VJudge 洛谷 未找到

M↓

D. Find the Last Number

time limit per test: 3 seconds
memory limit per test: 256 megabytes

This is an interactive problem.

There is a hidden permutation* p of length n. You are allowed to interact with it by asking the following query at most $2n$ times:

- Select two integers i and x such that $1 \le i \le \boldsymbol{n-1}$ and $1 \le x \le 10^9$. The grader will respond 0 if p_i & x† is equal to zero, and 1 otherwise.

Important: You cannot make queries involving the last element p_n (because $i \le n-1$).

Your goal is to determine the value of the last element of the permutation, p_n, using at most $2n$ queries.

Note that the interactor is non-adaptive. This means that the hidden permutation p is fixed at the beginning and will not change based on your queries.

* A permutation of length n is an array consisting of n distinct integers from 1 to n in arbitrary order. For example, $[2,3,1,5,4]$ is a permutation, but $[1,2,2]$ is not a permutation (2 appears twice in the array), and $[1,3,4]$ is also not a permutation ($n=3$ but there is 4 in the array).

† & denotes the bitwise AND operation.

DeepL 翻译

这是一个互动问题。

有一个长度为 n 的隐藏排列 * p 。你可以通过最多 $2n$ 次提出下面的问题来与它交互：

- 选择两个整数 i 和 x ，使得 $1 \le i \le \boldsymbol{n-1}$ 和 $1 \le x \le 10^9$ 。如果 p_i & x † 等于零，评分员将回答 0 ，否则回答 1 。

重要：不能查询最后一个元素 p_n （因为 $i \le n-1$ ）。

您的目标是最多使用 $2n$ 次查询，确定排列的最后一个元素 p_n 的值。

请注意，交互器是非适应性的。这意味着隐藏排列 p 在开始时是固定的，不会根据您的查询而改变。

* 长度为 n 的排列是一个数组，由 n 个不同的整数组成，这些整数从 1 到 n 按任意顺序排列。例如， $[2,3,1,5,4]$ 是一个排列，但 $[1,2,2]$ 不是一个排列（ 2 在数组中出现了两次）， $[1,3,4]$ 也不是一个排列（ $n=3$ ，但数组中有 4 ）。

† & 表示 **位与运算**。

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^3$). The description of the test cases follows.

The first line of each test case contains a single integer n ($2 \le n \le 2 \cdot 10^4$) — the length of permutation p.

Codeforces Round 1061 (Div. 2)

比赛进行中

01:45:27

Contestant

→ 提交?

语言: GNU G++17 7.3.0

选择文件: 选择文件 未选择文件

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

提交

→ 评分表

	Score
Problem A	472
Problem B	708
Problem C	1416
Problem D	1888
Problem E	2596
Problem F1	3068
Problem F2	3068
Successful hack	100
Unsuccessful hack	-50
Unsuccessful submission	-50
Resubmission	-50

* If you solve problem on 00:14 from the first attempt

For each test case, after reading n, you should begin the interaction and find the answer before proceeding to the next test case.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^4$.

DeepL 翻译

输入

每个测试包含多个测试用例。第一行包含测试用例的数量 t ($1 \le t \le 10^3$)。测试用例说明如下。

每个测试用例的第一行都包含一个整数 n ($2 \le n \le 2 \cdot 10^4$) - 排列长度 p 。

对于每个测试用例，在读取 n 后，应开始交互并找到答案，然后再继续下一个测试用例。

保证所有测试用例中 n 的总和不超过 $2 \cdot 10^4$ 。

Interaction

To make a query, output a single line in the following format:

- ? i x $(1 \le i \le n-1, 1 \le x \le 10^9)$ — the index to query and the value to take bitwise AND with.

After each query, you should read a single integer b:

- $b =$ 0 if $p_i \ \& \ x = 0$,
- $b =$ 1 otherwise.

You may make at most $2n$ such queries in each test case.

To give the final answer, output a single line in the following format:

- ! x $(1 \le x \le n)$ — the value of p_n.

Note that giving the answer does not count as one of the $2n$ queries.

After printing each query do not forget to output the end of line and flush* the output. Otherwise, you will get `Idleness limit exceeded` verdict.

If, at any interaction step, you read -1 instead of valid data, your solution must exit immediately. This means that your solution will receive `Wrong answer` because of an invalid query or any other mistake. Failing to exit can result in an arbitrary verdict because your solution will continue to read from a closed stream.

Hacks

To hack, use the following format.

The first line should contain a single integer t $(1 \le t \le 10^3)$ — the number of test cases.

The first line of each test case should contain a single integer n $(2 \le n \le 2 \cdot 10^4)$ — the length of permutation p.

The second line of each test case should contain n integers $p_1, p_2, \ldots, p_n$ $(1 \le p_i \le n)$ — the elements of permutation p.

The sum of n over all test cases should not exceed $2 \cdot 10^4$.

* To flush, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `sys.stdout.flush()` in Python;
- see the documentation for other languages.

