

Doremy's Perfect DS Class (Hard Version)

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

The only difference between this problem and the other two versions is the maximum number of queries. In this version, you are allowed to ask at most 20 queries. You can make hacks only if all versions of the problem are solved.

This is an interactive problem.

“Everybody! Doremy's Perfect Data Structure Class is about to start! Come and do your best if you want to have as much IQ as me!” In today's Data Structure class, Doremy is teaching everyone a powerful data structure — Doremy tree! Now she gives you a quiz to prove that you are paying attention in class.

Given an array a of length m , Doremy tree supports the query $Q(l, r, k)$, where $1 \leq l \leq r \leq m$ and $1 \leq k \leq m$, which returns the number of distinct integers in the array $[\lfloor \frac{a_l}{k} \rfloor, \lfloor \frac{a_{l+1}}{k} \rfloor, \dots, \lfloor \frac{a_r}{k} \rfloor]$.

Doremy has a secret permutation p of integers from 1 to n . You can make queries, in one query, you give 3 integers l, r, k ($1 \leq l \leq r \leq n$, $1 \leq k \leq n$) and receive the value of $Q(l, r, k)$ for the array p . Can you find the index y ($1 \leq y \leq n$) such that $p_y = 1$ in at most 20 queries?

Note that the permutation p is fixed before any queries are made.

Interaction Protocol

You begin the interaction by reading an integer n ($3 \leq n \leq 1024$) in the first line — the length of the permutation.

To make a query, you should output

- “? l r k ” ($1 \leq l \leq r \leq n$, $1 \leq k \leq n$)

in a separate line. After each query, you should read an integer x — the value of $Q(l, r, k)$ for p . In this version of the problem, you can make at most 20 such queries.

To give the answer, you should output

- “! y ” ($1 \leq y \leq n$)

in a separate line, where $p_y = 1$.

After printing a query or the answer, do not forget to output the end of line and flush the output. Otherwise, you will get Idleness limit exceeded. To do this, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- see documentation for other languages.

1 Hacks Format

The first line of the hack contains an integer n ($3 \leq n \leq 1024$) — the length of the permutation.

The second line of the hack contains n distinct integers p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$) — the permutation.

Example

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

Note

The permutation in the example is $[3, 5, 2, 1, 4]$.

The input and output for example illustrate possible interaction on that test (empty lines are inserted only for clarity).

In this interaction process:

- For the first query, $\lfloor \frac{3}{4} \rfloor = 0$, $\lfloor \frac{5}{4} \rfloor = 1$, $\lfloor \frac{2}{4} \rfloor = 0$, so the answer is 2.
- For the second query, $\lfloor \frac{2}{3} \rfloor = 0$, $\lfloor \frac{1}{3} \rfloor = 0$, $\lfloor \frac{4}{3} \rfloor = 1$, so the answer is still 2.
- For the third query, $\lfloor \frac{2}{5} \rfloor = 0$, $\lfloor \frac{1}{5} \rfloor = 0$, so the answer is 1.
- For the fourth query, $\lfloor \frac{2}{2} \rfloor = 1$, $\lfloor \frac{1}{2} \rfloor = 0$, $\lfloor \frac{4}{2} \rfloor = 2$, so the answer is 3.

The correct answer is got after 4 queries, so this process will be judged correct.