67

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

*The MITIT Winter 2025-26 contest takes place on December **6-7**, 2025.*

This is an interactive problem.

Busy Beaver has a secret array $a_1, \ldots, a_N$, where $1 \le a_i \le 10^9$ for all i and **every two elements are coprime**. (Two integers are *coprime* if the only positive integer dividing both is 1.)

You may ask up to 100 queries of the following form:

- Pick two distinct indices i and j. In response, you will receive the product $a_i \times a_j$.

Let Q be the maximum number of queries you use to determine Busy Beaver's array. For full points, you must have $Q \le \mathbf{67}$.

Input

Each test contains multiple test cases. The first line contains an integer T ($1 \le T \le 1000$) — the number of test cases.

The first line of each test case contains an integer N ($5 \le N \le 100$). After reading this line, you should begin the interaction.

Interaction Protocol

For each test case, begin by reading N.

To make a query, output "`?` i j" ($1 \le i, j \le n$; $i \ne j$) without quotes. Afterwards, you should read in a single integer — the product $a_i \times a_j$. You can make at most 100 such queries in a single test case.

If you receive the integer -1 instead of an answer, it means your program has made an invalid query, has exceeded the limit of 100 queries, or has given an incorrect answer on some previous test case. Your program must terminate immediately to receive a `Wrong Answer` verdict. Otherwise, you can get an arbitrary verdict because your solution will continue to read from a closed stream.

When you are ready to give the final answer, output "`!` a_1 $\ldots$ a_N" ($1 \le a_i \le 10^9$) without quotes — Busy Beaver's array. Giving this answer does not count towards the limit of 100 queries. Afterwards, your program must continue to solve the remaining test cases, or exit if all test cases have been solved.

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;
- `stdout.flush()` in Python;
- see your language's documentation for other languages.

Scoring

- For full points, you must have $Q \le \mathbf{67}$.
- For partial points, using $67 < Q \le 100$ queries will award $\lfloor 1.067^{125-Q} \rfloor$ points.

Example

standard input	standard output
2 5 77 30 85 5 69	 ? 1 2 ? 3 4 ? 4 5 ! 7 11 6 5 17 ? 1 5 ! 1 40 61 41 69

Note

During an actual run the solution does not know the hidden array; it is shown here only to justify the sample.

In the first test case, the judge prints `5`, so $N = 5$. Its hidden array is $[7, 11, 6, 5, 17]$.

- The program asks `? 1 2` and receives `77` from 7×11.
- It asks `? 3 4` and receives `30` from 6×5.
- It asks `? 4 5` and receives `85` from 5×17.

The program then correctly outputs `! 7 11 6 5 17`.

In the second test case, the judge prints `5`. Its hidden array is $[1, 40, 61, 41, 69]$.

The program asks `? 1 5` and receives `69` from 1×69. It then outputs `! 1 40 61 41 69`, which matches the judge's array.

This illustrates one valid interaction sequence; any correct determination of the array using allowed queries is acceptable.