

Service Desk

time limit per test: 1 second (1 second for Python)
memory limit per test: 976.6 megabytes
input: standard input
output: standard output
points: 100 · problem code: servicedesk

Customers arrive at a service desk at given seconds. Each customer is either normal or urgent.

The desk processes time in whole seconds starting from second 1. At each second, all customers whose arrival time is at most that second are considered waiting. If at least one customer is waiting, exactly one customer is served.

- On an **odd** second, serve the waiting customer who arrived earliest.
- On an **even** second, serve an urgent customer first (if any urgent customer is waiting); otherwise serve the waiting normal customer who arrived earliest.

If no customer is waiting at a second, nobody is served at that second.

Determine the order in which customers are served.

Input

The first line contains an integer n , the number of customers.

The next n lines each contain:

- an integer t_i (arrival second), and
- a character c_i (**U** for urgent, **N** for normal).

Customer i corresponds to line i (1-indexed) in this list.

Output

Print n lines. On line k , print the index of the customer served as the k -th served customer.

Constraints

- $1 \leq n \leq 2 \cdot 10^5$
- $1 \leq t_i \leq 10^9$
- all t_i are pairwise distinct
- $c_i \in \{\text{U}, \text{N}\}$

Example 1

Input

```
5
1 U
2 N
3 N
4 U
5 N
```

Output

```
1
2
3
4
5
```

Explanation

- Each customer arrives at a different second, and the first four are served immediately.

- At second 4 (even), customer 4 is urgent and is served.
- Then customer 5 is served.

Example 2

Input

```
4
2 N
4 U
5 N
7 U
```

Output

```
1
2
3
4
```