Problem Wool. Strong Connectivity Strikes Back

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	1 second
Memory limit:	512 megabytes

Vertices u and v of a directed graph are called strongly connected if there exists a path from u to v and a path from v to u in the graph. Note that if u and v are strongly connected, and v and w are strongly connected, then u and w are also strongly connected. Therefore, the vertices of the graph are divided into sets—strongly connected components. A vertex belonging to a strongly connected component is strongly connected to all vertices in the component (including itself) and is not strongly connected to any vertices outside the component.

During a graph class, Alice drew a directed graph with n vertices on the board and highlighted its strongly connected components. During the break, Bob decided to play a trick on Alice and erase the directions on some edges of the graph. He wants Alice to be able to uniquely restore the erased directions based on the remaining edges and the partition into strongly connected components after the break.

Help Bob by determining the maximum number of edges in the graph whose directions he can erase, as well as the number of ways he can do this.

More formally, find the maximum size of a subset of edges A that has the following property: if the directions of the edges in the set A are erased, then based on the information about the old strongly connected components and the directions of the edges not in the set A, the directions of the edges in the set A can be uniquely restored in such a way that the strongly connected components remain unchanged.

Since the number of such maximum subsets can be very large, output it modulo $10^9 + 7$.

Solutions that correctly determine the maximum size of the set A but incorrectly present the number of such subsets will receive partial points.

Input

The first line contains three integers n, m, and g ($2 \le n \le 2000$, $1 \le m \le 2000$, $0 \le g \le 7$)—the number of vertices and edges in the graph, respectively, as well as the test group number.

The next m lines contain two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$)—the numbers of the vertices that are the start and end of the i-th edge.

It is guaranteed that for any $1 \le i, j \le n$ $(u_i, v_i) \ne (u_j, v_j)$ and $(u_i, v_i) \ne (v_j, u_j)$, meaning that any two vertices are connected by at most one edge, regardless of direction.

Output

In the first line, output a single number—the size of the maximum subset of edges whose directions can be erased.

In the second line, output a single number—the number of such subsets modulo $10^9 + 7$. If you do not want to present the number of such subsets, then in the second line, instead, output any number from -1 to $10^9 + 6$. In this case, your solution will receive partial points for the test.

Note that omitting any number in the second line leads to a verdict of "Wrong Answer"and zero points for this test, regardless of the correctness of the size of the subset.

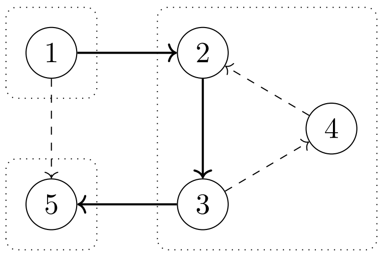
Example

input	output

5 6 0
1 2
1 5
2 3
3 4
3 5
4 2
	3
	3

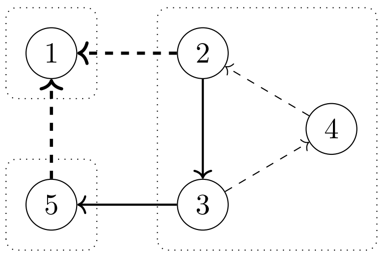
Note

The graph from the example with strongly connected components highlighted:



The directions on the dashed edges can be removed. Indeed, the edge $(1, 5)$ cannot be oriented in the opposite direction, because otherwise, vertices 1, 2, 3, and 5 would belong to the same strongly connected component. The edges $(3, 4)$ and $(4, 2)$ cannot be oriented differently either, because then vertices 2, 3, and 4 would not belong to the same strongly connected component.

Now, let's consider an incorrect way of selecting a subset of edges:



Here, the directions on the bold dashed edges cannot be removed. For example, if we reverse the orientation of edges $(1, 2)$ and $(1, 5)$, we get a graph with the same decomposition into strongly connected components.

It can be shown that the directions on exactly 4 edges cannot be removed, so the answer is 3.

Scoring

The tests for this problem consist of seven groups. The rules for scoring the subgroups are described below.

If the size of the maximum subset of edges whose directions can be erased and the number of such subsets are correctly calculated in all tests of the group, full points are awarded for the group.

