gave them a sequence v of N numbers, an initial number K and Q updates and queries of the following form:

- ***Update***: Change the value of an element
- ***Query***: You are given two numbers l and r. A *tambourinishment* is the operation of taking a subarray $v_l, v_{l+1}, ..., v_r$ and transforming each of its elements $v_{l \leq i \leq r}$ into $\sum_{k=l}^{i} v_k$ (in some more elevated circles, they call this applying partial sums only to the elements of the given subarray). You have to output the value of v_r after tambourinishing the subarray K times. As these values can get quite large, you have to output their remainder after being divided by $10^9 + 7$. The query is not persistent (after applying the tambourinishments, the sequence goes back to its state before the query).

Standard input

The first line will contain the numbers N, K and Q in this order. The second line will contain N numbers representing the values of sequence v The i-th of the next Q lines will describe updates and queries in the following format: the line will begin with the character `Q` or `U`. In case the character is `Q`, the line will describe a query and the character will be followed by two numbers l and r mentioned in the problem statement. In case the character is `U`, the line will describe an update and the character will be followed by two numbers p and x implying that the value of v_p will be changed with x.

Standard output

The output will contain the answers of the queries, each written in order

Constraints and notes

- $1 \leq N, Q \leq 10^5$
- $2 \leq K \leq 8$
- The elements of the sequence will never exceed 10^9
- For all queries, $1 \leq l \leq r \leq N$
- For all updates, $1 \leq p \leq N$
- Note: the author knows that the title of the problem is not fully adequate to the problem statement, but couldn't resist making that pun, also *tambourinishment* is a made-up word that the author had to use after being forced by his girlfriend.

Input	Output	Explanation
5 3 4 3 1 4 1 5 Q 1 5 Q 2 4 U 4 6 Q 3 5	87 19 47	We have 3 queries and 1 update. The last query is performed on the following subarray: $[4, 6, 5]$. After each of the 3 steps, the subarray looks as the following: $k = 1 : [4, 10, 15]$ $k = 2 : [4, 14, 29]$ $k = 3 : [4, 18, 47]$ The result is 47.

WORKSPACE / SUBMIT