Parallelogramisation

People are starting to form a line in front of Tyx2019 so that they can turn into parallelograms! Initially, the line is empty. There will be Q queries, and you must handle 3 types of queries:

- Type 1: A person with area X joins the back of the line.
- Type 2: The person currently at the back of the line reconsiders the decisions that has led to this point, and leaves the line.
- Type 3: The person at the front of the line is turned into a parallelogram by Tyx2019, and is translated out of the line. (Note: The person leaves the line)

For each Type 3 query, please output the area of the person who has been turned into a parallelogram!
Note that queries of type 2 and 3 will **NOT** be given if the line is empty.

Input Format

Your program must read from standard input.

The input consists of $Q + 1$ lines.
The first line consists of one integer, Q.
The next Q lines each describe one query, consisting of the type of query, and then X for that query if it is a Type 1 query.

Output Format

Your program must print to standard output.

For each Type 3 query, output the area of the person who has been turned into a parallelogram, separated by a new line.

Subtasks

For all test cases, the input will satisfy the following bounds:

- $2 \leq Q \leq 4 \cdot 10^6$
- For all Type 1 queries, $1 \leq X \leq 10^{18}$

Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Marks	Additional Constraints
1	69	$N \leq 10^5$ and $X \leq 10^9$, and there are no Type 2 queries.
2	30	$N \leq 10^5$ and $X \leq 10^9$
3	1	No additional constraints

Sample Testcase 1

This testcase is valid for subtask 1, 2 and 3.
Input:

```
5
1 55
1 69
3
1 1
3
```

Output:

```
55
69
```

Explanation:
After the first query, the line is {55}.
After the second query, the line is {55, 69}.
After the third query, the person at the front of the line is parallelogramised, and hence you should print 55, then the line is {69}.
After the fourth query, the line is {69, 1}.
After the fifth query, the person at the front of the line is parallelogramised, and hence you should print 69, then the line is {1}.

Sample Testcase 2

This testcase is valid for subtask 2 and 3.
Input:

```
6
1 314
1 15
2
3
1 69
3
```

Output:

```
314
69
```