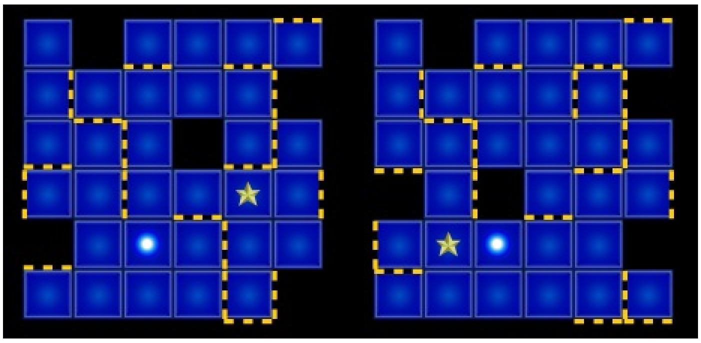
ACM-ICPC Live Archive

5008 Double Maze

Unlike single maze, double maze requires a common sequence of commands to solve both mazes. See the figure below for a quick understanding.



Reference: `http://www.onemorelevel.com/game/double_maze`

A maze is made up of 6×6 cells. A cell can be either a hole or a square. Moreover, a cell may be surrounded by barriers. There is ONLY one start cell (with a ball) and ONLY one end cell (with a star) in a single maze. These two cells are both squares. It is possible that the start cell and the end cell are the same one. The goal of a single maze is to move the ball from the start cell to the end cell. There are four commands in total, '`L`', '`D`', '`R`' and '`U`' corresponding to moving the ball left, down, right and up one cell, respectively. The barriers may make the commands take no effect, i.e., the ball does NOT move if there is a barrier on the way. When the ball gets to a hole or outside of the maze, it fails.

A double maze is made up of two single mazes. The commands control two balls simultaneously, and the movements of two balls are according to the rules described above independently. Both balls will continue to move simultaneously if at least one of the balls has not got to the end cell. So, a ball may move out of the end cell since the other ball has not been to the target. A double maze passes when both balls get to their end cells, or fails if either of the two mazes fails. The goal of double maze is to get the shortest sequence of commands to pass. If there are multiple solutions, get the lexicographically minimum one.

To simplify the input, a cell is encoded to an integer as follows. The lowest 4 bits signal the existence of the barriers around a cell. The fifth bit indicates whether a cell is a hole or not. The sixth and seventh bits are set for the start cell and end cell. Details are listed in the following table with bits counted from lowest bit. For a barrier, both of the two adjacent cells will have the corresponding barrier bit set. Note that the first two mazes in the sample input is the encoding of two mazes in the figure above, make sure you understand the encoding right.

Bit	Value	Description
1	0	No barrier to the left
	1	A barrier to the left
2	0	No barrier to the bottom
	1	A barrier to the bottom
3	0	No barrier to the right
	1	A barrier to the right
4	0	No barrier to the up
	1	A barrier to the up
5	0	A hole
	1	A square
6	0	Not start cell
	1	A start cell
7	0	Not end cell
	1	An end cell

Input

The first line of input gives the total number of mazes, T $(1 < T \leq 20)$. Then follow T mazes. Each maze is a 6×6 matrix, representing the encoding of the original maze. There is a blank line between mazes.

Output

For every two consecutive mazes, you should treat them as a double maze and output the answer. So there are actually $T - 1$ answers. For each double maze, output the shortest sequence of commands to pass. If there are multiple solutions, output the lexicographically minimum one. If there is no way to pass, output '-1' instead.

Sample Input

```
3
16 0 18 16 18 24
20 19 24 16 28 1
18 28 17 0 22 17
25 20 17 18 88 20
2 16 48 28 17 16
24 16 16 20 23 1

16 0 18 16 18 24
20 19 24 20 29 1
18 28 17 16 22 17
8 20 1 18 24 20
19 80 48 24 16 0
24 16 16 16 22 19

18 16 18 16 18 80
24 18 24 16 24 18
18 24 0 0 18 24
24 18 0 0 24 18
18 24 18 16 18 24
56 18 24 18 24 18
```

Sample Output

```
RRLULLLRRDLU
RURDRLLLURDULURRRRRDDU
```