

Problem L. Lazy Judge

Input file: *standard input*
Output file: *standard output*
Time limit: 4 seconds
Memory limit: 512 mebibytes

This is an interactive problem.

The judges are working on the strategy for the jury program for the modified version of the problem **J** from the current contest.

In that problem, Alice secretly invents a permutation of first N integers a_1, a_2, \dots, a_N and tells N to Bob. Bob asks some questions to identify this permutation. Alice may change the permutation in the process as long as it is consistent with her previous answers.

The judges are planning to create an AliceBot that will do the following.

There are two phases: the question phase and the answer phase.

In the question phase, the judge tells to AliceBot an integer N . Then AliceBot has to answer some questions the judge asks about the permutation.

In the subsequent answer phase, AliceBot must compose two different permutations a_1, \dots, a_N and b_1, \dots, b_N that are consistent with the answers from the previous phase.

The judge who asks questions has an initial patience $P = 2N$. Each time the judge asks a question, the judge's patience decreases.

There are three types of questions the judge can ask:

- Type 1, formatted as “? 1 i j k ”: the judge chooses three different integers i, j, k ($1 \leq i, j, k \leq N$), AliceBot looks at the three integers a_i, a_j , and a_k , and tells Bob the **value** of their median (the one that is neither minimum nor maximum). Each such question decreases the judge's patience by 2.
- Type 2, formatted as “? 2 i j ”: the judge chooses two different integers i, j ($1 \leq i, j \leq N$), and AliceBot answers i if $a_i < a_j$, or j otherwise. Each such question decreases the judge's patience by 2.
- Type 3, formatted as “? 3 i j ”: the judge chooses two different integers i, j ($1 \leq i, j \leq N$), and AliceBot tells the minimum **value** among a_i and a_j . Each such question decreases the judge's patience by 1.

You may assume that judge's patience will never go below 2 after asking a question. When the judge decides that they asked enough questions, the command “!” is sent to the AliceBot, switching it to the answer phase.

In the answer phase, AliceBot tells the judge two permutations a_1, \dots, a_N and b_1, \dots, b_N . These two permutations must be consistent with all the answers given in the question phase, and the number of positions i such that $a_i \neq b_i$ has to be at least $\lceil p/2 \rceil$, where p is the judge's patience at the end of the question phase.

Because the judge is too lazy, you are asked to implement the AliceBot. It can be shown that the problem is solvable for every possible N from the constraints.

Interaction Protocol

First, the jury program tells you one integer N on a separate line ($4 \leq N \leq 50\,000$).

Then the jury asks questions.

A question of type 1 is a line formatted as “? 1 i j k ” ($1 \leq i, j, k \leq N$; i, j , and k are pairwise distinct). You then print a line with a single integer: the median of the values a_i, a_j , and a_k .

A question of type 2 is a line formatted as “? 2 i j ” ($1 \leq i, j \leq N; i \neq j$). You then print a line with a single integer: i if $a_i < a_j$, or j if $a_i > a_j$.

A question of type 3 is a line formatted as “? 3 i j ” ($1 \leq i, j \leq N; i \neq j$). You then print a line with a single integer: the minimum value among a_i and a_j .

Let there will be a total of q_1 questions of type 1, q_2 questions of type 2, and q_3 questions of type 3. You may assume that the value $p = 2N - 2q_1 - 2q_2 - q_3$ is not less than 2.

To switch to the answering phase, the jury program issues a line consisting of the ‘!’ sign. After that, you must print two lines, the first line containing N space-separated integers a_1, \dots, a_N , and the second line containing N space-separated integers b_1, \dots, b_N . Each of these two sequences must be a permutation of $1, \dots, N$, and they must differ in at least $\lceil p/2 \rceil$ positions.

Don’t forget to print the newline character and flush the output buffer after printing the answers and after printing each permutation, otherwise, you may get the “Idleness Limit Exceeded” error.

Example

standard input	standard output
5	
? 1 1 2 3	
	4
? 2 2 4	
	4
? 3 4 5	
	1
!	
	3 5 4 1 2
	5 4 3 2 1