

A. Yet Another Short Statement

1 second, 256 megabytes

Given four positive integers l , r , k , and x , we want to find the k^{th} smallest positive integer in the range from l to r inclusive, which has a digit sum of exactly x .



Input

First line of the input contains a single integer t ($1 \leq t \leq 10^5$) — number of test cases.

Each test case contains four integers l, r, k, x ($1 \leq l \leq r \leq 10^{18}, 1 \leq k, x \leq 10^{18}$).

Output

For each test case, print the answer, asked in the statement, if there are at least k numbers that have digit sum of exactly x in the range from l to r . Print -1 otherwise.

input
6
1 10 1 1
1 10 2 1
1 10 3 1
1 100 3 1
2 10000000 10 5
546445 10000000 10 5
output
1
10
-1
100
131
1000202

B. Johnny English and Group Formation

1 second, 256 megabytes



After all these years as a spy Johnny English found the hardest task of his life. He has to divide some people into groups. There are n people standing in a line. The person at i^{th} belongs to the country c_i . Johnny English need to divide these people into groups following these conditions:

1. Each person must be in some group
2. Each group can consist of one or two people
3. No two people from the same country can be in the same group

Johnny English wants the total number of groups minimum. Now as his close friend you have to calculate the minimum number of groups he can divide those n people into.

Input

The first line of the Input contains two integers n ($1 \leq n \leq 10^5$) denoting the number of people. The next line contains n integers denoting c_i ($1 \leq c_i \leq 10^5$).

Output

Print the minimum number of groups Johnny English can divide those n people into.

input
6
1 2 3 1 1 5
output
3

input
6
1 2 2 1 2 2
output
4

C. Johnny English Strikes Again

3 seconds, 256 megabytes



After successfully forming groups once Johnny English became famous for group formation. Now the Prime Minister wants to test if Johnny English is actually good at group formation or if it was just a lucky shot. There are n people standing in a line. The person at i^{th} belongs to the country c_i . The Prime Minister asked Johnny English to divide these people into groups following these conditions:

1. Each person must be in some group
2. Each group can consist of one or two people
3. No two people from the same country can be in the same group
4. A VIP can't be in the same group with a non-VIP

Before starting the group formation Johnny English will ask you some queries. In the i^{th} query he will give you two integers l_i and r_i . For each query, you have to tell him the minimum number of groups he can divide

those n people into if and only if people from l_i^{th} position to r_i^{th} position are VIPs.

Input

The first line of the Input contains two integers n ($1 \leq n \leq 10^6$) and q ($1 \leq q \leq 10^5$) denoting the number of people. The next line contains n integers denoting c_i ($1 \leq c_i \leq 10^5$). i^{th} of the next q lines contains two integers l_i and r_i $1 \leq l_i \leq r_i \leq n$.

Output

For each query print one integer in a single line denoting the answer to the query.

input
6 9
2 3 3 1 1 1
3 5
4 5
1 5
1 1
1 6
3 5
3 4
3 3
1 6
output
4
4
4
4
3
4
3
4
3

D. Search For Beauty

2 seconds, 256 megabytes

Kalu has an integer number N . He defines the beauty of a number k as follows:

If k and N are co-prime, the beauty of k is $GCD(k - 1, N)$. Otherwise, the beauty is 0.

Kalu wants to find the **sum of beauty** for all integers k , such that $1 \leq k \leq N$.

However, Kalu is busy, so he asked you to write a program to help him with his task.

Notes

- $GCD(a, b)$, the greatest common divisor of two integers a and b , is the largest positive integer that divides both a and b without leaving a remainder.
- Two numbers are co-prime if their greatest common divisor (GCD) is 1.

Input

The first line of input contains an integer T ($1 \leq T \leq 10^5$), the number of test cases. Each of the next T lines contains a single integer N ($1 \leq N \leq 10^5$).

Output

For each test case, output a single integer, the sum of the beauty of all positive integers less than or equal to N .

input
1
5
output
8

For $N = 5$, we need to find the sum of beauty for all integers K , such that $1 \leq K \leq N$. Let's start by finding the beauty of each number K .