DeepL 翻译

互动

要进行查询，请按以下格式输出一行：

- ? i x ($1 \le i \le n-1$ ， $1 \le x \le 10^9$) - 要查询的索引和要与之进行位和运算的值。

每次查询后，都应读取一个整数 b ：

- $b=0$ 如果是 $p_i \,\&\, x = 0$ 、
- 否则为 $b=1$ 。

每个测试用例中最多只能进行 $2n$ 次这样的查询。

要给出最终答案，请按以下格式输出一行：

- $!\ x$ （ $1 \le x \le n$ ）- p_n 的值。

请注意，给出答案并不算 $2n$ 查询之一。

打印完每个查询后，不要忘记输出行尾并 flush * 输出。否则，您将收到 "超过闲置限制 "的提示。

如果在任何交互步骤中读取的不是有效数据，而是 -1 ，则必须立即退出解决方案。这意味着，由于查询无效或其他错误，您的解决方案将收到 "错误答案"。未退出可能会导致任意判定，因为您的解决方案将继续从封闭流中读取数据。

黑客

要黑客攻击，请使用以下格式。

第一行应包含一个整数 t （ $1 \le t \le 10^3$ ）- 测试用例数。

每个测试用例的第一行应包含一个整数 n （ $2 \le n \le 2 \cdot 10^4$ ）--置换长度 p 。

每个测试用例的第二行应包含 n 个整数 $p_1, p_2, \ldots, p_n$ （ $1 \le p_i \le n$ ）- 排列 {8829962} 的元素。

所有测试用例中 n 的总和不应超过 $2 \cdot 10^4$ 。

* 若要刷新，请使用：

- fflush(stdout) 或 C++ 中的 cout.flush();
- 在 Python 中使用 sys.stdout.flush();
- 其他语言请参见文档。

Example

input Copy

```
2
2

0

1

3

1

0

1
```

output Copy

```


? 1 1

? 1 2

! 1

? 1 3

? 1 2

? 2 3

! 2
```

Note

In the first test, the interaction proceeds as follows.

Solution	Jury	Explanation
	2	There are 2 test cases.
	2	In the first test case, the hidden permutation is $[2, 1]$ (of length 2).
? 1 1	0	The solution queries $p_1 \ \& \ 1$. Since $p_1 = 2$ and $2 \ \& \ 1 = 0$, the jury responds with 0.
? 1 2	1	The solution queries $p_1 \ \& \ 2$. Since $p_1 = 2$ and $2 \ \& \ 2 = 2$, the jury responds with 1.
! 1		The solution determines that the last element is 1 since it knows that the first element is not 1 from the first query.
	3	In the second test case, the hidden permutation is $[1, 3, 2]$ (of length 3).
? 1 3	1	The solution queries $p_1 \ \& \ 3$. Since $p_1 = 1$ and $1 \ \& \ 3 = 1$, the jury responds with 1.
? 1 2	0	The solution queries $p_1 \ \& \ 2$. Since $p_1 = 1$ and $1 \ \& \ 2 = 0$, the jury responds with 0.
? 2 3	1	The solution queries $p_2 \ \& \ 3$. Since $p_2 = 3$ and $3 \ \& \ 3 = 3$, the jury responds with 1.
! 2		The solution determines that the last element is 2.

Note that the empty lines in the example input and output are only for readability. They do not appear in the actual interaction.

DeepL 翻译

注

在第一次测试中，互动过程如下。

解决方案	**评审委员会**	**解释**
	2	有 2 个测试案例。
	2	在第一个测试用例中，隐藏排列为 $[2,1]$ （长度为 2 ）。
? 1 1	0	解答查询 $p_1 \& 1$ 。由于 $p_1 = 2$ 和 $2 \& 1 = 0$ ，评委的回应是 0 。
? 1 2	1	解答质疑 $p_1 \& 2$ 。由于 $p_1 = 2$ 和 $2 \& 2 = 2$ ，评审团的回应是 1 。
! 1	由于知道第一个元素不是第一次查询的 1 ，因此解法确定最后一个元素是 1 。	
	3	在第二个测试用例中，隐藏排列为 $[1,3,2]$ （长度为 3 ）。
? 1 3	1	解答查询 $p_1 \& 3$ 。由于 $p_1 = 1$ 和 $1 \& 3 = 1$ ，陪审员的回应是 1 。
? 1 2	0	解答质疑 $p_1 \& 2$ 。因为 $p_1 = 1$ 和 $1 \& 2 = 0$ ，所以评审团的回应是 0 。
? 2 3	1	解答质疑 $p_2 \& 3$ 。因为 $p_2 = 3$ 和 $3 \& 3 = 3$ ，所以评审团的回应是 1 。
! 2	解答确定最后一个元素是 2 。	

请注意，输入和输出示例中的空行只是为了便于阅读。它们不会出现在实际交互中。

GNU G++17 7.3.0

▶ 自定义测试数据(自动保存)