



## Jump!

Time Limit	0.3 s
Memory Limit	64 MB

### Problem Description

Mr. Dengklek has a smart frog called Si Katak. Si Katak lives in a circular pond. Mr. Dengklek has an unlimited number of stones, numbered 1, 2, 3, ... . Initially, he surrounds the pond clockwise with stones 1, 2, ...,  $N$  in that order. Mr. Dengklek also assigns scores to the stones such that stone  $i$  is assigned  $A_i$  points. The other stones have 0 points.

Mr. Dengklek loves to request Si Katak to jump on the stones. For each request, Si Katak starts on stone 1 and picks a non-negative integer  $P$ . If  $P = 0$ , then it decides to ignore Mr. Dengklek's request. Otherwise the following happens. Si Katak jumps clockwise to the next stone repeatedly. As the pond is circular, stone  $N$ 's next stone is stone 1. After  $P$  jumps, Si Katak stops on a stone. Mr. Dengklek marks that stone and then Si Katak continues to jump until it jumps  $P$  times, and so on. Si Katak stops its actions right after stone 1 has been marked.

Deliberately, Si Katak will never pick  $P = 1$ . It will also never pick  $P \geq N - 1$  if  $N \geq 2$ . This is because the total distance of its jumps will be too near or too far. So,  $P$  will always be between 2 and  $N - 2$ , inclusive, or 0. If  $P = 0$ , then Mr. Dengklek will only mark stone 1.

Finally, Mr. Dengklek will accumulate the scores on the marked stones. Si Katak has to pick  $P$  in such a way that the total score is maximum.

To be more challenging, sometimes Mr. Dengklek changes the value of  $N$ , the number of stones surrounding the pond. Suppose that Mr. Dengklek changes the value of  $N$  to  $M$ . Then,

- If  $M < N$ , then Mr. Dengklek will remove stones  $M + 1, M + 2, \dots, N$ . The removed stones will retain their scores.
- If  $M > N$ , then Mr. Dengklek will add stones  $N + 1, N + 2, \dots, M$  clockwise in that order between stone  $N$  and stone 1.

After that, Mr. Dengklek sets  $N = M$ .

Not only that, sometimes Mr. Dengklek changes the score of a stone. Mr. Dengklek can only change the score of stones that are currently surrounding the pond (stones 1, 2, ...,  $N$ ).

Help Si Katak determine the value of  $P$  for each of Mr. Dengklek's jump requests, such that the total score of the resulting marked stones is maximum.

### Input Format

The first line contains 2 integers  $N$  and  $Q$ .  $N$  is the number of stones surrounding the pond initially, and  $Q$  is the number of operations. The next line contains  $N$  integers  $A_1 \dots A_N$ .

Each of the next  $Q$  lines contains one of these actions:

- `JUMP`, Si Katak must jump in such a way that the total score of marked stones is maximum.
- `RESIZE M`, Mr. Dengklek changes the value of  $N$  to  $M$ .



- UPDATE  $X$   $Y$ , Mr. Dengklek changes the score of stone  $X$  into  $Y$ .

## Output Format

For each JUMP request, output the maximum possible total score of Si Katak's jumps.

## Sample Input

```
8 8
3 -1 -100 0 8 1 5 2
JUMP
RESIZE 4
JUMP
RESIZE 8
JUMP
UPDATE 3 4
RESIZE 9
JUMP
```

## Sample Output

```
11
3
11
22
```

## Explanation

For the first request, Si Katak should pick  $P = 4$ . The marked stones are stone 5 and stone 1 in that order. The total score will be  $A_5 + A_1 = 11$ .

Next, the value of  $N$  is changed to 4. So, stones 5, 6, 7, 8 are removed. The only possible values of  $P$  are 0 and 2. The highest possible total score of Si Katak's jumps will be 3 (when  $P = 0$ ).

Next, the value of  $N$  is changed to 8. So, stones 5, 6, 7, 8 are added back. Hence, Si Katak should pick  $P = 4$  again and will score 11 points.

Finally, the score of stone 3 is changed from  $-100$  points to 4 points, and the value of  $N$  is changed to 9. The score of stone 9 is 0. Si Katak could choose  $P = 4$ . The marked stones will be stones 5, 9, 4, 8, 3, 7, 2, 6, 1 in that order. The total score is 22.

## Subtasks

For each subtask,

- $1 \leq i \leq N$
- $1 \leq X \leq N$
- $-1,000,000 \leq Y \leq 1,000,000$
- $-1,000,000 \leq A_i \leq 1,000,000$

### Subtask 1 (13 points):

- $1 \leq N \leq 100$  at any moment.
- $1 \leq Q \leq 100$



**Subtask 2 (27 points):**

- $1 \leq N \leq 2,000$  at any moment.
- $1 \leq Q \leq 1,000$

**Subtask 3 (24 points):**

- $1 \leq N \leq 2,000$  at any moment.
- $1 \leq Q \leq 50,000$
- There will be no `RESIZE` actions.

**Subtask 4 (36 points):**

- $1 \leq N \leq 2,000$  at any moment.
- $1 \leq Q \leq 50,000$