Funny Chase

This is an interactive problem.

The Fornax Diamond, the most precious gem in the galaxy, has been stolen by Bash, the most famous thief in the galaxy! Alexia, an officer of the Intergalactic Police, is now chasing him. The galaxy is a Cartesian plane, and both Alexia and Bash always stay at points (x, y) with integer coordinates satisfying $|x| \le 10^9$ and $|y| \le 10^9$.

Alexia has a portal gun, which can take her to **any** point of the galaxy. Bash escapes with a portal gun of an older model, which is broken: it always takes him to a **random** point. Due to its low battery, Bash must alternate between a *long jump* and a *short jump*. More precisely, Bash's jumps alternate as follows:

- For the 1-st, 3-rd, 5-th, ... jump (a *long jump*), Bash moves to a point chosen uniformly at random among all points of the galaxy.
- For the 2-nd, 4-th, 6-th, ... jump (a *short jump*), if Bash is at (x, y), he moves to a point (x', y') chosen uniformly at random among all points of the galaxy with $|x - x'| + |y - y'| \le 10^6$.

All random choices are independent, and Bash's jumps do not depend on Alexia's moves.

Fortunately, the Intergalactic Police had hidden a tracking device inside the Fornax Diamond to protect it. Let (x_A, y_A) and (x_B, y_B) be the positions of Alexia and Bash. At the start of the chase, whenever Alexia moves, and whenever Bash makes a long jump, Alexia learns the **square** of the Euclidean distance between them, i.e., $(x_A - x_B)^2 + (y_A - y_B)^2$. However, with its battery low, Bash's old portal gun runs in power-saving mode during a short jump and leaks so much electromagnetic noise that it jams the tracker's signal, so the tracker can only estimate the distance roughly: whenever Bash makes a short jump, the Euclidean distance $\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$ is first **rounded to the nearest multiple of** $1\,000$ (if the distance is exactly between two multiples of $1\,000$, it is rounded up), then Alexia only learns the **square** of this rounded distance. For example, the distance $\sqrt{4\,020\,041} \approx 2\,005.004$ is rounded to $2\,000$, and Alexia learns $2\,000^2 = 4\,000\,000$.

Initially, Alexia is at $(0, 0)$ and Bash is at a point chosen uniformly at random among all points of the galaxy. Then Alexia and Bash take turns, starting with Alexia:

- Alexia moves to any point of the galaxy (staying in her current position is allowed).
- Bash makes his next jump.

Alexia catches Bash by moving to the point where Bash is. Note that if Bash jumps to Alexia's position, he is not caught yet; he can only be caught in Alexia's next turn if she chooses to stay in her current position. Help Alexia catch Bash using at most $2\,000$ moves.

Interaction Protocol

First, your program reads the number of test cases τ ($1 \le \tau \le 100$). For each test case:

- First, your program reads an integer d – the square of the distance between Alexia's initial position $(0, 0)$ and Bash's initial position.
- Then, your program makes **at most** $2\,000$ moves. For each move:
  - Your program prints two integers x and y ($|x| \le 10^9$; $|y| \le 10^9$), meaning that Alexia moves to (x, y).

– Your program reads an integer d – the square of the distance between Alexia and Bash after Alexia's move. If $d = 0$, Alexia has caught Bash, and your program should continue with the next test case, or terminate if this is the last one.
– Otherwise, Bash jumps, and your program reads an integer d – the square of the distance between Alexia and Bash if this is a long jump, or the square of the distance rounded to the nearest multiple of 1 000 if this is a short jump.

If your program prints an invalid move, or Alexia has not caught Bash after the 2 000-th move, the interactor prints -1 instead of the distance. In this case, your program should terminate immediately to receive a *Wrong Answer* verdict.

After printing each query do not forget to output the end of line and flush[1] the output.

Sample Interaction

Stdin	Stdout	Comment
2		There are two test cases.
		The first test case starts: Alexia is at $(0, 0)$.
25		Bash is at $(3, -4)$. Squared distance $= (0-3)^2 + (0-(-4))^2 = 25$.
	4 3	Alexia moves to $(4, 3)$.
50		Bash is not there. Squared distance $= (4-3)^2 + (3-(-4))^2 = 50$.
170		Bash makes a long jump to $(-7, 10)$. Squared distance $= (4-(-7))^2 + (3-10)^2 = 170$.
	-7 10	Alexia moves to $(-7, 10)$.
0		Alexia catches Bash. Squared distance $= (-7-(-7))^2 + (10-10)^2 = 0$.
		The second test case starts: Alexia is at $(0, 0)$.
4073		Bash is at $(37, -52)$. Squared distance $= (0-37)^2 + (0-(-52))^2 = 4\,073$.
	1 -1	Alexia moves to $(1, -1)$.
3897		Bash is not there. Squared distance $= (1-37)^2 + (-1-(-52))^2 = 3\,897$.
106		Bash makes a long jump to $(6, 8)$. Squared distance $= (1-6)^2 + (-1-8)^2 = 106$.
	2 3	Alexia moves to $(2, 3)$.
41		Bash is not there. Squared distance $= (2-6)^2 + (3-8)^2 = 41$.
4000000		Bash makes a short jump to $(6, 2008)$. Distance $= \sqrt{(2-6)^2 + (3-2008)^2} \approx 2\,005.004$, rounded to $2\,000$. Squared rounded distance $= 2\,000^2 = 4\,000\,000$.
	7 2008	Alexia moves to $(7, 2008)$.
1		Bash is not there. Squared distance $= (7-6)^2 + (2008-2008)^2 = 1$.
3984160		Bash makes a long jump to $(-5, 12)$. Squared distance $= (7-(-5))^2 + (2008-12)^2 = 3\,984\,160$.
	-5 12	Alexia moves to $(-5, 12)$.
0		Alexia catches Bash. Squared distance $= (-5-(-5))^2 + (12-12)^2 = 0$.

The sample interaction only shows how your program interacts with the interactor, and does not show how the answer is deduced.

[1]To flush, use: `fflush(stdout)` or `cout.flush()` in C++; `sys.stdout.flush()` in Python; or see the documentation for other languages.