Problem I. Interactive Interception

Input file: `standard input`
Output file: `standard output`

This is an interactive problem.

North Eastern Emergency Rocket Control agency (NEERC) has developed a new radar control system that is designed to better control ballistic rocket interception. To test the new system NEERC agency had developed a mathematical model that is intended to show this system's abilities.

Let us represent a rocket as a point on a line. Initially the point is at some unknown integer location between 0 and p, inclusive. It has some unknown speed of q which is an integer between 0 and v, inclusive.

Each second the following happens. First, the control system makes a query to the radar of a form "`check` L R" and gets an answer whether the point is currently between L and R, inclusive, or not. After that, the point's coordinate increases by q.

The goal of the radar control system is to learn the exact location of the point at the beginning of some second. When it does learn the point's location, then instead of making a query to the radar, it gives a command to intercept the point at that location.

You have to implement the control system that locates and intercepts the point while making at most 100 queries to the radar.

Interaction protocol

Interaction starts with your program reading two integers — the values of p and v from the standard input ($1 \le p \le 10^5$, $1 \le v \le 10^5$).

After that your program must print commands to the standard output. Each command must be one of the following two.

- "`check` L R" — make a query to the radar to get an answer whether the point is currently between L and R, inclusive, or not. The answer must be read from the standard input and is either "`Yes`" or "`No`". After that the point's coordinate is increased by q. L and R must be integers ($0 \le L \le R \le 10^9$).

  There must be at most 100 "`check`" commands.

- "`answer` x" — the exact coordinate x of the point is known, and you order to intercept the point. After printing this command your program must exit.

Your program must write end-of-line sequence and flush the standard output after each command, including the last command "`answer` x" (end-of-line must be written and flushed before exiting).

Sample input and output

standard input	standard output
2 2 Yes No Yes Yes	check 1 3 check 3 5 check 2 4 check 4 5 answer 5

In the given example the point was initially at location 1 and is moving at a speed $q = 1$.