I Isaac's Queries



You have reached the final level of the popular roguelike game "Isaac's Keybindings". Instead of a boss, you encounter a shopkeeper who holds a hidden array of integers $a_1, a_2, \ldots, a_n$, where $0 \le a_i < 2^{30}$ for each i in $[1, n]$.

It is guaranteed that **the array is generated randomly**, i.e., a_i is a uniformly random integer in $[0, 2^{30})$ for each i in $[1, n]$, in all the tests excluding the example.

Let $f(u, v) = a_u \oplus a_{u+1} \oplus \ldots \oplus a_v$, where $\oplus$ is the bitwise `XOR`.

You can ask queries of the following form: `? u v`, with $1 \le u \le v \le n$.

The answer to the query is:

- -1, if $f(u, v) = 0$;
- $\lfloor \log_2(f(u, v)) \rfloor$ otherwise.

Each query has a cost of $\frac{1}{v-u+1}$ robocoins. You initially have 35 robocoins and if your balance ever becomes negative you lose. Note that your robocoin balance does not need to be an integer at any moment.

Find the answer to all possible $\frac{n(n+1)}{2}$ queries without losing.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 30$). The description of the test cases follows.

The first line of each test case contains one integer n ($1 \le n \le 100$) — the length of the array $a_1, a_2, \ldots, a_n$. It is guaranteed that the array is generated randomly in all the tests excluding the example.

There are exactly 50 tests in this problem (including the example). The example has $t = 1$ and $n = 3$, and all the other tests have $t = 30$ and $n = 100$.

Interaction

For each test case, first read a single integer n. If the integer you read is -2, it means that the answer to the previous test case was wrong, and you should exit immediately.

To ask a query, print a line in the format `? u v` with $1 \le u \le v \le n$.

As a response to the query, you will receive -2 if you made an invalid query (i.e., the format is invalid, or you would reach a negative amount of robocoins). In that case, you should exit immediately. Otherwise, you will receive the answer to the query.

When you determine the answers to all the $\frac{n(n+1)}{2}$ queries, print them in the following format.

Print $n+1$ lines. The first line must contain a single character `!`. The i-th of the next n lines must contain $n - i + 1$ integers: the answers to queries `? i i`, `? i i+1`, ..., `? i n`, respectively.

After printing a query, do not forget to output the 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 the documentation for other languages.

Samples

Sample input 1	Sample output 1
1 3 2 -1 1	 ? 1 2 ? 1 3 ? 2 3 ! 1 2 -1 2 1 2

Explanation of sample 1.
In the example, the hidden array is $[a_1, a_2, a_3] = [2, 4, 6]$.

- First, you ask `? 1 2`. Since $f(1,2) = a_1 \oplus a_2 = 6$, the answer is $\lfloor \log_2(6) \rfloor = 2$.
- Then, you ask `? 1 3`. Since $f(1,3) = a_1 \oplus a_2 \oplus a_3 = 0$, the answer is -1.
- Then, you ask `? 2 3`. Since $f(2,3) = a_2 \oplus a_3 = 2$, the answer is $\lfloor \log_2(2) \rfloor = 1$.

Now, even without knowing the hidden array, you claim the answer to all the possible 6 queries. For example, you claim that the answer to `? 1 1` is 1.

You have spent $\frac{1}{2} + \frac{1}{3} + \frac{1}{2} = \frac{4}{3}$ robocoins, which is less than the allowed 35 robocoins.

Image credits: Photo by Michael Ocampo, CC BY 2.0, modified to remove background, via Wikimedia Commons

BLANK PAGE