

# Guess The Number With Lies v2

## GUESSN2

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 is a liar. Judge is allowed to lie up to  $w$  times 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  $n, w, m$ , where  $n, w$  are 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  $0 \leq y \leq n$ ;  $y=0$  means, that You skip this game without the correct answer. Otherwise  $y$  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 and You find the  $x$  value, Your score is  $c^2$ . If You skip the game, the score is  $4m^2$ . The smaller score is the better score.

### Constraints

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

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

$$2 \leq w \leq 2^4$$

$$1 \leq w * n \leq 2^{19}$$

## Example

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

J: 3

P: START\_GAME

J: 2 2 10

P: QUERY 1 1

J: YES

P: QUERY 1 1

J: YES

P: QUERY 1 1

J: YES

P: ANSWER 1

P: START\_GAME

J: 2 4 10

P: QUERY 1 2

J: YES

P: QUERY 1 2

J: YES

P: QUERY 1 1

J: YES

P: QUERY 1 1

J: YES

P: QUERY 1 2

J: YES

P: QUERY 1 2

J: YES

P: QUERY 1 2

J: NO

P: QUERY 1 1

J: NO

P: ANSWER 2

P: START\_GAME

J: 12345 7 100

P: ANSWER 0

Explanation:

In 1st game Judge said 3 times, that his number is 1 and he didn't lie. The answer is 1, because he can lie only 2 times.

In 2nd game the Judge lied in 3rd, 4th and 7th query.

In 3rd game the Player gave up. The score is  $4 * 100^2$ .

The score is  $3^2 + 8^2 + 4 * 100^2 = 9 + 64 + 40000 = 40073$

## 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  $w$  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 2 14

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 4th query, there are unnecessary numbers 1 and 2. This numbers cannot be the answer for this game, because the Judge said three times (in 1st, 2nd and 3rd query's reply) "1 and 2 are not OK!", but the Judge can lie only 2 times. From the same reason in 5th query, the unnecessary numbers are 1, 2 and 3. 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
<a href="#">GUESSN1</a>	1	subset	interactive	challenge	easy
<a href="#">GUESSN2</a>	2-16	subset	interactive	challenge	medium/hard
<a href="#">GUESSN3</a>	1	range	interactive	classical	medium

<a href="#">GUESSN4</a>	1	subset	non-interactive	challenge	medium
<a href="#">GUESSN5</a>	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