Aegisfortia

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

This is an interactive problem.

Yurisaki is trying to recover a hidden sequence $a_1, a_2, \ldots, a_n$, where each element is either 0 or 1. In one query, she chooses integers l, r $(1 \le l \le r \le n)$ and receives

$$\left(\sum_{i=l}^{r} a_i\right) \mid (r - l + 1),$$

where $\mid$ denotes bitwise OR: a bit of $x \mid y$ is 1 if at least one of the corresponding bits of x and y is 1, and 0 otherwise.

Help *Yurisaki* recover the sequence using at most n queries. If another binary sequence of length n gives the same response for all $\binom{n+1}{2}$ possible queries (l, r) with $1 \le l \le r \le n$, including queries you have not asked, you must report that the sequence cannot be uniquely determined instead.

Input

This problem contains multiple test cases.

The first line contains a positive integer T $(1 \le T \le 10^4)$, representing the number of test cases. At the beginning of each test case, read an integer n $(1 \le n \le 10^4)$, the length of the hidden sequence.

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

Output

For each test case, output `!` followed by the n elements of a on a single line if the sequence can be uniquely determined, or `! -1` otherwise. Separate adjacent tokens by a space. The final answer does not count as a query.

Interaction Protocol

To make a query, output `? l r` on a single line, then read one integer: the response defined above. You may make at most n queries per test case.

The hidden sequence is fixed before the interaction begins and does not change. After outputting the final answer, proceed to the next test case, or terminate if all test cases are complete. If you receive -1 or reach the end of input, terminate immediately.

Flush your output after every query and final answer. To flush your output, you can use:

- `fflush(stdout)` or `cout.flush()` in C/C++;
- `System.out.flush()` in Java;
- `sys.stdout.flush()` in Python.

Example

3 3 4 2 3 4 5 3 3 6 7 5	 ! -1 ? 1 2 ? 1 3 ? 1 4 ! -1 ? 1 2 ? 2 4 ? 2 5 ? 1 5 ? 1 4 ! 0 1 0 0 1

Note

In the sample interaction, the hidden sequences are $[1, 1, 0]$, $[1, 1, 1, 1]$, and $[0, 1, 0, 0, 1]$, respectively. Blank lines are included only to align the interaction.