Otherwise, if the size of the maximum subset of edges whose directions can be erased is correctly calculated in all tests of the group, but the number is incorrect in at least one test of the group, partial points are awarded for the group. Note that dependent groups will be tested in this case and may even be scored full points.

Group	Points		Additional Constraints		Required Groups	Comment
	Partial	Full	n	m		
0	0	0	–	–	–	Examples.
1	7	11	$n \le 14$	$m \le 14$	0	
2	5	9	$n \le 20$	$m \le 20$	0, 1	
3	8	12	–	–	–	$u_i < v_i$, for all $1 \le i \le n-1$ there is an edge $(i, i+1)$
4	8	13	–	–	3	$u_i < v_i$
5	12	20	–	–	–	for all $1 \leqslant i \leqslant n-1$ there is an edge $(i, i+1)$, there is an edge $(n, 1)$
6	13	21	–	–	5	the graph consists of one strongly connected component
7	8	14	–	–	0 – 6	

Problem Sand. Best Runner

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	1 second
Memory limit:	256 megabytes

There are n running tracks in the stadium with lengths a_1, a_2, …, a_n. There are also m runners, with the i-th runner starting at the beginning of track b_i.

All runners will train for T seconds. The training of a runner proceeds as follows:

Let the runner currently be at the beginning of track i. They will run to the end of the current track in a_i seconds. After that, they can either instantly return to the beginning of the current track, or move to the beginning of track $(i-1)$ (if $i > 1$), or to the beginning of track $(i+1)$ (if $i < n$). After this, they continue running from the track they moved to. Once the training duration reaches T seconds, they finish their training.

We define the best runner as the one who runs the most number of **full** tracks during the training time (there may be several such runners). Determine how many tracks the best runner will run.

Input

The first line contains three integers n, m, and T ($1 \le m \le n \le 300\,000$, $1 \le T \le 10^9$) — the number of tracks, the number of runners, and the duration of the training.

The second line contains n integers a_1, a_2, …, a_n ($1 \le a_i \le 10^9$) — the lengths of the tracks.

The third line contains m integers b_1, b_2, …, b_m ($1 \le b_1 < b_2 < \ldots < b_m \le n$) — the track numbers from which the runners start.

Output

Output a single integer — the maximum number of full tracks that one of the runners can run during the training time.

Examples

input	output

`5 3 10`
`4 5 2 7 1`
`1 2 4`
	`4`

`4 2 11`
`4 5 7 10`
`2 3`
	`2`

Note

In the first example, the runner starting on track 4 can run the most tracks: they should run track 4, then move to track 5 and run it 3 times.

In the second example, the runner starting on track 2 can run the second track 2 times.

Scoring

The tests for this problem consist of six groups. Points for each group are given only if all tests of the group and all tests of the required groups are passed.

Group	Points	Additional constraints			Required Groups	Comment
		n	T	a_i		
0	0	–	–	–	–	Examples.
1	23	$n \le 1000$	–	–	0	
2	10	–	–	–	–	$a_i \le a_{i+1}$ for all $1 \le i < n$
3	16	–	$T \le 20$	–	0	
4	19	–	–	$a_i \le 20$	0	
5	11	–	–	–	–	$m = n$
6	21	–	–	–	0 – 5	

Problem Glass. Card Flip

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	1 second
Memory limit:	256 megabytes

Petya and Vasya bought a new card game called ”Flip”. The game contains n double-sided cards and m single-sided cards:

1. **Double-sided card.** The front side of the card has the number a_i, and the back side has the number b_i.
2. **Single-sided card.** The front side of the card has a single number c_i.

All numbers written on the cards on all sides are distinct. Initially, all cards are placed on the table with the front side up. On his turn, a player can perform exactly one of two actions:

1. Remove from the table the card with the smallest number among the remaining cards.
2. If the card with the smallest number is double-sided and facing up, it can be flipped.

The player who removes the last card from the table wins. Determine the winner of the game if Petya goes first.

Input

The first line contains two integers n and m ($1 \le n, m \le 500\,000$) — the number of double-sided and single-sided cards, respectively.

The second line contains n integers a_1, a_2, …, a_n ($1 \le a_i \le 2 \cdot n + m$) — the numbers written on the front side of the double-sided cards.

The third line contains n integers b_1, b_2, …, b_n ($1 \le b_i \le 2 \cdot n + m$) — the numbers written on the back side of the double-sided cards.

