G. Wonderful Guessing Game

time limit per test: 2 seconds
memory limit per test: 256 megabytes

This is an interactive problem.

You are a proud teacher at the Millennium Science School. Today, a student named Alice challenges you to a guessing game.

Alice is thinking of an integer from 1 to n, and you must guess it by asking her some queries.

To make things harder, she says you must **ask all the queries first**, and she will **ignore** exactly 1 query.

For each query, you choose an array of k **distinct** integers from 1 to n, where k is even. Then, Alice will respond with one of the following:

- $\texttt{L}$: the number is one of the first $\frac{k}{2}$ elements of the array;
- $\texttt{R}$: the number is one of the last $\frac{k}{2}$ elements of the array;
- $\texttt{N}$: the number is not in the array;
- $\texttt{?}$: this query is ignored.

Alice is impatient, so you must find a strategy that **minimizes** the number of queries. Can you do it?

Formally, let $f(n)$ be the minimum number of queries required to determine Alice's number. Then you must find a strategy that uses **exactly** $f(n)$ queries.

Note that the interactor is **adaptive**, which means Alice's number is not fixed at the beginning and may depend on your queries. However, it is guaranteed that there exists at least one number that is consistent with Alice's responses.

We can show that $f(n) \leq 20$ for all n such that $2 \leq n \leq 2 \cdot 10^5$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^4$). The description of the test cases follows.

The only line of each test case contains a single integer n ($2 \leq n \leq 2 \cdot 10^5$) — the maximum possible value of Alice's number.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Interaction

The interaction begins by reading the integer n.

Then, output a single integer q ($1 \leq q \leq 20$) — the number of queries.

To ask a query, output a line in the following format:

- $k\ a_1\ a_2 \ldots a_k$ ($2 \leq k \leq n$, k is even, $1 \leq a_i \leq n$, the a_i are distinct) — the length of the array, and the array itself.

Once you've asked all q queries, read a string s ($|s| = q$) — the responses to the queries as described above.

When you know Alice's number, output a single integer x ($1 \leq x \leq n$) — the value of the number.

Then, move on to the next test case, or terminate the program if there are no more test cases.

After outputting all q queries, do not forget to output the end of the line and flush the output. Otherwise, you will get `Idleness limit exceeded`. To do this, use:

- `fflush(stdout)` or `cout.flush()` in C++;

Neowise Labs Contest 1 (Codeforces Round 1018, Div. 1 + Div. 2)

Finished

Practice

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Clone Contest to Mashup

You can clone this contest to a mashup.

Clone Contest

→ Submit?

Language: GNU G++23 14.2 (64 bit, ms

Choose file: Choose File No file chosen

Submit

→ Last submissions

Submission	Time	Verdict
316692462	Apr/22/2025 16:53	Accepted

→ Problem tags

bitmasks constructive algorithms interactive *3200

No tag edit access

→ Contest materials

- Announcement (en)
- Tutorial (en)

- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- See the documentation for other languages.

Note that even if you correctly determine Alice's number but use more than $f(n)$ queries, you will get `Wrong answer`.

For this problem, hacks are disabled.

Example

input

```
2
3



?N

5




R?L
```

output

```


2
2 1 2
2 1 2

3

3
4 3 2 4 1
4 5 4 3 1
4 1 5 3 4

1
```

Note

In the first test case, $n = 3$. We ask 2 queries: $[1, 2]$, and $[1, 2]$ again.

- For the first query, Alice's response is $\texttt{?}$, which means this query is ignored.
- For the second query, Alice's response is $\texttt{N}$, which means her number is not in the array $[1, 2]$.

From the information above, we can determine that Alice's number is 3.

It can be shown that all valid strategies for $n = 3$ require at least 2 queries.

In the second test case, $n = 5$. We ask 3 queries: $[3, 2, 4, 1]$, $[5, 4, 3, 1]$, and $[1, 5, 3, 4]$.

- For the first query, Alice's response is $\texttt{R}$, which means her number is in the array $[4, 1]$.
- For the second query, Alice's response is $\texttt{?}$, which means this query is ignored.
- For the third query, Alice's response is $\texttt{L}$, which means her number is in the array $[1, 5]$.

From the information above, we can determine that Alice's number is 1.

It can be shown that all valid strategies for $n = 5$ require at least 3 queries.

Codeforces (c) Copyright 2010-2025 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Apr/22/2025 15:53:26$^{\text{UTC+2}}$ (l1).
Desktop version, switch to mobile version.
Privacy Policy | Terms and Conditions

Supported by