Exchanging Kubic

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

This is an interactive problem.

In computer science, the maximum sum subarray problem, also known as the maximum segment sum problem, is the task of finding a contiguous subarray with the largest sum, within a given one-dimensional array $A_1, A_2, \cdots, A_n$ of numbers. Formally, the task is to find indices i and j, such that the following sum is as large as possible:

$$\sum_{i \le k \le j} A_k$$

It is also possible to choose an empty subarray, which means you found an empty array with the sum 0. The value of the maximum sum subarray is denoted by $MSS(A)$. For example, $MSS([-2,1,4,-3,5]) = 7$, $MSS([-5]) = 0$, and $MSS([-1,-2]) = 0$.

Little Cyan Fish is taking a course on Kubic Theory at Powerful Kubic University (PKU). Today, Prof. Kubic asked Little Cyan Fish to play the following game with him during the course:

Prof. Kubic has a sequence of integers $a_1, a_2, \cdots, a_n$. Little Cyan Fish could ask Prof. Kubic at most $2n$ questions in the following form:

- `? l r`: query the value of $MSS(a_l, a_{l+1}, \cdots, a_r)$.

The task of Little Cyan Fish is to report a sequence $b_1, b_2, \cdots, b_n$ satisfying:

- $MSS(a_l, a_{l+1}, \cdots, a_r) = MSS(b_l, b_{l+1}, \cdots, b_r)$ for all $1 \le l \le r \le n$.

Little Cyan Fish found this task is very challenging. Can you help him to develop a strategy to finish Prof. Kubic's task?

Interaction Protocol

There are multiple test cases in a single test file. The first line of the input contains a single integer T ($1 \le T \le 10^4$), indicating the number of the test cases.

For each test case, the first line of the input contains a single integer n ($1 \le n \le 2\,000$).

Then, the interaction begins. You may perform at most $2n$ queries in each test case. To perform a query, you need to print a single line "`?` l r" ($1 \le l \le r \le n$), indicating a query. Then, you need to read the result of your query from the standard input.

To give your answer, you need to print "`!` b_1 b_2 $\cdots$ b_n". You need to ensure that $-10^{15} \le b_i \le 10^{15}$. Printing the answer is not considered a query and does not count toward the $2n$ limit. After printing you answer, you need to read the next case, or terminate your program immediately.

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, "`flush(output)`" in Pascal, or "`stdout.flush()`" in Python.

It is guaranteed that $-10^9 \le a_i \le 10^9$, and the sum of n over all test cases does not eceed 10^4.

In this problem, it is guaranteed that the interactor is **non-adaptive**. That is, the values of a_i are decided before the interaction process. They will not be changed based on your queries.

Example

standard input	standard output
2	
3	
	? 1 1
1	
	? 2 2
0	
	? 3 3
1	
	! 1 -1 1
5	
	? 1 3
4	
	? 3 5
5	
	! 2 -1 3 -4 5