Problem F. Inversion

Input file: `standard input`
Output file: `standard output`

This is an interactive problem.

There is a hidden permutation $p_1, p_2, \ldots, p_n$ of $\{1, 2, \ldots, n\}$. You want to find it by asking the parity of the number of inversions of $p_l, \ldots, p_r$.

You can query in the format "? l r", and the interactor will respond you $\left(\sum_{l \le i < j \le r} [p_i > p_j]\right) \bmod 2$. $[p_i > p_j]$ is 1 when $p_i > p_j$ and 0 when $p_i \le p_j$.

Interaction Protocol

Firstly, you should read the integer n ($1 \le n \le 2000$).

After that, you can make no more than 4×10^4 queries. To make a query, output "? l r" ($1 \le l \le r \le n$) on a separate line, then you should read the response from standard input.

To give your answer, print "! p_1 p_2 ... p_n" on a separate line. The output of the answer is **not** counted towards the limit of 4×10^4 queries.

After that, your program should terminate.

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, `flush(output)` in Pascal, or `stdout.flush()` in Python.

It is guaranteed that the permutation is fixed in advance.

Example

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