Problem A. Get the Password

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	256 megabytes

***Implement Linked list for this question. If found using any other data structure your submission will not be considered.**

Jake is frustrated with the DSA labs. So he decides to hack into the laptop of DSA TA and get the solutions for the next lab. For that he need the password. But he has to solve a puzzle to get the password.

He is given a Linked list of length N where each Node contains two fields:

- value(Integer)
- Reference to Next Node in linked list

He is also given a list of M operations which he must perform on the list.

There are two operation you can perform on the linked list:

'I' <pos>(Integer) <val>(Integer): This inserts a new Node with $value = val$ at the index(0-indexed) *pos* in the Linked List.

e.g If the list contains 1 3 4, I 2 5 will insert value 5 at 2nd index(0-indexed) into the linked list and the linked list becomes 1 3 5 4. then I 0 5 will insert at 0th index so new list will be 5 1 3 5 4.

'D' <val>(Integer) <order>(Integer): This will delete the node with $value = val$ from the Linked list. *order* parameter can only be 0 or 1. If *order* is 0, it means the first occurrence of node which satisfies the condition gets deleted . If order is 1 last node with $value = val$ will get deleted.If no node with $value = val$ is found do not do anything.

e.g if the linked list is 1 3 4 3 2 4 6 . D 3 0 will make the linked list 1 4 3 2 4 6, whereas now if D 4 1 is performed, this will make linked list 1 4 3 2 6.

It is guaranteed that pos will be within range of length of linked list at that moment.

Jake is unable to solve the Puzzle. Help him solve the puzzle and get the password.

Input

First line contains two integers N and M, representing the current size of the list and total number of operations to be performed respectively.

Second Line contains N space separated integers representing node values as they are in the list. Next M line contains instructions that need to be performed over the list.

$1 \leq N \leq 1000$

$1 \leq M \leq 1000$

$0 \leq value \leq 10000$

$0 \leq pos \leq N$ (At the time of Insertion)

Output

Return space separated node values representing status of linked list after performing M operations onto the linked list. In next line print the sum of all values in the list.

If there are no nodes remaining in the linked list return -1 as output.

Examples

standard input	standard output
5 4 1 2 3 4 5 I 2 6 I 4 5 D 5 1 I 1 3	1 3 2 6 3 5 4 24
1 1 2 D 2 1	-1
5 4 1 2 3 4 5 I 2 6 I 4 5 D 6 1 I 0 3	3 1 2 3 5 4 5 23
5 5 1 2 3 4 5 I 5 6 I 4 5 D 6 1 I 0 3 D 3 0	1 2 3 4 5 5 20
6 6 2 2 1 3 1 0 I 0 4 I 1 0 D 2 0 I 0 2 I 7 1 D 2 0	4 0 2 1 3 1 1 0 12

Problem B. Special Friend

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Shivansh has a special friend among his juniors. He will leak the problems of upcoming lab to his special friend if his speical friend is able to correctly able to guess the output of the game.

They are playing a game with sequences. Intitally the sequence is empty.

He can ask his friend to add an Integer at the end of the sequence **(Query** 1**)**, delete an element at any position **(Query** 2**)**, insert an element at any particular position **(Query** 3**)** and modify the Sequence as follows **(Query** 4**)**.

In **Query** 4 his friend has to reorder the sequence such that she takes the last value of the sequence, and then the first value of the sequence, and then the second last value of the sequence, and then the second value of the sequence, etc.

For his special friend the instructions are too much to keep track of. So she has asked you to write a program for it. (See sample test case for reference)

Input

First line contains a single integer q, the number of queries. ($1 \leq q \leq 1000$)

Following q lines of queries follows: ($1 \leq a, b \leq 1000$)
The queries are as follows:

- 1 a : Add integer a to the end of the sequence.
- 2 a : Delete the element present at position a. If position is invalid, do nothing.
- 3 a b : Add integer b at position a
- 4 : Modify the sequence as described

The sequence is 1-based indexed.

In query 3 all positions are guaranteed to be ($1 \leq$ positions $\leq$ the size of sequence at that point).

Output

After end of queries print the modified sequence or print -1 if the sequence is empty.

Examples

standard input	standard output
8 1 3 2 1 1 4 1 2 1 0 1 1 4 3 2 5	1 5 4 0 2
5 1 1 2 1 1 2 2 1 3 1 1	1
4 1 1 2 1 1 2 2 1	-1
5 1 1 2 1 2 1 2 1 1 5	5
11 3 1 5 1 10 1 15 1 25 2 4 1 20 4 1 99 1 100 1 101 4	101 20 100 5 99 15 10
5 1 1 1 2 1 3 1 4 4	4 1 3 2