Programming: RAM Stick Permutations

Problem Story

In his free time outside of work at Lemon AI, Jānis trades RAM sticks to make extra income.

One day, Jānis found an incredible deal on a set of high-performance RAM sticks and decided to upgrade the company's server. Each stick has a unique serial number (1 through n).

However, there's a catch: the server has a peculiar BIOS configuration system. For optimal performance, the RAM sticks must be installed in a specific order determined by the motherboard's memory controller, but the manual is nowhere to be found!

Fortunately, the BIOS has a diagnostic mode: Jānis can test an arrangement of RAM sticks, and the system will give him a score indicating how many slots have the correct stick.

Being a competitive programmer, Jānis recognizes this as a **Combinatorial Optimization (CO)** problem —finding the right arrangement in discrete space.

> ***Combinatorial Optimization (CO)*** *— a field focused on finding optimal solutions in discrete spaces (e.g., permutations, combinations, graphs). CO problems are extremely hard as the number of possible combinations grows exponentially with the number of elements. They're often solved using neural solvers and Reinforcement Learning (RL), though clever algorithms can solve specific cases efficiently.*

A permutation is a particular ordering of given elements. For instance, the possible permutations of numbers 1, 2, 3 are:

```
- 1 - 2 - 3
- 1 - 3 - 2
- 2 - 1 - 3
- 2 - 3 - 1
- 3 - 1 - 2
- 3 - 2 - 1
```

Your task is to help Jānis figure out the correct RAM configuration efficiently!

Problem

This is an interactive problem.

The server's BIOS has a secret optimal configuration C of n RAM sticks ($1 \leq n \leq 31$).

Each stick is denoted with a unique serial number ($1 \leq \text{id} \leq n$).

Jānis's goal is to discover the correct configuration by using the BIOS diagnostic mode.

Each query, Jānis tests an arrangement P, and receives a score S.

The score S is calculated as follows:

- For each position i ($1 \leq i \leq n$), we add 2^i to S if the stick at position i in P matches the stick at position i in C.

For example, if:

- The optimal BIOS configuration $C = [2, 1, 3, 4]$
- Jānis's test arrangement $P = [4, 1, 3, 2]$

Then, score $S = 2^2 + 2^3 = 12$, because the sticks at positions 2 and 3 match (both have stick 1 at position 2, and both have stick 3 at position 3).

Your task is to help Jānis discover the configuration C in at most n queries.

Input

The first line contains an integer n ($1 \leq n \leq 31$).

Interaction

You can test an arrangement using the BIOS diagnostic by querying `? p_1 p_2 ... p_n` trough given `query` function, where:

- `p_1, p_2, ..., p_n`: a proposed arrangement P of n sticks

The BIOS will respond with an integer score S.

Output

Once you have determined the optimal configuration, query `! c_1 c_2 ... c_n` where $c_1, c_2, \ldots, c_n$ is the configuration C.

Limits

The limits on input are:

- test case count $\leq 2 \times 10^4$
- sum of n over all test cases $\leq 1 \times 10^6$

The execution time limit is 20 seconds. The memory limit is 512 MB.

Example

Take $n = 3$, $C = [2, 1, 3]$.
We will ask two queries to determine C, and then output the result.

Input:

```
3
```

Interaction:

```
? 1 2 3
8
? 2 3 1
2
! 2 1 3
```

Grading

Your solution will be checked on number of hidden tests.

Score = passed tests / total tests