The fourth line contains m integers c_1, c_2, …, c_m ($1 \le c_i \le 2 \cdot n + m$) — the numbers written on the single-sided cards.

It is guaranteed that each number from 1 to $2 \cdot n + m$ appears exactly once in one of the arrays a, b, or c.

Output

Print ”First” (without quotes) if Petya wins in this game, and ”Second” (without quotes) if Vasya wins.

Examples

input	output

2 1
5 3
1 2
4
	First

1 2
2
3
4 1
	Second

Note

In the first example, initially, the cards 3, 4, and 5 are on the table. To win, Petya discards card 3 on his turn, after which Vasya must discard card 4, as it is single-sided. Finally, Petya discards card 5, meaning there will be no cards left for Vasya, and Petya wins.

In the second example, initially, the cards 1, 2, and 4 are on the table. Petya must discard the first card, as it is single-sided. Then, to win, Vasya flips card 2, so there will be cards 3 and 4 on the table. Petya must discard the flipped card number 3, after which Vasya will discard card 4. Since the cards will run out, Vasya will win.

Scoring

The tests for this problem consist of nine groups. Points for each group are given only if all tests of the group and all tests of the required groups are passed. Please note that passing the example tests is not required for some groups. **Offline-evaluation** means that the results of testing your solution on this group will only be available after the end of the competition.

Group	Points	Additional constraints		Required Groups	Comment
		n	m		
0	0	–	–	–	Examples.
1	12	$n \le 20$	$m \le 10$	0	
2	13	$n \le 20$	–	0, 1	
3	9	–	–	–	$a_i > b_i$
4	10	–	–	–	$max_{i=1}^{n}(a_i) < min_{i=1}^{n}(b_i)$
5	6	–	–	–	The segments $[min(a_i, b_i);\ max(a_i, b_i)]$ do not intersect.
6	11	$n \le 200$	$m \le 200$	–	The segments $[min(a_i, b_i);\ max(a_i, b_i)]$ are nested or do not intersect.
7	14	–	–	5, 6	The segments $[min(a_i, b_i);\ max(a_i, b_i)]$ are nested or do not intersect.
8	13	$n \le 5000$	$m \le 5000$	0, 1, 6	
9	12	–	–	0 – 8	**Offline-evaluation.**

Problem Clay. Order Statistics

Input file:	`input.txt` or standard input
Output file:	`output.txt` or standard output
Time limit:	4 seconds
Memory limit:	1024 megabytes

You are given an array a_1, a_2, …, a_n, consisting of integers, as well as integers k and m. The following operation is performed m times on the array:

- Select i_1, i_2, …, i_k — the positions of the k largest elements in the array a. If two elements are equal, the element that appears earlier in the array is considered larger.
- Decrease a_{i_1}, a_{i_2}, …, a_{i_k} by 1.

For x from 1 to n, let $F_{m,k}(x)$ denote the value of the x-th order statistic in the array obtained from a after applying the operation m times with the given parameter k. For x from 1 to n, the x-th order statistic of the array $a_1, a_2, \ldots, a_n$ is the element that would be in position x if the array a were sorted in non-decreasing order.

For all l, r such that $1 \le l \le r \le n$, let $S_{m,k}(l, r)$ denote the sum of $F_{m,k}(x)$ for all integers x from l to r. More formally:

$$S_{m,k}(l, r) = \sum_{x=l}^{r} F_{m,k}(x)$$

You are given integers m_0 and k_0. You must compute the values of $F_{m_0,k_0}(x)$ for all x from 1 to n.

After that, you must process q queries. The j-th query $(1 \le j \le q)$ can be one of three types:

1. Compute the value of $F_{m_j,k_j}(x_j)$.
2. Change the value of a_{p_j} to v_j.
3. Compute the value of $S_{m_j,k_j}(l_j, r_j)$.

All calculations of the functions F and S are performed independently each time and do not change the array. All changes to the array in queries of the second type are preserved for subsequent queries.

Input

The first line contains four integers n, m_0, k_0, and q ($1 \le n \le 200\,000$, $0 \le m_0 \le 10^9$, $1 \le k_0 \le n$, $0 \le q \le 200\,000$) — the length of the array a; the number of operations; the number of largest elements decreased in each operation, and the number of queries.

