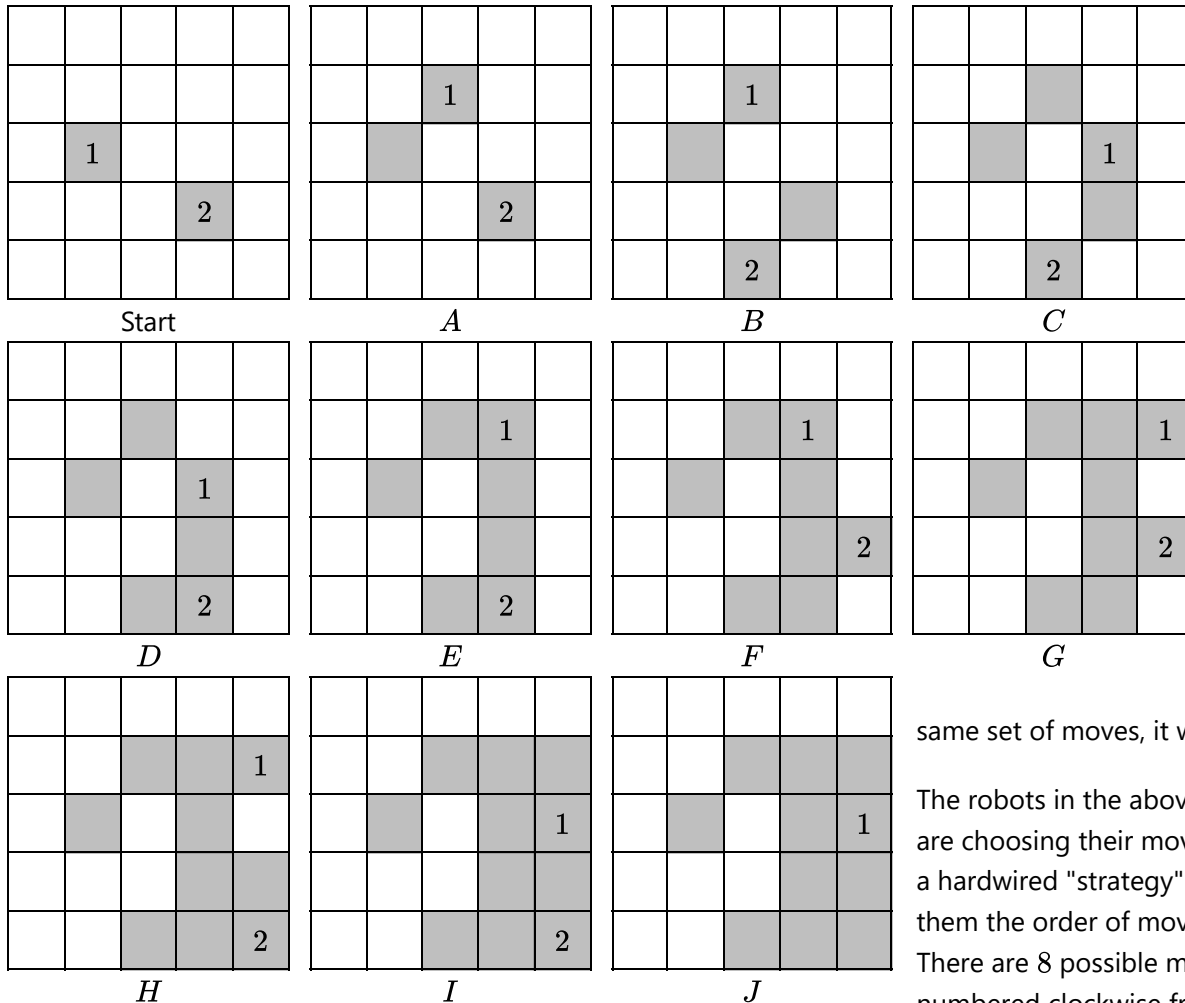


# ECOO '13 R2 P3 - Last Robot Standing

**Time limit:** 1.0s **Memory limit:** 64M

You are entering your robot into a simple game of strategy. The game is played on a grid and the robots take it in turns to move one space at a time (including diagonally) around the grid. When a robot moves into a square, that square is taken and nobody else can move into it for the duration of the game. When a robot is boxed in and cannot move on its turn, it is out. The winner is the Last Robot Standing.

Here's an example of the game in action on a small grid with 2 robots. The positions of the robots are marked 1 and 2 and the shaded squares are taken and cannot be used again by either robot.



In this example, Robot 1 is the Last Robot Standing. Note that Robot 2 was boxed in on Move (H), but was not removed from the game at that point.

Any robot that is able to move on its turn is safe for that turn. It is only on its next move (Move (J)) that it is removed from the game. Note that if Robot 2 had started out just one square to the left and used the same set of moves, it would have won.

The robots in the above game are choosing their moves from a hardwired "strategy" that tells them the order of moves to try. There are 8 possible moves, numbered clockwise from 1 to 8 as shown at

8	1	2
7		3
6	5	4

right. A robot's strategy can be represented as a list of numbers. Robot 1's strategy is `2481377746543`, and Robot 2's strategy is `62437428513`.

On Robot 1's first turn (A), it uses move 2, which is up and right. Then on its next turn (C) it uses move 4 (down and right). Then on its next turn (E) it tries move 8 (up and left) but is blocked, so it uses the next move in its strategy, which is 1 (up). If a Robot ever runs out of moves in its hardwired strategy, it simply starts again from the beginning. Note that this can only work if each of the 8 moves appears in each strategy at least once.

Using your evil hacking skills, you have found out the strategies of the other robots. Since you get to pick your starting position and you're moving last, you can figure out a winning start position.

The input will contain 10 cases. Each case starts with a line of 3 integers  $R$ ,  $C$ , and  $N$ .  $R$  and  $C$  are the number of rows and columns on the playing surface ( $4 \leq R, C \leq 10$ ) and  $N$  is the number of robots in the competition ( $2 \leq N \leq 16$ ). Your robot is the  $N^{\text{th}}$  one. Following this are  $N - 1$  lines containing the integer starting position for each of the other robots (row then column, numbered from 1 upwards), then  $N$  lines containing the strategy for each robot (including your own). During the game, the robots take turns in the order they appear in the file, with your robot moving last.

Your job is to output a list of starting positions that will lead to a guaranteed win for your robot. If there is no such starting position, you should simply output the word `LOSE`. Your output must be formatted exactly as shown below with no extra spaces or punctuation, but the order you output the starting positions for each case can be different from that shown.

Note that the sample input below only contains 4 cases, but the data files will contain 10 cases each.

## Sample Input

---

```
5 5 2
3 2
2481377746543
62437428513
3 3 3
1 1
2 2
45362718
45362718
87654321
5 5 3
1 1
3 3
81726354
45362718
12345678
10 10 8
1 3
1 8
3 1
8 1
3 10
8 10
10 3
12345678
87654321
18273645
54637281
21436587
78563412
16372485
73265841
```

## Sample Output

---

```
(1,2) (1,3) (2,3) (2,4) (3,1) (4,5) (5,2) (5,5)  
(3,3)  
(3,2) (4,2)  
(1,9) (4,9) (4,10) (5,7) (7,1) (7,2) (7,6) (7,7) (9,4)
```

Educational Computing Organization of Ontario - statements, test data and other materials can be found at [ecocs.org](http://ecocs.org)