Problem L: Lookup Table Tennis

Time limit: 1 second

Table Tennis at the 2012 Summer Olympics.
CC BY-SA 2.0 by Joel Solomon on Wikimedia Commons

For this year's Olympic tournament, the International Table Tennis Federation (ITTF) is pioneering a new ball tracking technology, intended to provide the live broadcast with additional data and insights. In the software, the playing area surrounding the table is overlaid with a three-dimensional grid of dimensions $n \times n \times n$. At any time, the position of the table tennis ball is assumed to be at an integer location within this grid, that is, at a position (x, y, z) where $0 \leq x, y, z \leq n$ are integers.

The system was created using the latest and greatest AI driven technology, which unfortunately means that none of the engineers behind it understand the details of its inner workings. In particular, it is not even possible to directly obtain the location of the table tennis ball from the computer! Instead, the only way to interact with the system is to ask queries consisting of a position and a distance, to which the system will respond whether the table tennis ball is at most the given distance away from the given position. In other words, each query is in the shape of a ball, and the system will tell you whether the table tennis ball is within the query ball.

The engineers now plan to use multiple of these queries to determine the location of the table tennis ball, and have asked you to assist them in doing so. Note that because the ball tracking needs to be done in real time you should never use more than 5000 queries to locate the table tennis ball.

Interaction

This is an interactive problem. Your submission will be run against an *interactor*, which reads from the standard output of your submission and writes to the standard input of your submission. This interaction needs to follow a specific protocol:

The interactor first sends one line with an integer n ($1 \leq n \leq 10^6$), the size of the grid. The table tennis ball is at some point (x_0, y_0, z_0) with integer coordinates $0 \leq x_0, y_0, z_0 \leq n$. For simplicity, its size is considered to be negligible.

Then, you need to start asking queries consisting of four integers "$x \ \ y \ \ z \ \ s$", where x, y and z ($0 \leq x, y, z \leq n$) are the centre coordinates and s ($0 \leq s \leq 3 \cdot n^2$) is the *square* of the radius of your query ball. The interactor will respond with 1 if (x_0, y_0, z_0) is within the query ball (including the boundary) and 0 otherwise. In other words, the answer will be 1 if and only if $(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 \leq s$.

When you ask the query "$x_0 \ \ y_0 \ \ z_0 \ \ 0$", the interactor will respond 1 and the interaction ends. Your program must then exit.

You may send at most 5000 queries; any more than that results in a "Wrong Answer" verdict.

Interaction is not adaptive; the location (x_0, y_0, z_0) is fixed at the start and does not change.

After every request you should *flush* the standard output to ensure that the request is sent to the interactor. For example, you can use `fflush(stdout)` in C++, `System.out.flush()` in Java, `sys.stdout.flush()` in Python, and `hFlush stdout` in Haskell.

A testing tool is provided to help you develop your solution.

Read	Sample Interaction 1	Write
5		
		0 0 2 25
1		
		0 0 2 24
0		
		2 3 2 1
0		
		2 3 2 3
1		
		3 4 2 1
1		
		3 4 2 0
1		

Read	Sample Interaction 2	Write
1		
		0 0 0 1
1		
		1 1 1 1
0		
		1 1 1 2
1		
		1 0 0 0
0		
		0 1 0 0
1		