Problem A. Stickers Fetish

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

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

Sanyam and Shivansh likes to talk using stickers only. They use only 2 types of stickers L and R. Pr0hum one of their friend have found the string of characters L and R denoting the stickers they both sent to each other and he doubt this string might contain some hints about the today's lab. Hence he will like to decode that string for his juniors.

The way he decodes the string $S = S_1, S_2 \ldots S_n$ where $S_i = L$ or R $(1 \leq \text{i} \leq n)$ is, he considers an empty set P and then ,

1) For $i = 1$ he simple adds 1 to the set P irrespective of the S_1.

2) For $i = 2, 3 \ldots N$ in this order only,

- if $S_i = L$, append i to left of $i - 1$ in the set.
- if $S_i = R$, append i to right of $i - 1$ in the set.

Since pr0hum is a bit busy learning linked lists, he has told you this strategy and would like you to decode this string for him.

Input

First line contain an integer n $(1 \leq \text{n} \leq 10^6)$ denoting the length of string S.

Second line contains the string S of length n consisting of L and R.

Output

Print the final decoded set P .

Examples

standard input	standard output
5 LLRRL	2 3 5 4 1
8 RLLRLRRL	3 5 6 8 7 4 2 1
5 LLLLL	5 4 3 2 1
5 RRRRR	1 2 3 4 5
8 LLRRLLRR	2 3 6 7 8 5 4 1
10 LRRLRLRLRL	1 2 4 6 8 10 9 7 5 3
7 LRLLLRL	1 5 7 6 4 3 2
10 LRRRRRRRLL	1 2 3 4 5 6 7 10 9 8

Note

For testcase 1,

Consider an empty set P = [] and S = LLRRL

For $i = 1$, insert 1 in the set. [1]

For $i = 2$, insert 2 to the left of 1. [$2, 1$]

For $i = 3$, insert 3 to the right of 2. [$2, 3, 1$]

For $i = 4$, insert 4 to the right of 3. [$2, 3, 4, 1$]

For $i = 5$, insert 5 to the left of 4. [$2, 3, 5, 4, 1$]

Problem B. Bring out your Likes

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

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

You have been given the task to design a fancy number sequence. A fancy number sequence is a queue which contains numbers which a user likes the most.

Users can also remove or add some numbers from the sequence. You have to implement a Data Structure to design this fancy number sequence.

There are 4 operations user can perform on this queue :

'I' <pos>(Integer) <val>(Integer): This inserts a new number with $ID = val$ at the index(0-indexed) *pos* in the sequence. E.g If the list contains 1 3 4, I 2 5 will insert value 5 at 2nd index, hence linked list becomes 1 3 5 4.

'D' <val>(Integer): This will delete all the numbers with $ID = val$ from the sequence. If there is no element with $ID = val$ do nothing.

'M': This will print the middle element of the sequence. If there are two middle elements print the first one ie. If list contains 1,4,9,2 simply 4 is the middle element.

'P': This will print "YES" (without quotes) if the current sequence is a palindrome "NO" (without quotes) otherwise.

Input

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

Second-line contains N space-separated integers representing ID of the numbers in the order same as they are in the sequence.

The next M line contains operations that needs to be performed sequentially on the initial sequence.

Query 3 and 4 will not be asked to perform if sequence is empty.

$1 \leq N \leq 5000$

$1 \leq M \leq 1000$

$0 \leq val \leq 10^9$

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

Output

For queries of type 3 print the ID of the middle element in the sequence.

For queries of type 4 print "YES" or "NO".

After end of all the queries print the whole number sequence or print -1 if it is empty.

Examples

standard input	standard output
5 2 1 2 3 4 5 I 2 3 D 2	1 3 3 4 5
5 2 1 1 2 1 1 M P	2 YES 1 1 2 1 1
5 5 1 2 2 2 1 D 2 P M D 1 D 1	YES 1 -1
10 5 1 2 3 4 5 5 4 3 2 1 P D 5 P D 4 P	YES YES YES 1 2 3 3 2 1
10 8 1 2 3 4 5 5 4 3 2 1 M D 5 M D 4 M D 3 M D 5	5 4 3 2 1 2 2 1
1 5 1 I 0 2 I 0 1 M D 1 P	2 YES 2
2 4 1 1 D 1 D 2 D 4 I 0 4	4