The second line contains n integers $a_1,\ a_2,\ \ldots,\ a_n$ ($-10^9 \le a_i \le 10^9$, $1 \le i \le n$) — the elements of the array a.

The next q lines contain the queries. In the j-th query, the first number is t_j ($1 \le t_j \le 3$) — the type of the j-th query.

- If $t_j = 1$, then the next line contains three integers m_j, k_j, and x_j ($0 \le m_j \le 10^9$, $1 \le k_j, x_j \le n$) — the parameters of the first type query.
- If $t_j = 2$, then the next line contains two integers p_j and v_j ($1 \le p_j \le n$, $-10^9 \le v_j \le 10^9$) — the parameters of the second type query.
- If $t_j = 3$, then the next line contains four integers m_j, k_j, l_j, and r_j ($0 \le m_j \le 10^9$, $1 \le k_j, l_j, r_j \le n$, $l_j \le r_j$) — the parameters of the third type query.

Output

In the first line, output n integers $F_{m_0,k_0}(1), F_{m_0,k_0}(2), \ldots, F_{m_0,k_0}(n)$.

Then, for each first type query, output the value $F_{m_j,k_j}(x_j)$ on a separate line, and for each third type query, output the value $S_{m_j,k_j}(l_j, r_j)$ — the answer to the j-th query.

Example

input	output
`8 3 2 16`	`-1 -1 0 1 1 1 1 2`
`3 1 2 -1 0 2 -1 4`	`2`
`3 3 2 2 6`	`1`
`1 3 2 4`	`-4`
`3 4 5 3 5`	`-1`
`1 4 5 6`	`-1`
`2 5 -1`	`-1`
`2 6 3`	`1`
`1 3 2 1`	`2`
`1 3 2 3`	`3`
`1 3 2 4`	`7`
`1 3 2 8`	`8`
`1 0 5 6`	`4`
`2 1 5`	`2`
`3 1 3 7 8`	
`3 2 3 5 8`	
`3 3 3 4 7`	
`3 4 3 4 7`	

Note

In the example, $n = 8$, $m_0 = 3$, $k_0 = 2$, $q = 16$. Initially, the array a is $[3, 1, 2, -1, 0, 2, -1, 4]$. Let's see how the array will change if we apply the operation m_0 times with parameter k_0:

1. The array is $[3, 1, 2, -1, 0, 2, -1, 4]$. The two largest elements are in positions 1 and 8. They are decreased by 1, after which the array becomes $[2, 1, 2, -1, 0, 2, -1, 3]$.
2. The array is $[2, 1, 2, -1, 0, 2, -1, 3]$. The two largest elements are in positions 1 and 8. They are decreased by 1, after which the array becomes $[1, 1, 2, -1, 0, 2, -1, 2]$.
3. The array is $[1, 1, 2, -1, 0, 2, -1, 2]$. The two largest elements are in positions 3 and 6. They are decreased by 1, after which the array becomes $[1, 1, 1, -1, 0, 1, -1, 2]$.

We find that after applying the operation 3 times with parameter 2 to the array a, it becomes $[1, 1, 1, -1, 0, 1, -1, 2]$. If this array is sorted, it results in the array $[-1, -1, 0, 1, 1, 1, 1, 2]$. Thus, the order statistics are $F_{3,2}(1) = -1$, $F_{3,2}(2) = -1$, $F_{3,2}(3) = 0$, $F_{3,2}(4) = 1$, $F_{3,2}(5) = 1$, $F_{3,2}(6) = 1$, $F_{3,2}(7) = 1$, $F_{3,2}(8) = 2$.

In the example, we need to process 16 queries; let's analyze the first 10 of them in detail:

1. The first query is of type $t_1 = 3$, with parameters $m_1 = 3$, $k_1 = 2$, $l_1 = 2$, $r_1 = 6$ and requires computing the value of $S_{3,2}(2, 6)$. We have already computed the values of $F_{3,2}(x)$ for x from 1 to 8, from which we find the answer to the query:
$$S_{3,2}(2,6) = F_{3,2}(2) + F_{3,2}(3) + F_{3,2}(4) + F_{3,2}(5) + F_{3,2}(6) = (-1) + 0 + 1 + 1 + 1 = 2$$
2. The second query is of type $t_2 = 1$, with parameters $m_2 = 3$, $k_2 = 2$, $x_2 = 4$ and requires computing the value of $F_{3,2}(4)$. We have already computed it, and it equals 1.

