and a special operation "MEX":

- `1 x` : Push. Push an element x onto the stack.
- `2` : Pop. Remove the element on top of the stack. (If the stack is empty, do nothing.)
- `3` : Top. Output the element on top of the stack. (If the stack is empty, output -1 .)
- `4` : MEX. Output the smallest non-negative integer that does not exists in the stack.

Input

The first line contains an integer q, being the number of operations.

The following q lines are the operations described in the problem statements.

Restrictions

- $1 \leq q \leq 8 \cdot 10^5$
- $1 \leq x \leq 2 \cdot 10^5$

Output

For each operation with type 3 and type 4, print the answer in one line.

Sample Input 1

```
24
2
3
4
1 2
1 4
4
1 1
4
1 3
4
1 5
4
1 3
4
3
2
4
2
4
2
4
2
4
3
```

Sample Output 1

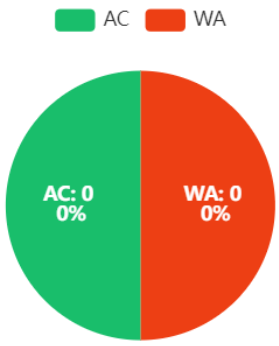
```
-1
1
1
3
5
6
6
3
6
5
3
1
4
```

Submissions
Rankings
View Contest

Information

ID	3
Time Limit	1000MS
Memory Limit	256MB
IO Mode	Standard IO
Created By	ta_david
Level	Hidden
Score	100
Tags	Show

Statistic Details



- The definition of MEX number is the smallest non-negative integer that **does not** exists in the stack.
- Maintain **more than one** data structures may be a good way to prevent time limit exceeded.

Explanation of Sample IO

Below is the explanation of first 10 operations in sample IO.

- 1st operation `2` : Pop, since the stack is empty, do nothing.
- 2nd operation `3` : Top, since the stack is empty, output `-1` .
- 3rd operation `4` : MEX, since the stack is empty, the smallest non-negative integer that does not exists in the stack is 1, output `1` .
- 4th operation `1 2` : Push, push 2 onto the stack.
- 5th operation `1 4` : Push, push 4 onto the stack.
- 6th operation `4` : MEX, since the stack has 2 and 4 now, the smallest non-negative integer that does not exists in the stack is 1, output `1` .
- 7th operation `1 1` : Push, push 1 onto the stack.
- 8th operation `4` : MEX, since the stack has 2, 4 and 1 now, the smallest non-negative integer that does not exists in the stack is 3, output `3` .
- 9th operation `1 3` : Push, push 3 onto the stack.
- 10th operation `4` : MEX, since the stack has 2, 4, 1 and 3 now, the smallest non-negative integer that does not exists in the stack is 5, output `5` .

Detailed Constraints

For test ID 1 - 4 (40% of total points):

- $q \leq 2500$

For test ID 5 - 10 (60% of total points)

- No additional constraints

Language: C Theme: Solarized Light

```
1
```

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