Guess The Number With Lies v3

GUESSN3

Judge has chosen an integer x in the range [1, n]. Your task is to find x, using query as described below. But be careful, because the Judge can lie. Judge is allowed to lie only once in single game and only in reply for query.

Query

Single query should contains range R = [a, b]. The reply for query is "YES", if a ≤ x ≤ b. Otherwise the reply is "NO".

$1 \le a \le b \le n$

Communication

You should communicate with Judge using standard input and output.

Attention: the program should clear the output buffer after printing each line. It can be done using fflush(stdout) command or by setting the proper type of buffering at the beginning of the execution - setlinebuf(stdout).

The first line of input contains single integer T, the number of games. Then T games follow.

At the beggining of each game You should send to the Judge a line with command "START_GAME". The Judge will answer You with numbers n, m, where n is as described above and m is the maximum number of queries that You can use in this game.

Then You should send some queries, every query is a line with "QUERY" keyword, then single-space separated values a b. After each query the Judge will answer "YES" or "NO".

At the end of game You should give answer: "ANSWER y", where $1 \le y \le n$ is the answer for the game. When y ≠ x, the solution will got WA.

Then start the next game (if there is any).

Constraints

$1 \le T \le 2^{10}$

$2 \le n \le 10^{18}$

m: This value will be as small as possible. This means, that there exists strategy for player to find the value x using m queries, but there doesn't exist such strategy using m-1 queries.

Example

The example of communication. J=Judge, P=Player.

J: 3

P: START_GAME
J: 2 3

P: QUERY 1 1
J: YES
P: QUERY 2 2
J: YES
P: QUERY 1 1
J: NO
P: ANSWER 2

P: START_GAME
J: 2 3

P: QUERY 1 1
J: YES
P: QUERY 2 2
J: NO
P: ANSWER 1

P: START_GAME
J: 5 6

P: QUERY 1 3
J: YES
P: QUERY 4 5
J: YES
P: QUERY 2 4
J: YES
P: QUERY 3 4
J: NO
P: ANSWER 2

Explanation:

In 1st game there is conflicting information in first and second query, so we know, that the last query isn't lie.

In 2nd game the Judge don't use lie (Judge can lie once, but don't need to do it).

In 3rd game there is conflicting information in first and second query, so the next queries has correct answers.

Note

Be careful. The Judge will try to maximize the number of queries that You will ask. If necessary, the Jugde can also replace chosen value x with the other one. But don't worry too much - at the end of the game, the value x chosen by Judge will satisfy all except at most one of Your queries.

Similar problems

There is a family of similar problems. Here is the table with them:

Code	Number of lies	Query format	Type	Section	Difficulty
GUESSN1	1	subset	interactive	challenge	easy
GUESSN2	2-16	subset	interactive	challenge	medium/hard
GUESSN3	1	range	interactive	classical	medium
GUESSN4	1	subset	non-interactive	challenge	medium
GUESSN5	2-16	subset	non-interactive	challenge	hard

subset - the query is about any subset of {1,2,...,n}
range - the query is about any range [a,b]
 interactive - the Judge replies after every query
non-interactive - all queries have to be asked at once, before any reply