Solving the Puzzle

The 15 puzzle is a classic puzzle made famous in the 19th century. It consists of 4x4 board with 15 sliding tiles numbered from 1 to 15. The objective is to get them into this pattern:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Here we will deal with a generalized version of the above puzzle. You should write a program that given some initial state of the nxn board finds a sequence of moves that transforms it so that in the i-th row there are tiles with numbers i*n+1,i*n+2,...,i*n+n (from left to right) - with the exception of the lower right corner where the hole should be. The less moves you use, the more points you get.

Input

The first line of input contains the number of test cases c (c<=200). Then c test cases follow, each of them begins with a line with a single integer n (3<=n<=10) in it. The next n lines describe the initial state of the board - the i-th line consists of exactly n integers describing the i-th row. The position of the hole is indicated by 0.

Output

For each test case output one line - the found sequence of moves. Write 'D' to move the hole down, 'U' to move it up, 'R' to move it right and 'L' to move it left. You shouldn't use more than 10000 moves. All moves should be valid (so for example don't try to move the hole up when it is in the first row).

Scoring

Your program will receive n^3/(m+1) points for each test case where m is the number of moves.

Example

Input:
2

4
1 2 7 3
5 6 0 4
9 10 11 8
13 14 15 12

3

0 1 2
4 5 3
7 8 6

Output:
URDDD
RRDD