Problem A. Important Guests

Input file:	guests.in
Output file:	guests.out
Time limit:	4 seconds
Memory limit:	512 megabytes

You are probably familiar with the famous dancing-master, professor Padegras, who is now preparing students of Mvodsk State University for their graduation ball. Tomorrow, the head of the University has important foreign guests who asked to show them all aspects of the students' life. That is why Padegras was asked to choose a small subset of his students for a promo-dance.

Professor has N boys and M girls in his class, and he knows all K mutual sympathy relations between them. For the planned dance, Padegras needs to choose exactly two boys and exactly two girls, and since the chosen dance is known to be very energetic and passionate, only the pairs with mutual sympathy are able to participate. Moreover, the pairs are going to mix many times during the dance, so all four possible mutual sympathy relations between each of the chosen boys and each of the chosen girls are required.

One more problem is that this event will occur tomorrow, meaning Padegras cannot be sure who of his students will be able to come. Cheer him up and calculate how many different subsets of exactly two boys and exactly two girls meet all the given requirements.

Input

The first line of input contains three integer numbers N, M and K: the number of boys, the number of girls and the number of mutual sympathy relations ($1 \le N, M \le 100\,000$, $1 \le K \le 200\,000$).

Each of next K lines describe one mutual sympathy relation by two integer numbers v_i ($1 \le v_i \le N$) and u_i ($1 \le u_i \le M$), denoting the boy and the girl in this relation. It is guaranteed that each possible pair occurs at most once.

Output

Print one number: the answer to the problem.

Examples

guests.in	guests.out
3 4 9 1 1 1 2 1 4 2 1 2 3 3 1 3 2 3 3 3 4	4

Problem B. Combinations Strike Back

Input file:	`combo.in`
Output file:	`combo.out`
Time limit:	4 seconds
Memory limit:	512 megabytes

Given a set containing n *different* elements, one can easily find the number of distinct k-element subsets of it. This value is the well-known binomial coefficient $\binom{n}{k} = \frac{n!}{k!\cdot(n-k)!}$.

However, this problem is a bit harder. You are a given a multiset with n (possibly equal) elements and you would like to answer several queries of the form "if we insert an additional element x into the multiset, how many distinct k-element submultisets it will have?". Note that, after each query, the given multiset remains **unchanged**.

Recall that a multiset is a generalization of the notion of set in which elements are allowed to appear more than once. A k-element submultiset of a multiset is obtained by selecting exactly k elements of the multiset (it is allowed to select equal elements). Two multisets are considered distinct if there is an element which occurs in one of them more times than in the other.

As the answers may be very large, you need to find each of them modulo 1 051 721 729.

Input

The first line of input contains an integer number n, the size of the multiset ($1 \le n \le 120\,000$). The next line contains n integer numbers a_i separated by spaces: the elements of the multiset ($1 \le a_i \le n$). Note that the elements are not necessarily pairwise distinct. The following line contains an integer number q, the number of queries ($1 \le q \le 120\,000$). The next q lines describe queries, each description consists of two integer numbers x and k ($1 \le x \le n$, $1 \le k \le n + 1$).

Output

For each query, output the required number of different submultisets, taken modulo 1 051 721 729.

Examples

combo.in	combo.out
6 1 2 2 3 3 3 4 1 2 2 3 3 4 4 5	6 7 6 8

Note

For the first query (if we add element `1`), the multiset will look like `1 2 2 3 3 3 1`, so it will have six different two-element submultisets: `(1 1)`, `(1 2)`, `(1 3)`, `(2 2)`, `(2 3)` and `(3 3)`.

Problem C. For Fun and Lulz

Input file:	joy.in
Output file:	joy.out
Time limit:	2 seconds
Memory limit:	512 megabytes

Together with a group of like-minded persons, you are investigating and inventing new awesome ways to spend free time. Sport games? Done. Board games? Done. ACM? No, thanks, we mean really free time. Random graphs? Yes, that is a freaking interesting one!

The process is simple. Initially, you get an empty graph: there are N nodes and no edges in it. On each step, a random edge is added to the graph. A random edge is formed in the following way:

- A random node v_i is chosen (each node can be chosen equiprobably).
- A random node u_i is chosen (each node can be chosen equiprobably, **including** v_i).
- A random integer w_i from the range $[1, 10^9]$ is chosen (each value can be chosen equiprobably).

It is guaranteed that the input data, except the example from the problem statement, was generated according to the way described above. The generation process used the quality of equiprobability that is more than enough to accept this task (actually, it was a pseudo-random linear congruential generator, as we see no reason to use Mersenne twister or something else in this problem).

