Yearning for Yonder 2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

This is an interactive problem.

> Thanks for your hard work, darling. Little Cyan Fish doesn't know how much of his code words can be conveyed, but still thanks you for walking to this point with Little Cyan Fish.
> So many things, so many absurd things happened, Little Cyan Fish is tired, very tired... In the past year, Little Cyan Fish felt several times that he was going to save himself, but after all, he was not strong enough to cut the strings on his body. Little Cyan Fish always imagines that in that faraway place, there is another self who can clearly see the map of his life.
>
> — "Longing for the Distance" · The 10th China Collegiate Programming Contest Final

Well, maybe, you can try, how difficult this will be. Little Cyan Fish feels that his life is a permutation of unknown shape. People always say the world is full of randomness, but why is Little Cyan Fish always so unlucky...? Aha! Maybe randomness is just this profound! So, Little Cyan Fish decides that this permutation of his will be randomly generated in the following way:

- Generate a random permutation $p_1, p_2, \ldots, p_n$: chosen uniformly at random from all $n!$ possible permutations of 1 to n.
- If $p_1 > p_n$, then reverse the entire permutation.

You cannot directly observe the structure of the permutation, but Little Cyan Fish has granted you a superpower: asking! Each time, you can interactively query the value of the distance between two numbers on the permutation modulo 3. Specifically, each time you can choose two numbers u, v ($1 \le u, v \le n$, $u \ne v$), the permutation will tell you the value of the distance between these two numbers modulo 3 (i.e., finding two indices i, j satisfying $p_i = u, p_j = v$, the permutation will answer $|i - j| \bmod 3$).

Now, Little Cyan Fish wants you to try and see what you can get. You need to help Little Cyan Fish determine this permutation in no more than $25n$ queries.

Input

There are multiple test cases. The first line of the input contains a single integer T ($1 \le T \le 1\,000$), indicating the number of test cases.

For each test case, first you need to read an integer n ($3 \le n \le 10^4$).

Interaction Protocol

Next, the interaction begins. You can make no more than $25n$ queries in each test case. To make a query, you need to output a single line "`?` u v" ($1 \le u, v \le n$, $u \ne v$), describing a query. Then, you need to read the result from standard input.

To provide your answer, you need to output "`!` $p_1\ p_2 \cdots p_n$". You output must satisfy $p_1 < p_n$. Outputting the answer will not be counted towards the $25n$ queries limit. After you output the answer, you need to read the next test case immediately, or exit your program immediately.

After outputting a query, **do not** forget to output a newline character and flush the output stream. To do this, you can use `fflush(stdout)` or `cout.flush()` in C++, `System.out.flush()` in Java, `flush(output)` in Pascal, and `stdout.flush()` in Python.

It is guaranteed that the sum of n over all test cases does not exceed 10^4.

In this problem, it is guaranteed that the interactor is **non-adaptive**. That is, the permutations are determined **randomly as the problem stated** before the interaction process. They will not change according to your queries. There are a total of 30 test cases (including the sample test case).

Example

standard input	standard output
2 3 2 1 1 4 1 1 1 0 2	 ? 1 2 ? 2 3 ? 1 3 ! 1 3 2 ? 2 3 ? 3 1 ? 1 4 ? 2 4 ? 2 1 ! 2 3 1 4

Note

Figure 1: Little Cyan Fish & <?> at <?> taken on March 4, 2026