- For $K = 1$: 1 and 5 are co-prime, we have $GCD(1 - 1, 5) = GCD(0, 5) = 5$. So the beauty of $K = 1$ is 5.
- For $K = 2$: 2 and 5 are co-prime, we have $GCD(2 - 1, 5) = GCD(1, 5) = 1$. So the beauty of $K = 2$ is 1.
- For $K = 3$: 3 and 5 are co-prime, we have $GCD(3 - 1, 5) = GCD(2, 5) = 1$. So the beauty of $K = 3$ is 1.
- For $K = 4$: 4 and 5 are co-prime, we have $GCD(4 - 1, 5) = GCD(3, 5) = 1$. So the beauty of $K = 4$ is 1.
- For $K = 5$: 5 and 5 are not co-prime, so the beauty of $k = 5$ is 0.

Thus, the sum of beauty for all integers K , such that $1 \leq K \leq 5$ is $5 + 1 + 1 + 1 + 0 = 8$. Therefore, the answer for $N = 5$ is 8.

E. Tree query with update

1 second, 256 megabytes

You are given a tree of n nodes. Every node has given some value. You have to process some queries and update on the tree. Queries are of two types-

- 1 $u\ x$ — Update the value of node u to x .
- 2 $u\ v$ — Find the maximum value of all node of sub-tree v , if u is root of the tree.

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases. The description of test cases follows.

The first line of each test case contains a single integer n ($1 \leq n \leq 200000$) — number of node in the tree.

The second line of each test case contains n integers $a_1, a_2 \dots, a_n$ ($1 \leq a_i \leq 10^9$).

The i -th of the following $n - 1$ lines contains two integers x_i and y_i ($1 \leq x_i, y_i \leq n, x_i \neq y_i$), meaning that the i -th edge connects vertices x_i and y_i in the tree.

The next line of each test case contains a single integer q ($1 \leq q \leq 200000$).

Each of the following q lines contain three integers — a description of the next action in one of the following formats:

- 1 $u\ x$ — Update the value of node u to x ($1 \leq u \leq n, 1 \leq x \leq 10^9$).
- 2 $u\ v$ — Find the maximum value of all node of sub-tree v , if u is root of the tree ($1 \leq u, v \leq n$).

It is guaranteed that sum of n and q over all test is ≤ 200000 .

Output

Print the answer for every second type of every test cases.

input
2 6 5 7 8 3 8 6 1 2 5 4 4 1 5 6 3 2 5 1 6 2 2 1 4 1 1 9 2 4 2 1 3 1 7 9 6 4 1 2 5 2 1 2 3 1 6 2 2 5 3 7 4 1 3 2 5 5 2 3 4 1 4 8
output
8 8 9 1

F. Find GCD

1 second, 256 megabytes

Pt is a brilliant mathematician who has just discovered the power of the Euclidean algorithm for finding the GCD of two numbers. Excited to test her new knowledge, she shares her discovery with her friend, Kalu, who challenges her with a problem to solve. She needs your help to solve the problem.

The problem is to find the GCD of two numbers, n raised to the factorials of a and b , respectively, and then take the result modulo 1000000007. In other words, you are given three integers: a , b , and n , and you must find the value of:

$$GCD(n^{a!}, n^{b!}) \bmod 1000000007$$

Input

The first line of input consists of an Integer T ($1 \leq T \leq 10^5$) denoting the number of test cases. Each of the following T lines contains 3 integers a , b and n ($0 \leq a, b, n \leq 10^5$).

Output

For each test case print the GCD in a separate line.

input
3 1 1 2 1 7 4 3 3 2
output
2 4 64

$GCD(p, q)$ denotes the greatest common divisor of p and q .

$x!$ denotes factorial of x i.e. $x! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot x$ (in particular, $0! = 1$).

a modulo b (shortened $a \bmod b$) is the only integer c , such that $0 \leq c < b$ and $a - c$ is divisible by b .

G. Another Tree Query

2 seconds, 256 megabytes

You are given a tree of n nodes. Initially all node are disconnected. You have to process m query. Queries are of two kind.

1 $u\ v$ - add a bidirectional edge between node u and v . It is guaranteed that, there is no path from u to v before this operation.

2 u - Print two value a, v . Where a is maximum possible length of simple path starting from u . And v is such a node so that $dist(u, v) = a$.

