Problem Maxstack

Input file	`stdin`
Output file	`stdout`

We define the following two operations which can be performed on a stack:

- push(x) — the number x is added to the top of the stack,
- pop — the number on top of the stack is removed from the stack.

A sequence of operations is considered to be *correct* if, when the operations are performed on an initially empty stack in order, the following two conditions are met:

1. No pop operation is performed on an empty stack,
2. After the last operation, the stack is empty.

For example, (push(1), pop) is correct, whereas (push(1), push(2)), or (pop, push(1)) are not.

We will consider a list $L_1, \ldots, L_N$ of operations, numbered from 1 to N. By $s(i, j)$, where $i \leq j$, we denote the sequence of operations $L_i, L_{i+1}, \ldots, L_j$.

We define the value maxstack(i, j) as follows. If $s(i, j)$ is not a correct sequence, then we define maxstack$(i, j) = 0$. Otherwise, we perform the operations $L_i, L_{i+1}, \ldots, L_j$ in order on an initially empty stack. After each operation, we calculate the maximum value in the stack. Let m_k be the maximum value after operation k, or zero if the stack is empty. Then, maxstack$(i, j) = m_i + m_{i+1} + \cdots + m_j$.

You are given N, the list L of operations, a number Q, and Q queries of the form (l, r), where $1 \leq l \leq r \leq N$. You are also given a number C. Depending on the value of C, you must calculate the following for all queries:

1. If $C = 1$, then you should calculate maxstack(l, r) modulo $10^9 + 7$. **It is guaranteed that $s(l, r)$ is correct for all queries.**
2. If $C = 2$, then you should calculate the sum of maxstack(i, j) for all $l \leq i \leq j \leq r$ modulo $10^9 + 7$. **It is guaranteed that for every query, if you perform the operations of $s(l, r)$ in order, then no pop operation is performed on an empty stack.**

Input data

The first line of input contains the value C. The second line contains the integers N and Q. The third line contains non-negative integers $X_1, X_2, \ldots, X_N$, which encode $L_1, \ldots, L_N$ as follows:

- If $X_i > 0$, then $L_i = \text{push}(X_i)$,
- If $X_i = 0$, then $L_i = \text{pop}$.

Each of the following Q lines contains two integers l and r, representing the queries.

Output data

Each of the Q lines of output should contain the answers to the queries, in order. All answers must be given modulo $10^9 + 7$.

Restrictions

- $1 \leq N, Q \leq 300\,000$.
- $0 \leq X_i \leq 10^9$, for all $1 \leq i \leq N$

- $L_1, \ldots, L_N$ is guaranteed to be a correct sequence of operations
- We call a sequence of operations *finally empty* if, when performing these operations on an empty stack, the stack is empty only before the first operation and after the last one. For example, (push(1), pop) is finally empty, but (push(1), pop, push(1), pop) is not.

#	Points	Restrictions
1	7	$C = 1, N, Q \leq 100$
2	14	$C = 1, N, Q \leq 1\,000$
3	15	$C = 1$, $X_i \leq 30$ and $s(l, r)$ is finally empty for every query (l, r)
4	13	$C = 1$
5	14	$C = 2$, $s(l, r)$ is correct and finally empty for every query (l, r)
6	11	$C = 2$, $s(l, r)$ is correct for every query (l, r)
7	10	$C = 2, N \leq 70\,000, Q \leq 50$
8	11	$C = 2, N, Q \leq 70\,000$
9	5	$C = 2$

Examples

Input file	Output file
1 6 2 5 4 0 0 23 0 1 4 5 6	15 23
1 10 4 22 0 26 0 72 447 0 497 0 0 1 10 3 10 8 9 1 4	1208 1186 497 48
2 10 5 22 0 26 0 72 447 0 497 0 0 1 10 3 10 8 9 1 4 1 9	5538 4260 497 96 1984

Explanations

First Example. For the first query, we will perform the operations from 1 to 4 in order. After the first operation, the stack looks like this: (5). $m_1 = 5$. After the second operation, the stack

looks like this: $(5, 4)$. $m_2 = 5$. After the third operation, the stack looks like this: (5). $m_3 = 5$. After the last operation, the stack is empty. $m_4 = 0$. $m_1 + m_2 + m_3 + m_4 = 5 + 5 + 5 + 0 = 15$.

For the second query, we must perform operations 5 and 6. After operation 5, the stack looks like this: (23). $m_5 = 23$. After the last operation, the stack is empty. $m_6 = 0$. $m_5 + m_6 = 23 + 0 = 23$.

Second Example. For the first query,

$$m_1 + \cdots + m_{10} = 22 + 0 + 26 + 0 + 72 + 447 + 72 + 497 + 72 + 0 = 1208.$$

For the second query,

$$m_3 + \cdots + m_{10} = 26 + 0 + 72 + 447 + 72 + 497 + 72 + 0 = 1186.$$

For the third query, $m_8 + m_9 = 497 + 0 = 497$.

For the last query, $m_1 + m_2 + m_3 + m_4 = 22 + 0 + 26 + 0 = 48$.

Third example. The values of maxstack for all subsequences (i, j) are written in the table below. We leave the cells where $i > j$ (for which the operation is not defined) empty.

j \ i	1	2	3	4	5	6	7	8	9	10
1	0									
2	22	0								
3	0	0	0							
4	48	0	26	0						
5	0	0	0	0	0					
6	0	0	0	0	0	0				
7	0	0	0	0	0	447	0			
8	0	0	0	0	0	0	0	0		
9	0	0	0	0	0	944	0	497	0	
10	1208	0	1186	0	1160	0	0	0	0	0