

Problem A. DIY Radar

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

This is an interactive problem.

Pin is an avid inventor who creates awesome mechanisms and robots from junk. While constructing his new junkyard worker robot, Pin made a mistake of activating the robot's AI earlier than usual, which made the robot unpredictably run away into the junkyard.

Pin wants to go into the junkyard to pick the robot up, but the area is just too big. The robot ran away before having a GPS tracker installed, so Pin has no information about the robot's exact coordinates. If only there was a way to get a rough estimate of where the robot could be...

Pin remembered the robot had an autonomous radio repeater in one of its parts. If Pin would send a specific radio signal, the reply would give him at least some important information: the distance between him and the robot, which can be used for triangulation.

Pin's junkyard is a huge square with its opposite corners lying at points $(0, 0)$ and $(100, 100)$, and small paths run parallel to square sides between junk piles, intersecting at integer points. Robot's AI is programmed to situate himself in a random path intersection at the junkyard (including the junkyard's border) and work around there. Unfortunately for Pin, a security algorithm in the rogue robot complicates the task. Every time the robot detects a searching radio signal, it would move to an adjacent intersection. Pin decided that it would be easier and more sensible to find the intersection where the robot initially was, and then search for it through a number of surrounding intersections.

Pin does not want to step into the junkyard before knowing where the robot is, so he will use his radio from the area around it. But he will not go beyond his privately owned area, which contains every point that is no further than 1000 vertically or horizontally from the junkyard's border.

Interaction Protocol

Your solution can print lines in the following formats:

- “Q X Y”: ask what is the distance between the point (X, Y) , which lies outside the junkyard but within the privately owned area, and the robot. The system prints a single line with an integer R^2 : the requested distance, squared. After that, the robot immediately moves to an adjacent intersection.
- “A X Y”: claim that the robot was at the point (X, Y) initially, before the measuring started. After printing this line, the solution must immediately terminate gracefully.

The total number of printed lines of both types must not exceed 5.

Example

standard input	standard output
25	Q 101 101
4	Q 97 101
	A 97 98

Note

The example shows a possible interaction between a solution and the testing system. Blank lines in the example are added there only to show the order of events.

Do not forget to flush the buffer after each line you output! Here are some functions which do it in different languages:

- C++: `fflush(stdout)` or `cout.flush()`



- **Python:** `sys.stdout.flush()`
- **Java:** `System.out.flush()`