Interactive Problem with Two Sorted Arrays

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 mebibytes

Alice has two arrays a and b each having n integers, both sorted in **increasing order**. All of the 2n elements are pairwise distinct. Her favorite integer is kth smallest integer among the $2n$ integers.

Alice tells you the length of the arrays but keeps the elements hidden. You have to guess the position of her favorite integer. You can ask her queries of the following type.

- `? i j` where $1 \le i, j \le n$. Alice tells you if $a_i < b_j$ or not.

Can you find the position of her favorite integer using no more than Q queries? In particular, you have to find out which array the kth smallest integer belongs to and what index it is at.

Interaction Protocol

- The first line of the input contains a single integer t ($1 \le t \le 20$) — the number of testcases.
- At the beginning of each testcase you should read from the input two integers, n ($1 \le n \le 500$) and k ($1 \le k \le 2n$).
- To ask a question print `? i j` on a separate line. i and j should be between 1 and n (inclusive).
- After printing a query you must flush the output stream. You may write `fflush(stdout)` or `cout.flush()` for this purpose.
- After printing a query, you should read a single integer from standard input. The response will be 0 if $a_i > b_j$ and 1 if $a_i < b_j$.
- You may ask at most Q such questions. See scoring section for the value of Q. After asking at most Q questions you have to guess the position. Print `! 1 i` if the kth smallest number is a_i, otherwise print `! 2 j` if the kth smallest number is b_j. Again you must flush the output stream after printing the answer.
- After printing the answer, the program should terminate if the current testcase is the last one, otherwise it should continue on to the next testcase.
- At any moment if the program reads -1 it should immediately terminate gracefully (for example, by calling `exit(0)`). It means that you made an invalid query or you made too many queries or you made an incorrect guess. In this case you'll receive a proper verdict. If your program continues to run even after reading -1, it may lead to unexpected verdicts.

Scoring

1. Subtask 1 (4 points): $Q = 1$, $k = 1$.
2. Subtask 2 (28 points): $Q = 500$.
3. Subtask 3 (68 points): $Q = 13$. For this subtask, your score for each testfile is calculated as follows. Let q be the maximum number of queries made in any of the testcases in the file.
   - If $q \le 10$, your score is 68.
   - If $10 < q \le 13$, your score is $17(14 - q)$.
   - If $q \ge 14$, your score is 0.

   Note that your score for each subtask is the minimum of the scores for the test cases in the subtask.

Example

standard input	standard output
2 1 2 1 1 2 3 0 1 0 0	? 1 1 ? 1 1 ! 2 1 ? 1 1 ? 1 2 ? 2 1 ? 2 2 ! 2 2

Explanation

- In the first test case, the arrays are `a = [1]` and `b = [2]`.
- In the second test case, the arrays are `a = [2, 4]` and `b = [1, 3]`.