NOI '05 P2 - Maintaining a Sequence

Time limit: 0.6s **Memory limit:** 64M

National Olympiad in Informatics, China, 2005

Please write a program that maintains a sequence, supporting the following 6 operations:

Operation	Input Format	Description
1. Insert	`INSERT posi tot` c_1 c_2 `...` c_{tot}	After the $posi$-th number in the current sequence, insert a total of tot numbers: $c_1, c_2, \ldots, c_{tot}$. Insertion to the beginning of the sequence will have $posi$ equal to 0.
2. Delete	`DELETE posi tot`	Starting at the $posi$-th number in the current sequence, delete a total of tot consecutive numbers.
3. Modify	`MAKE-SAME posi tot c`	Starting at the $posi$-th number in the current sequence, change all the values of tot consecutive numbers to c.
4. Reverse	`REVERSE posi tot`	Starting at the $posi$-th number in the current sequence, reverse the order of tot consecutive numbers.
5. Get Sum	`GET-SUM posi tot`	Starting at the $posi$-th number in the current sequence, output the sum of tot consecutive numbers. Note that $tot = 0$ is possible, in which case you should output 0.
6. Max Sum	`MAX-SUM`	Output the largest sum of any (non-empty) consecutive subsequence of the current sequence.

Input Specification

The first line of input contains two integers N and M, where N is the initial length of the sequence and M is the number of operations.

The second line of input contains N integers, describing the initial sequence.

For the next M lines, each line contains a command in one of the formats described above.

Output Specification

For each `GET-SUM` or `MAX-SUM` operation in the input, output the result of the query on a separate line.

Sample Input

```
9 8
2 -6 3 5 1 -5 -3 6 3
GET-SUM 5 4
MAX-SUM
INSERT 8 3 -5 7 2
DELETE 12 1
MAKE-SAME 3 3 2
REVERSE 3 6
GET-SUM 5 4
MAX-SUM
```

Sample Output

```
-1
10
1
10
```

Scoring

For each test case, your score will be determined as follows:

- If your program prints the correct answers to all `GET-SUM` operations at the correct locations in the output, then you will score 60% of points.
- If your program prints the correct answers to all `MAX-SUM` operations at the correct locations in the output, then you will score 40% of points.
- If your program correctly answers both types operations, then you will score 100% of points.

Please note: If your program can only correctly process one type of operation, then ensure that operations of the other type each receive a corresponding line. Otherwise, we cannot guarantee that your score will be accurate.

Constraints

You may assume that at any given time, the sequence will contain at least 1 number.
The data in the input is guaranteed to be valid, and will always refer to existing positions in the sequence.

In test data worth at most 50% of the points, the sequence may contain up to $30\,000$ numbers at any given moment.
In test data worth 100% of the points, the sequence may contain up to $500\,000$ numbers at any given moment.

In test data worth 100% of the points, the value of any number in the sequence will be in the range $[-1\,000, 1\,000]$.
In test data worth 100% of the points, $M \leq 20\,000$, the sum of all inserted values will not exceed $4\,000\,000$, and the input will not exceed 20MB.