

Yet Another Contest 1 P3 - Fluke

Time limit: 4.0s **Memory limit:** 256M

Nils is sick of his moments of strategic brilliance being called flukes! To prove himself, he will challenge Josh to the following game.

The game will be played on an N by N grid of initially unmarked cells. Both the rows and the columns of the grid are numbered from 1 to N . Denote the cell in the r -th row and the c -th column as cell (r, c) .

Nils and Josh will take alternating turns. In each turn, the player must select an unmarked cell **which does not share an edge with any marked cell**. They will then mark this selected cell. The last player to make a valid move is the winner.

In order to prove that his victories are a result of his strategic brilliance rather than just flukes, Nils must win T consecutive games. To make things a little fairer, Nils will be allowed to decide which player takes the first turn. Can you help Nils win all T games?

Constraints

$$1 \leq N, T \leq 100$$

Subtask 1 [20%]

$$1 \leq N \leq 3$$

Subtask 2 [80%]

No additional constraints.

Interaction

This is an interactive problem, where the interactor will play the role of Josh. Since the moves that the interactor can make depends on your moves, **the interactor will be adaptive**.

At first, you should read in two space separated integers on a single line, denoting the values of N and T respectively. Note that the value of N is the same for all T games.

Each of the following T interactions represents a single game.

At the beginning of each interaction, on a single line, you should output either `1` or `2`. Outputting `1` indicates that Nils will take the first turn, whilst outputting `2` indicates that Josh will take the first turn. If you do not do so, you will be responsible for terminating your program to receive a Wrong Answer verdict; otherwise, your program will receive an arbitrary verdict.

Then, the game will begin. When it is Nils' turn, on a single line, output your selected cell in the format `r c`, representing that Nils will mark the cell (r, c) . When it is Josh's turn, you should read in a line of input in the format `r c`, representing that Josh will mark the cell (r, c) .

If Josh is unable to make a valid move, then the interactor will respond with `0 0` on a single line. After receiving this feedback, if you have not yet played all T games, then you should begin the next interaction for the following game.

If at any point you make an invalid move (such as an invalid output format or an invalid cell), then instead of the interactor responding with a cell to mark, the interactor will respond with `-1 -1` on a single line. After receiving this feedback, you should terminate your program to receive a Wrong Answer verdict; otherwise, your program will receive an arbitrary verdict.

If Nils has no valid moves to play when it is his turn, then you should also terminate your program to receive a Wrong Answer verdict.

If you manage to win all T games, then you will receive an Accepted verdict for that test case.

Please note that you may need to flush `stdout` after each operation, or interaction may halt. In C++, this can be done with `fflush(stdout)` or `cout << flush` (depending on whether you use `printf` or `cout`). In Java, this can be done with `System.out.flush()`. In Python, you can use `sys.stdout.flush()`.

Sample Interaction

`>>>` denotes your output. Do not print this out.

```
2 1
>>> 2
1 1
>>> 2 2
0 0
```

Explanation

Here, the game is played on a 2 by 2 grid, and Nils only has to win a single game. Nils chooses for Josh to go first.

Josh begins by choosing the cell (1, 1), and Nils follows by choosing the cell (2, 2). Note that this is allowed as the cells do not share an edge, even though they share a corner. At this point, Josh is unable to make a move, so Nils wins.