Problem A. Hide-And-Seek Game

Input file: `standard input`
Output file: `standard output`
Time limit: 5 seconds
Memory limit: 128 megabytes

During the summer vacation, Serenade and Rhapsody are playing hide-and-seek in a park structured as a tree. Each edge of the tree has a weight of 1. Serenade keeps running back and forth between S_a and T_a ($S_a \neq T_a$), while Rhapsody runs back and forth between S_b and T_b ($S_b \neq T_b$). However, Aria doesn't want to run around with them and only wants to know the **earliest** location where Serenade and Rhapsody will meet. Please output the identification number of this location.If they will never meet, output -1.

To be more specific, Serenade starts from S_a and moves one edge towards T_a each time. Once reaching T_a, Serenade then moves one edge towards S_a each time. After reaching S_a, Serenade moves one edge towards T_a each time, and so on. Rhapsody follows a similar pattern of movement.

Note that this park is quite **mysterious**, so Serenade and Rhapsody will **not meet on an edge** (you can assume that they will choose different paths to traverse the same edge).

Input

The input consists of multiple test cases. The first line contains a single integer $t(1 \leq t \leq 500)$ — the number of test cases. Description of the test cases follows.

The first line of each test case contains two integers n and m ($2 \leq n, m \leq 3 \cdot 10^3$) — the number of the vertices in the given tree and the number of questions.

Each of the next $n-1$ lines contains two integers u and v ($1 \leq u, v \leq n$, $u \neq v$) meaning that there is an edge between vertices u and v in the tree.

Each of the next m lines contains four integers S_a , T_a , S_b and T_b ($1 \leq S_a, T_a, S_b, T_b \leq n$, $S_a \neq T_a$ and $S_b \neq T_b$) .

It is guaranteed that the given graph is a tree.

The data guarantees that there will be no more than 20 groups with a value of n exceeding 400.

The data guarantees that there will be no more than 20 groups with a value of m exceeding 400.

Output

For each test case print a single integer — the identification number of this location which *Serenade* and *Rhapsody* will meet or -1.

Example

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

Problem B. City Upgrading

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	128 megabytes

The city where crazyzhk resides is structured as a tree. On a certain day, the city's network needs to be upgraded. To achieve this goal, routers need to be deployed. Each router covers the node it is placed on and its neighboring nodes. There is a cost a_i associated with placing a router at each node. The question is: How can the routers be deployed at minimum cost to ensure that every node is covered?

Input

The input consists of multiple test cases. The first line contains a single integer $t (1 \le t \le 1000)$ — the number of test cases. Description of the test cases follows.

The first line of each test case contains a integer n $(1 \le n \le 10^5)$ — the number of the vertices in the given tree.

The second line of each case are n integers $a_i (1 \le a_i \le 10^5)$,denoting the cost of setting up a router at each node.

Each of the next $n - 1$ lines contains two integers u and v $(1 \le u, v \le n,\ u \ne v)$ meaning that there is an edge between vertices u and v in the tree.

The data guarantees that the sum of n will not exceed $2 \cdot 10^5$

Output

For each test case print a single integer ——the minimum cost to ensure that every node is covered

Example

standard input	standard output
2 7 13 20 1 20 6 9 8 1 2 1 3 2 4 2 5 3 6 5 7 4 1 17 13 4 1 2 1 3 3 4	27 5

Problem C. Mr. Liang play Card Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	128 megabytes

Recently, Mr. Liang has become obsessed with a card game and cannot break free from it. The game goes like this: there are n cards arranged in a row from left to right. Each card has a type and a level (initially, all card levels are 1). You can perform the following operations an unlimited number of times:

Operation 1: Select a card and play it. Each card type has a value V_i. Playing a level 1 card yields a profit of V_i, playing a level 2 card yields a profit of $P \cdot V_i$, playing a level 3 card yields a profit of $P \cdot P \cdot V_i$ and so on. However, there is a restriction on the card level, with the maximum level being R.

Operation 2: Select two adjacent cards of the same type and level, and merge them into a higher-level card.

As his good friend, cv4456 would like to ask you what is the maximum profit Mr. Liang can obtain in the end?

Input

The input consists of multiple test cases. The first line contains a single integer $t(1 \le t \le 50)$ — the number of test cases. Description of the test cases follows.

The first line of each case are four integers,n,m,R,$P$$(1 \le n \le 100, 1 \le m \le 20, 1 \le R \le 20, 1 \le P \le 10)$. denoting the number of cards, types of cards, the upper limit of card levels, and the doubling coefficient for higher-level cards.

