

Guess The Number With Lies v1

GUESSN1

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 set $S = \{a_1, a_2, \dots, a_k\}$. The reply for query is "YES", if x is in S . Otherwise the reply is "NO".

$$1 \leq k < n$$

$$1 \leq a_1 < a_2 < \dots < a_k \leq n$$

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 beginning 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 $k a_1 a_2 \dots a_k$. After each query the Judge will answer "YES" or "NO".

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

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

Scoring

Total score is the sum of scores of single games. If You use c queries in game, Your score is c^2 . The smaller score is the better score.

Constraints

$$1 \leq T \leq 2^7$$

$$2 \leq n \leq 2^{17}$$

Example

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

J: 3

P: START_GAME

J: 2 10

P: QUERY 1 1

J: YES

P: QUERY 1 2

J: YES

P: QUERY 1 1

J: NO

P: ANSWER 2

P: START_GAME

J: 2 10

P: QUERY 1 1

J: YES

P: QUERY 1 2

J: NO

P: ANSWER 1

P: START_GAME

J: 5 25

P: QUERY 3 1 3 5

J: YES

P: QUERY 2 2 4

J: YES

P: QUERY 3 1 2 3

J: YES

P: QUERY 2 1 2

J: NO

P: ANSWER 3

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.

The score is $3^2 + 2^2 + 4^2 = 9 + 4 + 16 = 29$

Note

Be careful. The Judge will try to maximize the number of queries that You will ask. If necessary, the Judge 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.

Note 2

If You got SIGXFSZ error, You probably use unnecessary numbers in queries. Let's see at the example:

P: START_GAME

J: 16 9

P: QUERY 2 1 2

J: NO

P: QUERY 3 1 2 3

J: NO

P: QUERY 4 1 2 3 4

J: NO

P: QUERY 5 1 2 3 4 5

J: NO

P: QUERY 6 1 2 3 4 5 6

J: NO

In 3rd query, there are unnecessary numbers 1 and 2. This numbers cannot be the answer for this game, because the Judge said twice (in 1st and 2nd query's reply) "1 and 2 are not OK!", but the Judge can lie only once. From the same reason in 4th query, the unnecessary numbers are 1, 2 and 3. In 5th query, unnecessary number are 1, 2, 3 and 4. When n is big enough, the profit from this optimization is huge (and probably SIGXFSZ won't appear).

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 $\{1,2,\dots,n\}$

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