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases. The description of each test cases follows.

The first line of each test case contains two integers n, m ($1 \leq n \leq 2 \cdot 10^5, n \leq m \leq 5 \cdot 10^5$) — number of node in the final tree and number of query.

Each of the following m lines contain three integers — a description of the next action in one of the following formats:

1 $u\ v$ — ($1 \leq u, v \leq n, u \neq v$).

2 u — ($1 \leq u \leq n$).

It is guaranteed that there will be $n - 1$ first type of query.

It is guaranteed that sum of n over all test is $\leq 2 \cdot 10^5$ and sum of m over all test case is $\leq 5 \cdot 10^5$.

Output

For each query of type 2 output the answer to it. If there are multiple answer, print any of them.

input
2 7 9 1 3 1 1 5 2 2 5 2 2 2 2 1 2 1 1 4 1 1 4 7 1 6 2 9 9 1 3 7 1 5 4 1 1 2 1 9 4 1 8 7 2 3 1 4 2 1 6 2 1 2 3
output
1 2 1 5 1 5 2 8

$dist(u, v)$ means number of edge in the simple path between u and v .

H. Sequential Nim

1 second, 256 megabytes

One day, Susu discovered a game that involved N piles of stones where the k^{th} pile has s_k stones. The game is played between two players. The players take turns removing stones from the piles with the following rules:

- A player can remove any positive number of stones from a single pile in turn.
- A player cannot remove stones from a pile until all the stones in the piles with lower indices have been removed.
- The player who cannot make a valid move loses.

There are two types of queries you need to handle:

Type 1: Update the size of the i^{th} pile to x .

Type 2: Given a range of piles L to R, determine which player will win the game if they play optimally.

As you are a skilled programmer, you decide to implement this game. Can you write a program to solve this game?

Input

The first line of input contains two integers N and Q ($1 \leq N, Q \leq 10^5$) — the number of piles and the number of queries, respectively. The second line of input contains N integers $s_1, s_2, \dots, s_N$ ($1 \leq s_i \leq 10^9$) — the initial sizes of the piles. Each of the following Q lines contains a query in the following format:

- 1 i x : update the i^{th} pile to size x ($1 \leq i \leq N, 1 \leq x \leq 10^9$).
- 2 L R : which player will win the game if they play optimally using only the piles from L to R ($1 \leq L \leq R \leq N$).

Output

For each query of type 2, output a single line containing either "First" or "Second", depending on which player will win the game if they play optimally.

input
5 3 20 1 1 2 5 2 1 3 1 5 3 2 3 5
output
First Second

I. The Secret Key

1.5 seconds, 256 megabytes

Kalu is a computer engineer working for an agency that needs to send and receive encrypted messages. The agency has given him a transmitter that contains a special number A and a receiver that contains a special number B . These two devices need to communicate with each other securely using a shared secret, which is an integer X .

The shared secret X must satisfy two properties:

- When you divide the transmitter's number A by X , the remainder is m_1 .
- When you divide the receiver's number B by X , the remainder is m_2 .

Your job is to write a program that finds the **smallest positive** integer X that satisfies these properties.

Note: When dividing the integer a by the integer b , the remainder is a unique integer c satisfying $0 \leq c < b$ and the property that $a - c$ is divisible by b .

Input

The first line of input consists of an Integer T ($1 \leq T \leq 5 \cdot 10^5$) denoting the number of test cases. Each of the following T lines contains four integers separated by a space: A, B, m_1 , and m_2 ($1 \leq A, B \leq 5 \cdot 10^5, 0 \leq m_1, m_2 \leq 5 \cdot 10^5$).

Output

For each test case, print the secret key X if it exists. Otherwise, print -1 .

input
3 2 4 0 0 3 6 1 0 10 10 2 6

output
1 2 -1

J. Magic Balls

2 seconds, 256 megabytes

Once a wise man said, "You need balls to do business". Now Mike has n balls and he wants to do business, i^{th} ball has color c_i ($1 \leq c_i \leq n$). He will sell exactly k of his balls. The price of a ball depends on its color. The price of a ball having color i is p_i .

Mike knows m types of magic operations. By applying the i^{th} type of operation, he can change the color of a ball from x_i to y_i . Before selling his balls Mike can apply as many operations as he wants on any number of balls.

