Problem I
Inversion

This is an interactive problem.

While preparing the problem *H. Hardest problem* - a problem about counting, permutation and inversion - for the 2022 ICPC Regional contest, the jury has found out that some slow solutions can be optimized greatly by swapping two arbitrary elements of one permutation to get a new one. Loc, one member of the jury staff, hypothesizes that, if the solution somehow can maintain the number of inversions without storing the actual permutation, that solution will then run blazingly fast, and can even beat the model solution!

But how can some information about the inversions can effectively be used to obtain the original permutation? To demonstrate this point, let say there is a hidden permutation p of length n. Loc said that there is a way to find p by asking n questions of the following form:

- Choose two indices i and j ($1 \le i, j \le n$). What is the number of inversions of p if p_i and p_j are swapped?

Right now, only Loc knows how to use these information to find the array p, so Loc challenges everyone to do the same!

The hidden permutation p of length n is now owned by Loc. Find p by asking Loc at most n questions of the above form.

Interaction protocol

- First, your program should read an integer n ($1 \le n \le 10^4$) – the length of the hidden permutation p.
- Next, you can ask Loc - the jury - up to n questions. Your program should print `?` i j ($1 \le i, j \le n$) to ask the question, and then read the answer – the number of inversions of p if p_i and p_j are swapped.
- Whenever you obtained the permutation p, your program should print `!` $p_1\ p_2 \ldots p_n$ and exit gracefully.

Sample communication

Sample communication when the hidden perumtation p is $[1, 3, 2]$.

standard input	standard output	Explanation
3		The length of p is $n = 3$.
	? 1 2	If p_1 and p_2 are swapped, $p = [3, 1, 2]$.
2		The number of inversions of $[3, 1, 2]$ is 2.
	? 2 3	If p_2 and p_3 are swapped, $p = [1, 2, 3]$.
0		$[1, 2, 3]$ has no inversions.
	? 3 1	If p_3 and p_1 are swapped, $p = [2, 3, 1]$.
2		The number of inversions of $[2, 3, 1]$ is 2.
	! 1 3 2	The program answers that $p = [1, 3, 2]$ and terminates.

Note

After printing a query do not forget to output end of line and flush the output. To do this, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `stdout.flush()` in Python;
- see documentation for other languages.