Your goal is to determine the weight of the minimal spanning forest right after each new edge appears. Why? Because you can!

Input

The first line of input contains two integer numbers N and M ($1 \le N \le 100\,000$, $1 \le M \le 200\,000$). The next M lines describe the random edges in the order of their addition. Each description consists of three integers v_i, u_i and w_i ($1 \le v_i, u_i \le N$, $1 \le w_i \le 10^9$).

Output

Print M lines, each containing one integer number: the weight of the minimal spanning forest after adding the i-th edge.

Examples

joy.in	joy.out
5 8 1 2 4 1 1 1 2 3 3 4 5 2 5 1 4 1 3 1 1 3 2 2 5 3	4 4 7 9 13 10 10 9

Note

The example above is not generated randomly, but all other tests are generated using the method described.

Problem D. Batman Returns

Input file:	batman.in
Output file:	batman.out
Time limit:	3.5 seconds
Memory limit:	256 megabytes

Gotham City consists of a single street, and there are n skyscrapers located along it. They are numbered from west to east with integers from 1 to n, the height of the i-th skyscraper is equal to h_i meters.

Every night Batman performs an observation flight over the city. He climbs on the roof of some skyscraper and glides down to the roof of some other skyscraper. Due to the strong permanent wind he is only able to flight westward, but his altitude remains almost the same. Thus, he is able to glide down from skyscraper q to skyscraper p if and only if $p < q$ and $h_p < h_q$. Moreover, Batman is very manoeuvrable, so the height of the buildings between p and q don't matter. Batman cares a lot about the crime level in the city so he chooses such pair of valid p and q that $q - p$ is maximum possible.

City authorities have developed m plans of city renewal. According to the i-th plan only skyscrapers from l_i to r_i, inclusive will remain on this street, while others will be destroyed. For each plan i Batman wants to know the optimal plan to observe the city, namely such p_i and q_i that $l_i \le p_i < q_i \le r_i$, $h_{p_i} < h_{q_i}$ and $q_i - p_i$ is maximum possible.

Input

The first line of the input contains one integer n ($1 \le n \le 200\,000$) — number of skyscrapers on the street.

The second line contains n integers $h_i (1 \le h_i \le 200\,000)$ — heights of the skyscrapers.

Third line contains integer m ($1 \le m \le 200\,000$) — the number of plans designed by the city authorities.

Each of the last m lines contains two integers l_i and r_i ($1 \le l_i < r_i \le n$), denoting the range of the skyscrapers that will remain according to the i-th plan.

Output

For each renewal plan you should print two integers — optimal p_i and q_i. If there is no possible observation flight at all, you should print `-1 -1`.

If there are many optimal answers, you may print any one of them.

Examples

batman.in	batman.out
4 2 3 1 4 2 2 3 1 3	-1 -1 1 2
7 5 4 2 4 3 1 5 4 2 6 2 7 1 7 3 7	3 5 2 7 2 7 3 7

Note

Consider the first sample test. In the first query the only two available skyscrapers have heights 3 and 1 but they are not valid since $3 \geq 1$. In the second query the pair consisting of the first and the second skyscrapers is valid since they have heights 2 and 3.

Consider the second sample test. In the first query the pair of skyscrapers with heights 2 and 3, and the pair of skyscrapers with heights 2 and 4 are valid. The distance between first two of them is greater so this pair produces the answer for this query.

Problem E. Madness and Courage

Input file:	`heroes.in`
Output file:	`heroes.out`
Time limit:	7 seconds
Memory limit:	512 megabytes

Many of us have been dreaming of becoming a game developer. For some of us this even has become the reason to start studying computer science. And one day the dream came true for lucky Michael — he works as a software engineer in a top company ¡¡Snowstorm¿¿, which is famous for it's well-known masterpieces, such as ¡¡Military Art¿¿ and ¡¡Stellar Job¿¿.

Recently Michael has joined a group that works on a new astonishing role playing game ¡¡Madness and Courage¿¿. Its main feature is the possibility for the player to choose a new character at the beginning of every level.

Before a new level starts, there are N heroes available to choose from. Each hero is characterized by the amount of strength of his attack points a_i and the amount of initial health points b_i. The level is a long cave with M monsters waiting inside. Each monster also has strength of attack c_i and initial healh points d_i. The chosen heroe enters the cave and starts to fight with the first monster. If he survives in the battle, he fights against the second one, then the third and so on, until he dies or the level ends. Health points are not restored between the fights, that means the hero always starts a new fight with the smaller amount of health points, than in the previous one.

The hero versus monster fight constists of simultaneous strikes they give to each other. Each strike decreases the amount of the enemy's health points by the value of the striker's strength of attack. As soon as anyone's amount of health points turns non-positive, he dies and the fight ends. Please note, that according to these fight rules, it can happen that both due hero and monster will die simultaneously.

