

Problem A. Circular Search

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

This is an interactive problem.

Little Liza studies mathematics on an educational website. She just solved another bunch of problems, and, as usual, a new game was revealed as a reward. This time, the game is as follows.

A coordinate plane is drawn on the screen. The drawing outlines a square with opposite corners in points $(0, 0)$ and (n, n) . The game secretly selects a point with integer coordinates in this square or on its border. The player has to find the secret point. To do this, the player can put a circle on the plane by selecting a point with integer coordinates as its center and a non-negative integer as its radius. After that, the game tells the player whether this circle contains the secret point, either inside or on its border. The game ends when the player puts a circle with radius 0 right at the secret point.

Liza played the game a few times, and after a long round she wondered: how to find the secret point fast when n is large?

Solve a generalized version of Liza's problem. Knowing the number n and putting circles on the plane, find the secret point fast enough.

Interaction Protocol

First, an integer n is given on a separate line ($1 \leq n \leq 10\,000$). After that, the player puts from 0 to 50 circles.

To put the next circle, print three integers on a separate line: " $x y r$ ". Here, x and y are the coordinates of the circle center, and r is its radius ($-10^9 \leq x, y \leq +10^9$, $0 \leq r \leq 10^9$). As an answer, the player gets a line with one word: "Yes" if the secret point is inside or on the border of the circle, and "No" otherwise.

After getting a "Yes" with a circle of radius 0, terminate the program gracefully. If, after putting 50 circles, this does not happen, the system attempts to terminate the run with "Wrong Answer" outcome.

To prevent output buffering, flush the output buffer after each printed line: this can be done by using, for example, `fflush (stdout)` in C or C++, `System.out.flush ()` in Java, `flush (output)` in Pascal, or `sys.stdout.flush ()` in Python.

In each test, the coordinates of the secret point are **fixed in advance** and do not change during the game.

Example

standard input	standard output	secret point
1		(1, 1)
No	0 0 1	
Yes	1 1 1	
Yes	1 1 0	