactly *c* characters. Each character is one of '**.**' (Empty), '**R**' (the Robot), '**#**' (an Obstacle), or '**E**' (the Exit). This is the grid. There will be exactly one '**R**' and one '**E**' in the grid, and it will always be possible to navigate the robot to the exit. The last line of input will contain a string *s* (1≤|*s*|≤50) of commands. The string *s* will consist only of characters '**L**' (left), '**R**' (right), '**U**' (up) or '**D**' (down).

Output
Output a single integer, which is the minimum number of operations necessary to fix the command string so that the robot makes it to the exit.

Sample Input	Sample Output
3 3 R.. .#. ..E LRDD	1
2 4 R.#. #..E RRUUDDRRUUUU	0