

Yearning for Yonder

Input file: **standard input**
Output file: **standard output**
Time limit: 5 seconds
Memory limit: 1024 megabytes

This is an interactive problem.

The background story in the Chinese statements is removed due to the translation difficulties. >_<

Little Cyan Fish has a weighted tree of n vertices generated in the following way:

- First, generate a random labeled tree: uniformly randomly select from all n^{n-2} possible labeled trees.
- Then, independently assign random integer edge weights in the range $[1, K]$ to each edge, where K is a hidden parameter.

You cannot directly observe the structure of the tree or the edge weights, but Little Cyan Fish grants you a superpower: querying! Each time, you can query the distance between two vertices. Specifically, you can choose two vertices u, v ($1 \leq u, v \leq n$, $u \neq v$), and we will tell you the distance between these two vertices (i.e., the sum of the edge weights on the simple path connecting these two vertices).

Now, Little Cyan Fish wants you to determine all the edges and their weights within no more than $7n$ queries.

Interaction Protocol

Each test case contains multiple sets of test data. First, you need to read an integer T ($1 \leq T \leq 10^4$) indicating the number of data sets.

For each set of test data, you first need to read an integer n ($1 \leq n \leq 10^5$).

Next, the interaction process begins. You can make no more than $7n$ queries in each set of test data. To make a query, you need to output a line “? $u v$ ” ($1 \leq u, v \leq n$, $u \neq v$), describing a query. Then, you need to read the result from standard input.

To provide your answer, you need to output “! $u_1 v_1 w_1 u_2 v_2 w_2 \cdots u_{n-1} v_{n-1} w_{n-1}$ ”. You can output these edges in any order. The output of the answer will not count towards the $7n$ query limit. After you output the answer, you need to immediately read the next set of test data or terminate your program.

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

It is guaranteed that $1 \leq K \leq 10^4$, and the sum of all n in the test data does not exceed 10^5 .

In this problem, it is guaranteed that the interaction library is **non-adaptive**. That is, the shape of the tree and the edge weights are determined before the interaction process. They will not change with your queries.

Example

standard input	standard output
2	
3	? 1 2
3	? 2 3
4	? 1 3
7	! 1 2 3 2 3 4
4	? 1 2
3	? 2 3
7	? 2 4
2	? 1 3
4	? 1 4
5	? 3 4
9	! 1 2 3 1 3 4 2 4 2