Back To School '16: Paradox

Time limit: 1.0s **Memory limit:** 256M

You are in English class and learning about paradoxen. Unbeknownst to the teacher, the `set<bool>` data structure is incredibly useful to determine whether or not a situation/statement is a paradox. Implement a `set<bool>` interface for your teacher!

You are given C commands, each in the following 4 forms:

1. `1 E` insert element `E` into the set. Print `true` or `false` depending on whether or not the element was successfully inserted (did it **not** exist in the set before?).
2. `2 E` erase element `E` from the set. Print `true` or `false` depending on whether or not the element was successfully erased (did it exist in the set before?).
3. `3 E` find element `E` in the set. Print the index of the element within the set (0-indexed). If the element does not exist, print `-1`.
4. `4` print the elements in increasing order (`false` < `true`).

`E` will be either `true` or `false`.

Input Specification

$1 \leq C \leq 100$

Output Specification

For each command, print a single line of output.

Sample Input

```
5
1 true
2 false
3 false
1 false
4
```

Sample Output

```
true
false
-1
true
false true
```