What is the maximum amount of money Mike can make selling exactly k balls?

Input

First line of the input contains a single integer T ($1 \leq T \leq 10$) denoting number of test cases.

The first line of each test case contains three integers n, m ($1 \leq n, m \leq 10^5$) and k ($0 \leq k \leq n$). Next line contains n integer, i^{th} of which denotes c_i ($1 \leq c_i \leq n$). Next line contains n integers, i^{th} of them denotes p_i ($1 \leq p_i \leq 10^9$). i^{th} of the next m lines contains two integer x_i, y_i ($1 \leq x_i, y_i \leq n$).

Output

For each test case, print a single integer denoting the maximum money Mike can make by selling exactly k balls.

input
1 5 3 5 1 2 3 4 5 5 4 5 2 3 2 3 3 4 4 5
output
21

In the beginning, Mike uses the 1st magic operation and changes the color of 2nd ball from color 2 to 3. Then he uses the 3rd magic to change the color of 4th ball from color 4 to color 5. Thus the colors of the balls are now 1, 3, 3, 5, 5. As the price of a ball with color 1 and color 3 is 5 and a ball with color 5 is 3, he will get $5 + 5 + 5 + 3 + 3 = 21$ amount of money by selling all the balls.

K. Special Lattice Path

1.5 seconds, 256 megabytes

Little King Terence Tao lives in a 2D grid. He is very fond of food and his favorite restaurant is in (R_x, R_y) and his Castle is in $(0, 0)$ coordinate.

But there are some rules: With a single step, if his current position is in (p, q) coordinate then he can go to any of these five positions $(p - 1, q + 1), (p, q + 1), (p + 1, q + 1), (p + 1, q), (p + 1, q - 1)$ but he has to make sure that he has to stay in the first quadrant, which means he can't cross the axis but can stay on it. Tao can't visit the same co-ordinate twice.

Determine the number of ways he can go to his favorite restaurant.

Input

Input starts with an integer T ($1 \leq T \leq 50000$) — number of test cases.

Each case contains two integers R_x and R_y ($0 \leq R_x, R_y \leq 5 \cdot 10^7$) — coordinate of the restaurant.

In the first sample, the line can be represented as: |__ * __| Wherever the barrier is, Pt cannot move the stone to any position. So her winning probability is 0.

Output

For each case, print the number of ways Tao can visit his favorite restaurant from his castle in a single line. As the result can be huge, you should print the value modulo 1000000007 (the remainder when divided by 1000000007)

input
4 2 3 1 5 10 1 0 5
output
5142 64159 423247468 5142

L. Ultimate Game

1 second, 256 megabytes

As Pt is devastated with all the academic pressures, she decided to play a game with Kalu for relaxation.

There are two brick walls on the number line at positions 0 and N . Some stones are placed on integer positions on the line. There will be exactly one barrier on the line. The barrier is placed on a random position $i + 0.5$ where $0 \leq i < N$. Pt and Kalu will move the rocks in turn following these rules:

- A player must select a stone to move and move the stone by a positive integer distance.
- A stone to the left of the barrier can only be moved to the left.
- A stone to the right of the barrier can only be moved to the right.
- A stone cannot be moved over any other stone or the walls.
- Each point can contain at most one stone at a time.

One who can not move any other stone loses the game. The location of the barrier is unknown, and Pt starts the game. Print the probability of Pt winning the game assuming that both players play optimally. Formally, let the probability be an irreducible fraction $\frac{x}{y}$. Print the value $x \cdot y^{-1} \bmod 1000000007$. Where y^{-1} is an integer such that $y \cdot y^{-1} \equiv 1 \bmod 1000000007$. We ensure that y^{-1} exists.

Input

The first line will contain two integers, N and M ($2 \leq N \leq 10^6, 0 \leq M < N$) denoting the position of the right brick wall and the number of stones to move.

The second line of input contains M space separated unique integers $x_1, x_2, x_3, \dots, x_m$ ($0 < x_i < N$) denoting the positions of the stones.

Output

Print an integer denoting the probability of Pt winning modulo 1000000007.

input
2 1 1
output
0

input
4 1 2
output
1

