Master of Both VII

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

This is an interactive problem.

Prof. Chen is proficient in computational geometry and interactive problems. After teaching a lesson on *polygon triangulation*†, Prof. Chen prepared the following homework assignment.

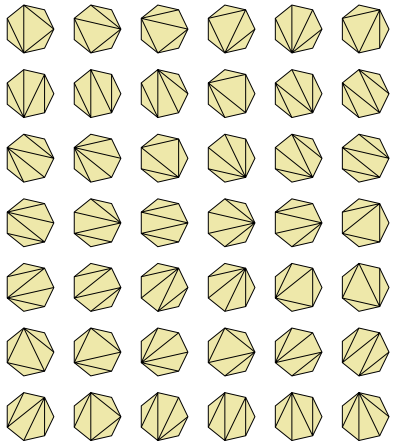
There is a regular n-gon with vertices labeled $1, 2, \ldots, n$ in counter-clockwise order. There is a hidden triangulation consisting of $n-3$ edges connecting the vertices. You can make the following query:

- "`?` u v": Query the information about the edge connecting u and v. If the edge u-v exists in the triangulation, it will return 0. Otherwise, it will return the number of edges in the triangulation that intersect with (u, v) at **non-endpoint** points.

Your task is to determine this triangulation using no more than $n - 3$ queries.

In this problem, the interactor is **not adaptive**, which means that the triangulation is predetermined and will not change during your interaction with the interactor.

†: In computational geometry, *polygon triangulation* is the partition of a simple polygon P into a set of triangles, i.e., finding a set of triangles with pairwise non-intersecting interiors whose union is P.



Pic. 1: An illustration of all 42 **VII**-gon triangulations (adapted from *Wikipedia*, CC BY-SA 3.0).

Input

The input contains multiple test cases. The first line contains an integer T ($1 \le T \le 2500$), denoting the number of test cases.

For each test case, the first line contains an integer n ($4 \le n \le 100$), denoting the polygon is a regular n-gon.

It is guaranteed that the sum of n does not exceed 10^4.

Interaction Protocol

For each test case, you can make no more than $n - 3$ queries. To make a query, output "`?` u v" ($1 \le u \ne v \le n$, u and v are non–adjacent on the convex polygon) on a separate line, then read the response from standard input. In response to the query, the interactor will return an integer x representing the result of the query. If $x = -1$, it means your query is invalid or exceeds the allowed number of queries, and you should terminate your program, resulting in a Wrong Answer verdict.

When you are ready to submit your answer, output "`!` a_1 b_1 a_2 b_2 $\ldots$ a_{n-3} b_{n-3}" on a separate line, representing the edges in the triangulation are (a_i, b_i) for $1 \le i \le n - 3$. The answer submission does not count toward the $n - 3$ query limit. In response to your answer, the interactor will return an integer r. If $r = 1$, it means your answer is correct; if $r = 0$, it means your answer is incorrect, and you should terminate your program, resulting in a Wrong Answer verdict.

After submitting your answer, your program should proceed to the next test case. Once all test cases are processed, your program should terminate.

After printing a query or submitting the answer, do not forget to output the end of the line and flush the output. To do this, use `fflush(stdout)` or `cout.flush()` in C++, `System.out.flush()` in Java, or `stdout.flush()` in Python.

Example

standard input	standard output
2 4 0 1 6 2 0 1	 ? 1 3 ! 1 3 ? 1 3 ? 4 6 ! 2 4 4 6 6 2