Problem L. Lost Permutation

Time limit: 2 seconds
Memory limit: 512 megabytes

This is an interactive problem.

You once had a permutation π of size n. And now it's gone. All you have left is an old device you made while studying group theory. To try and recover π you can input a permutation f of size n into this device. This device will then display a permutation $\pi^{-1} \circ f \circ \pi$. Find π using at most two interactions with the device.

A permutation of size n is a sequence of n distinct integers from 1 to n. The *composition* of two permutations a and b is a permutation $a \circ b$ such that $(a \circ b)_i = b_{a_i}$. That is, if we consider a permutation as an action on n elements, moving element at position i to a_i, then $a \circ b$ is the action that applies a, then applies b, so that element at position i first moves to a_i, then moves to b_{a_i}. Note that some definitions of composition use the reverse order.

The inverse permutation π^{-1} is a permutation σ such that $\sigma_{\pi_i} = i$. The composition of a permutation and its inverse is equal to an identity permutation: $(\pi \circ \pi^{-1})_i = (\pi^{-1} \circ \pi)_i = i$ for all i from 1 to n. For example, if $a = (4, 1, 3, 2)$ and $b = (3, 2, 1, 4)$, then $a \circ b = (4, 3, 1, 2)$, $a^{-1} = (2, 4, 3, 1)$ and $a^{-1} \circ b \circ a = (1, 2, 4, 3)$.

Interaction Protocol

Your program has to process multiple test cases in a single run. First, the testing system writes t, the number of test cases ($t \geq 1$). Then, t test cases should be processed one by one.

In each test case your program should start by reading the integer n ($3 \leq n \leq 10^4$), the size of permutation π. The sum of n over all test cases does not exceed 10^4. Then, your program can make queries of two types:

- `?` $f_1\, f_2\, \ldots\, f_n$, values $f_1, f_2, \ldots, f_n$ form a permutation of $1, 2, \ldots, n$. The testing system responds with a permutation $g_1, g_2, \ldots, g_n$, where $g = \pi^{-1} \circ f \circ \pi$.
- `!` $\pi_1\, \pi_2\, \ldots\, \pi_n$ — your guess for the secret permutation.

You can use at most two queries of the first type in each test case. After your program makes a query of the second type, it should continue to the next test case (or exit if that test case was the last one).

Example

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

Note

There are two test cases in the first test. In the first test case, $\pi = (4, 1, 3, 2)$ is the only permutation that satisfies $\pi^{-1} \circ (3, 2, 1, 4) \circ \pi = (1, 2, 4, 3)$ and $\pi^{-1} \circ (2, 4, 3, 1) \circ \pi = (2, 4, 3, 1)$. In the second test case, based on the interaction, π can be equal to either $(1, 3, 2)$, $(2, 1, 3)$, or $(3, 2, 1)$. The solution got lucky and guessed the correct one: $(3, 2, 1)$.