

Problem G. Battleship: New Rules

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

This is an interactive problem.

Ivan came up with new rules for the battleship game!

- The game will be played on an $n \times n$ board.
- The first player chooses an integer k ($n \leq k \leq \lceil \frac{n}{2} \rceil^2$).
- After that, the first player places k ships on the board so that the number of cells occupied by the ships is the maximum possible (among all valid placements of k ships of any sizes).
- Each ship should be a rectangle of size $1 \times a$ or $a \times 1$ (a is any integer from 1 to n inclusive). Any two ships should not have neighbouring cells (by side or by corner).

After that, the second player starts his game.

- The second player knows only the size of the board n .
- The second player can ask a query: is cell (x, y) occupied by some ship?
- The second player should find any empty 2×2 square on the board, or say that there are no such squares.

The second player can ask at most $6n$ queries. Please play as the second player and win the game!

Interaction Protocol

The first line contains a single integer t ($1 \leq t \leq 100$) — the number of games to be played. You should play t games and finish interaction after that.

At the start of the game, you are given a single integer n ($3 \leq n \leq 1000$) — the size of the board.

After that, you can ask some queries. To ask a query, print a single line “? x y ” ($1 \leq x, y \leq n$) — the coordinates of the cell. You will be given an answer c :

- If $c = -1$, you made too many queries. You should terminate your program.
- If $c = 0$, the cell (x, y) is empty.
- If $c = 1$, the cell (x, y) is occupied by some ship.

To finish the game, print a single line “! x y ”, where:

- $x = -1, y = -1$ if there are no empty 2×2 squares on the board.
- Otherwise, $1 \leq x, y \leq n - 1$ and the square with cells $(x, y), (x + 1, y), (x, y + 1), (x + 1, y + 1)$ is empty.

If your answer is incorrect, you will be given a line with value -1 , and you should terminate your program. Otherwise, you will be given a line with value 1 , and you should play the next game (or finish your program if it was the last game).

It is guaranteed that the sum of n for all games does not exceed 5000.

It is guaranteed that the board in each game is fixed, and the interactor is not adaptive.

Your solution will get **Idleness Limit Exceeded** if you don't print anything or forget to flush the output.

Example

standard input	standard output
2	
3	
0	? 2 1
1	
4	! -1 -1
0	
1	
4	? 1 3
0	
1	? 4 3
1	
1	! 2 2

Note

Boards from the first test are shown on pictures below. Rows correspond to x coordinates, columns correspond to y coordinates.

	1	2	3
1	1	1	1
2	0	0	0
3	1	0	1

Board from the first game.

	1	2	3
1	1	1	1
2	0	0	0
3	1	0	1

Board from the second game.

In the first game, there are no empty 2×2 squares on the board.

In the second game, there is exactly one empty 2×2 square on the board.