The company plans to make the game free and obtain money from selling different hints and additional content. Michael is going to create the hint, that tells every hero, how many monsters this hero will kill, if he is chosen by the player to fight at this level. As there may be an enormous number of heroes and monsters, Michael needs your help to compute the hint values.

Input

First line of the input contains two integers N and M ($1 \le N, M \le 200\,000$) — the number of heroes player can choose from and the number of monsters waiting in the cave respectively.

Each of next N lines contain hero description. For every hero two integer numbers a_i and b_i are given, they correspond to the hero's strength of attack and initial amount of health points ($1 \le a_i, b_i \le 10^9$).

Then follow M lines, each of them describing a single monster in the cave. For every monster values c_i and d_i ($1 \le c_i, d_i \le 200\,000$) are given, corresponding to his strength of attack and amount of health points. Monsters in the cave are situated in the same order as they appear in the input. That means, right after the hero enters the cave, he needs to kill the monster described in $N + 2$ line of the input. Right before the end of the level due hero will fight against the monster described in the line $N + M + 1$ of the input file.

Output

Print N lines containing one integer number each. On the i-th line print the answer for the i-th hero.

Examples

heroes.in	heroes.out
5 3 1 2 2 2 10 10 100 10 1 100 2 2 7 2 3 20	0 1 2 3 3

Note

First hero versus first monster fight will last for only one round, as the hero will be dead after it and the monster will be still alive with one health point left.

Second hero has absolutely the same parameters as the first monster. This means they will kill each other, so the answer for this hero is one.

If the player chooses the third hero to go through the cave, eight health points will be left after fighting due first monster, and only one health point after killing the second one. This hero need to give two strikes to kill third monster, but he will die after monster's first strike.

Fourth hero has the same amount of health points as the third, but has much greater strength of attack, so he will kill all the monsters, but will also die at the end of the fight with the last monster.

Fifth hero has minimal possible strength of attack, but is durable because of the high number of initial health points. He is the only hero to kill all the monsters and stay alive. After fighting first monster he will have 96 health points left, after fighting with due second — 82, and only 22 after killing the last monster.

Problem F. XOR and Favorite Number

Input file:	xor.in
Output file:	xor.out
Time limit:	2 seconds
Memory limit:	256 megabytes

Bob has a favorite number k and a_i of length n. Now he asks you to answer m queries.

Each query is given by a pair l_i and r_i and asks you to count the number of pairs of integers i and j, such that $l \le i \le j \le r$ and the xor of the numbers $a_i, a_{i+1}, \ldots, a_j$ is equal to k.

Input

The first line of the input contains integers n, m and k ($1 \le n, m \le 100\,000$, $0 \le k \le 1\,000\,000$) — the length of the array, the number of queries and Bob's favorite number respectively.

The second line contains n integers a_i ($0 \le a_i \le 1\,000\,000$) — Bob's array.

Then m lines follow. The i-th line contains integers l_i and r_i ($1 \le l_i \le r_i \le n$) — the parameters of the i-th query.

Output

Print m lines, answer the queries in the order they appear in the input.

Examples

xor.in	xor.out
6 2 3 1 2 1 1 0 3 1 6 3 5	7 0
5 3 1 1 1 1 1 1 1 5 2 4 1 3	9 4 4

Note

In the first sample the suitable pairs of i and j for the first query are: (1, 2), (1, 4), (1, 5), (2, 3), (3, 6), (5, 6), (6, 6). Not a single of these pairs is suitable for the second query.

In the second sample xor equals 1 for all subarrays of an odd length.

Problem G. Yet another stone game

Input file:	stonegame.in
Output file:	stonegame.out
Time limit:	2 seconds
Memory limit:	256 mebibytes

Vasya has friend Petya, that works in RIGS — Research Institute of Games with Stones. Yesterday, Petya was invited to play a new game, invented in RIGS. The game rules are quite simple. In the beginning of the game there are N stones on the table. The first player takes one stone from the table. Then on every move, if k stones were taken on the previous turn, than the player can take k or $k + 1$ stones. Player, that can not make a move loses. Vasya wants to win, of course. So he wants to know, who will win in the game — first player or second.

Input

Each line of the input will contain integer N ($1 \le N \le 10^5$). The last line will contain one integer 0.

Output

For each test case write the only integer — the answer for the problem. Adhere to the sample output format below as close as possible.

Example

stonegame.in	stonegame.out
1 2 3 0	Case #1: First player wins. Case #2: Second player wins. Case #3: Second player wins.