Hidden Order

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

This is an interactive problem.

VegiCat knows an array $a_1, a_2, \ldots, a_n$, where each element is 0, 1, or 2. He does not want to tell you its values, but you can ask him questions. In each query, you choose two indices i and j, and VegiCat tells you whether $a_i > a_j$.

VegiCat asks you to output the indices in nondecreasing order of the corresponding array values. That is, find a permutation $p_1, p_2, \ldots, p_n$ of $1, 2, \ldots, n$ such that

$$a_{p_1} \leq a_{p_2} \leq \cdots \leq a_{p_n}.$$

You may make at most $2n - 3$ queries in each test case.

The interactor is adaptive. This means that the array is not fixed in advance. The interactor guarantees that there always exists at least one array with elements in $\{0, 1, 2\}$ consistent with all responses given so far. Your final permutation must be valid for **every array consistent with all responses**.

Input

The first line contains an integer T ($1 \leq T \leq 250$), the number of test cases. The description of the test cases follows.

Each test case begins with an integer n ($2 \leq n \leq 500$), the length of the array.

The sum of n over all test cases does not exceed 500.

Interaction Protocol

To make a query, output a line in the following format:

```
? i j
```

Here, $1 \leq i, j \leq n$. After each query, read one integer:

- 1, if $a_i > a_j$;
- 0, if $a_i \leq a_j$.

Queries with $i = j$ and repeated queries are allowed and are included in the query count.

When you have determined the order, output a line in the following format:

```
! p_1 p_2 ... p_n
```

Each index from 1 to n must appear exactly once. Printing the answer does not count as a query.

If your answer is correct, proceed to the next test case by reading its n, or terminate your program if this was the last test case. The interactor does not send a separate confirmation after an answer.

After outputting each query or answer, output a newline and flush the output buffer. To flush your output, you can use:

- `fflush(stdout)` or `cout.flush()` in C/C++;
- `System.out.flush()` in Java and Kotlin;
- `sys.stdout.flush()` in Python.

If you make an invalid query, exceed the query limit, or print an incorrect answer, you will receive a **Wrong Answer** verdict. Output that does not follow the specified format is also judged **Wrong Answer**. In these cases, the interactor terminates without sending an error response.

Example

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

Note

In the first test case, the queries `? 1 2` and `? 2 3` both return 1, so $a_1 > a_2 > a_3$. Therefore, $(3, 2, 1)$ is a correct answer.

In the second test case, the query `? 1 2` returns 0, so $a_1 \leq a_2$. Therefore, $(1, 2)$ is a correct answer.