E. Nikita and stack

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Nikita has a stack. A stack in this problem is a data structure that supports two operations. Operation `push(x)` puts an integer x on the top of the stack, and operation `pop()` deletes the top integer from the stack, i. e. the last added. If the stack is empty, then the operation `pop()` does nothing.

Nikita made m operations with the stack but forgot them. Now Nikita wants to remember them. He remembers them one by one, on the i-th step he remembers an operation he made p_i-th. In other words, he remembers the operations in order of some permutation $p_1, p_2, \ldots, p_m$. After each step Nikita wants to know what is the integer on the top of the stack after performing the operations he have already remembered, in the corresponding order. Help him!

Input

The first line contains the integer m ($1 \le m \le 10^5$) — the number of operations Nikita made.

The next m lines contain the operations Nikita remembers. The i-th line starts with two integers p_i and t_i ($1 \le p_i \le m$, $t_i = 0$ or $t_i = 1$) — the index of operation he remembers on the step i, and the type of the operation. t_i equals 0, if the operation is `pop()`, and 1, is the operation is `push(x)`. If the operation is `push(x)`, the line also contains the integer x_i ($1 \le x_i \le 10^6$) — the integer added to the stack.

It is guaranteed that each integer from 1 to m is present exactly once among integers p_i.

Output

Print m integers. The integer i should equal the number on the top of the stack after performing all the operations Nikita remembered on the steps from 1 to i. If the stack is empty after performing all these operations, print `-1`.

Examples

input
2 2 1 2 1 0
output
2 2

input
3 1 1 2 2 1 3 3 0
output
2 3 2

input
5 5 0 4 0 3 1 1

```
2 1 1
1 1 2
```

output

```
-1
-1
-1
-1
2
```

Note

In the first example, after Nikita remembers the operation on the first step, the operation `push(2)` is the only operation, so the answer is 2. After he remembers the operation `pop()` which was done before `push(2)`, answer stays the same.

In the second example, the operations are `push(2)`, `push(3)` and `pop()`. Nikita remembers them in the order they were performed.

In the third example Nikita remembers the operations in the reversed order.