3. The third query is of type $t_3 = 3$, with parameters $m_3 = 4$, $k_3 = 5$, $l_3 = 3$, $r_3 = 5$ and requires computing the value of $S_{4,5}(3, 5)$, that is, to calculate the sum of the order statistics from the third to the fifth in the array obtained from a after applying the operation $m_3 = 4$ times with parameter $k_3 = 5$. At the time of the third query, the array a is $[3, 1, 2, -1, 0, 2, -1, 4]$. The five largest elements are in positions $1, 2, 3, 6, 8$. Decreasing them by 1, we get the array $[2, 0, 1, -1, 0, 1, -1, 3]$. Applying the operation three more times, we get the array $[-1, -2, -2, -2, -1, -1, -1, 0]$. After sorting, it becomes $[-2, -2, -2, -1, -1, -1, -1, 0]$. Thus, the answer to the query is

$$S_{4,5}(3, 5) = F_{4,5}(3) + F_{4,5}(4) + F_{4,5}(5) = (-2) + (-1) + (-1) = -4$$

4. The fourth query is of type $t_4 = 1$, with parameters $m_4 = 4$, $k_4 = 5$, $x_4 = 6$. After applying four operations with parameter 5 and sorting the array a, it will be $[-2, -2, -2, -1, -1, -1, -1, 0]$, so the sixth order statistic is -1.

5. The fifth query is of type $t_5 = 2$, with parameters $p_5 = 5$ and $v_5 = -1$. It changes the value of a_5 to -1, after which the array a becomes $[3, 1, 2, -1, -1, 2, -1, 4]$.

6. The sixth query is of type $t_6 = 2$, with parameters $p_6 = 6$ and $v_6 = 3$. It changes the value of a_6 to 3, after which the array a becomes $[3, 1, 2, -1, -1, 3, -1, 4]$.

7. The seventh query requires finding the value of $F_{3,2}(1)$. At the time of the seventh query, the array a is $[3, 1, 2, -1, -1, 3, -1, 4]$. After applying 3 times the operation with parameter 2, it will be $[1, 1, 1, -1, -1, 2, -1, 2]$. The first order statistic of this array is -1.

8. The eighth, ninth, and tenth queries require finding the values of $F_{3,2}(3)$, $F_{3,2}(4)$ and $F_{3,2}(8)$, that is, the third, fourth, and eighth order statistics in the array $[1, 1, 1, -1, -1, 2, -1, 2]$. They are equal to -1, 1, and 2, respectively.

Scoring

The tests for this problem consist of eleven groups. Points for each group are given only if all tests of the group and all tests of the required groups are passed. Please note that passing the example tests is not required for some groups. **Offline-evaluation** means that the results of testing your solution on this group will only be available after the end of the competition.

If there are constraints on m or k in a subtask, they apply to both m_0 and k_0, as well as to the parameters of all first and third type queries.

The table with the groups is on the next page.

Group	Points	Additional Constraints				Required Groups	Comment
		n	m	k	q		
0	0	–	–	–	–	–	Examples.
1	4	$n \leq 1000$	$m \leq 1000$	–	$q = 0$	–	–
2	5	–	–	$k = 1$	$q = 0$	–	–
3	6	–	–	$k = 1$	$q \leq 100\,000$	2	$t_j = 1$ for all queries
4	7	–	–	$k = 1$	$q \leq 100\,000$	2, 3	$t_j \neq 3$ for all queries
5	11	–	–	$k = 2$	$q = 0$	–	–
6	9	–	$m \leq 10^6$	–	$q = 0$	1	–
7	10	$n \leq 1000$	–	–	$q = 0$	1	–
8	7	–	–	–	$q = 0$	1, 2, 5 – 7	–
9	11	–	–	–	$q \leq 100\,000$	1 – 3, 5 – 8	$t_j = 1$ for all queries
10	13	–	–	–	$q \leq 100\,000$	1 – 3, 5 – 9	$t_j \neq 2$ for all queries
11	9	–	–	–	$q \leq 100\,000$	0 – 10	–
12	8	–	–	–	–	0 – 11	**Offline-evaluation.**