Problem I. Intriguing Selection

Time limit: 5 seconds

This is an interactive problem.

You are the head coach of a chess club. The club has $2n$ players, each player has some *strength* which can be represented by a number, and all those numbers are distinct. The strengths of the players are not known to you.

You need to select n players who would represent your club in the upcoming championship. Naturally, you want to select n players with the highest strengths.

You can organize matches between the players to do that. In every match, you pick two players, they play some games, and you learn which one of the two has higher strength. You can wait for the outcome of a match before deciding who will participate in the next one.

However, you do not want to know *exactly* how those n players compare between themselves, as that would make the championship itself less *intriguing*. More formally, you must reach a state where there is exactly one way to choose n players with the highest strengths that is consistent with the outcomes of the matches you organized, but there must be at least two possible orderings of those n players by strength that are consistent with the outcomes of the matches you organized.

Interaction Protocol

Your program has to process multiple test cases in one run. First, it should read the integer t ($t \geq 1$) — the number of test cases. Then, it should process the test cases one by one.

In each test case, your program should start by reading the integer n ($3 \leq n \leq 100$) — the number of players to select out of $2n$ players. The sum of squares of the values of n over all test cases does not exceed 10 000.

Then your program can organize matches zero or more times. To organize a match, your program should print a match description formatted as `?` i j — a question mark followed by two distinct numbers of players participating in the match. The players are numbered from 1 to $2n$, inclusive. Remember to flush the output after printing the match description. Then your program should read the match outcome — it will be either the greater-than character (`>`), if the first player in the match description has higher strength, or the less-than character (`<`), if the second player in the match description has higher strength.

Your program can organize at most $4n^2$ matches. After it is done organizing matches, it should print the exclamation mark (`!`) and continue to the next test case, or exit gracefully if this was the last test case. Remember to flush the output after printing the exclamation mark.

There must be exactly one way to choose n players with the highest strength that is consistent with the outcomes of the matches you organized, but there must be at least two possible orderings of those n players by their strength that are consistent with the outcomes of the matches you organized.

The judging program picks some distinct numbers as the strengths of all players before your program starts organizing matches and uses them to answer the requests.

Example

standard input	standard output
2 3 > < > < > > 3 < < < > >	 ? 1 3 ? 4 2 ? 4 5 ? 6 5 ? 3 4 ? 5 6 ! ? 3 4 ? 4 2 ? 5 3 ? 6 4 ? 3 1 !

Note

In the example, the players in the first test case are sorted by strength in decreasing order. From the matches in the example output, we can deduce that players 1, 2, and 3 have the highest strength, but we do not know how the player 1 compares to the player 2.