

## Problem H. Non-Shortest Path

Input file:            nsp.in  
Output file:           nsp.out  
Time limit:            2 seconds  
Memory limit:         256 mebibytes

*In this problem, you have to travel from one corner of a checkered  $4 \times 4$  field to the opposite corner using any simple path which is not the shortest one.*

There is a checkered field on a plane consisting of  $4 \times 4$  cells. Each cell of the field is either empty or fully occupied by a wall.

Robot Aurelius is passing a test to measure the intelligence level by solving various problems on that field. He has already completed the parts which involved searching for an arbitrary path and the shortest path between a pair of cells.

The next task requires Aurelius to walk from the upper left corner of the field to the lower right corner by moving through empty cells. A single step consists in moving from the current cell to any other cell which shares a side with the current one. The path must be simple, that is, for each cell, the robot can not visit that cell more than once. Additionally, the length of the path must be strictly greater than the length of the shortest path between the corners on the given field. The length of the path is the number of steps on that path.

Write a program that will help robot Aurelius accomplish the task.

### Input

The input contains one or more test cases. Each test case is given on four lines. Each of these lines describes one row of the field and contains exactly four characters defining the cells of that row. An empty cell is denoted by character “.” (dot, ASCII code 46), and a wall is denoted by character “#” (hash, ASCII code 35). Consecutive test cases are separated by a line consisting of four characters “-” (dash, ASCII code 45).

It is guaranteed that the upper left cell and the lower right cell are empty. Additionally, it is guaranteed that all test cases in a single input are distinct.

### Output

For each test case, print a line defining the required path. The line must consist of characters corresponding to the commands for moving to an adjacent cell in the order of their execution: “D” to move down, “U” to move up, “L” to move left, and “R” to move right. If there are several possible paths, print any one of them. If there is no path with the required properties, print the number “-1” instead of the path.

## Example

nsp.in	nsp.out
.#..	DDRURURDDD
....	-1
..#.	-1
.#..	
----	
..##	
...#	
#...	
.#..	
----	
...#	
..#.	
.#..	
#...	

## Explanation

The example contains three test cases.

In the first test case, the length of the shortest path is six steps: the only shortest path is “DRRRDD”. Other than that, there are three simple paths which are not the shortest: these are the paths “DDRURRDD” and “DRRURDDD” of length 8, and the path “DDRURURDDD” of length 10. Any of them can be printed.

In the second test case, there are eight possible simple paths, but they all have the same length, and thus all are the shortest paths.

In the third test case, there is no path from the upper left corner of the field to the lower right one consisting of empty cells.