The second line of each case are n integers $a_i(1 \le a_i \le m)$,denoting the types of the n cards initially placed on the table.(all cards on the table is level 1)

The third line of each case are m integers $V_i(1 \le V_i \le 10^5)$,denoting the weight of each kinds of card.

The data guarantees that there will be no more than 10 groups with a value of n exceeding 20.

Output

For each test case print a single integer —the maximum profit Mr. Liang can obtain in the end.

Example

standard input	standard output
1 7 3 4 3 1 3 2 3 2 3 3 1 2 3	32

Problem D. Amazing spacecraft

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	128 megabytes

On this day, Sonetto purchased her first spacecraft (which can be considered as a convex polygon) and eagerly began to operate it. This spacecraft had a touch screen interface where the user could click on a position, and the spacecraft would instantly teleport to that location. However, since *Sonetto* bought a smuggled spacecraft, after Sonetto clicks on a location, the system randomly selects a point within a circle centered at Sonetto's clicked position with a radius of R, and the spacecraft teleport to that point.On this day, there was a Mr.Cookie's spacecraft parked in the vicinity, which can also be seen as a convex polygon. Now, given the position where Sonetto clicked on the screen, you are asked to calculate the probability of $Sonetto's$ spacecraft colliding with Mr.Cookie's spacecraft parked in the area.

Because the space where Sonetto is located is a rather mysterious space, Sonetto's spacecraft may initially intersect with Mr.Cookie's spacecraft. However, we don't need to be concerned about Sonetto's initial position. We only need to focus on whether the position of her spacecraft **after the instant teleportation** will collide with Mr.Cookie's spacecraft.

To be more specific, you are given two convex polygons A and B, and a circle P (centered at point X with radius R). You need to determine the probability of randomly selecting a point S within the circle P, such that when the convex polygon A moves along the vector $\vec{OS}$(O is the origin point (0,0)), it transforms into a new convex polygon A', and A' intersects with B (intersection implies that there exists a point w such that $w \in A'$ and $w \in B$).

Input

The input consists of multiple test cases. The first line contains a single integer $t(1 \le t \le 1200)$ — the number of test cases. Description of the test cases follows.

The second line contains a integer n ($3 \le n \le 30000$), denoting the number of vertices of the convex polygons A.

Then follows n lines, each line contains two integers x_i, y_i ($-10^8 \le x_i,\ y_i \le 10^8$), denoting the ith point of the convex polygon A. The points are given in counter-clockwise order.

The next line contains a integer m ($3 \le m \le 30000$), denoting the number of vertices of the convex polygons B.

Then follows m lines, each line contains two integers x_i, y_i ($-10^8 \le x_i,\ y_i \le 10^8$), denoting the ith point of the convex polygon B. The points are given in counter-clockwise order.

The last line contains three integers x,y and r,denoting the position of the center of the circle P and the radius of the circle. ($-10^8 \le x_i,\ y_i \le 10^8, 1 \le r \le 10^8$)

The data guarantees that the sum of n will not exceed $2 \cdot 10^5$

The data guarantees that the sum of m will not exceed $2 \cdot 10^5$

Output

For each test case print a single floating-point number denoting the probability of A' intersects with B.(keep 4 decimal places)

Example

standard input	standard output
2 5 0 -2 4 -1 4 0 1 1 0 0 4 0 -2 3 -1 2 1 1 0 -2 -2 3 4 -2 0 -1 -2 1 2 -1 2 3 2 0 5 1 3 1 1 -3 4	0.5247 0.1185

Problem E. Cyclically Isomorphic

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

If there exists an integer k such that string S becomes equal to string T after being **cyclically right-shifted** by k positions, then the strings S and T are said to be **cyclically right-shifted**.

Now, given n strings of length m consisting of **lowercase letters** , there are a total of Q queries. Each query provides two positive integers x and y. If the strings s_x and s_y are **cyclically right-shifted**, output 'Yes'; otherwise, output 'No'.

Input

The input consists of multiple test cases. The first line contains a single integer $T(1 \le T \le 5)$ — the number of test cases. Description of the test cases follows.

The first line of each test case contains two integers n and m $(1 \le n \times m \le 10^5)$— the number of the strings and the length of strings.

Each of the next n lines contains a string of lowercase letters s_i.

The next line contains a positive integer Q $(1 \le Q \le 10^5)$.

Each of the next Q lines contains two integers x, y $(1 \le x, y \le n)$ asks whether the string s_x and the string s_y are cyclic isomorphic.

Output

For each test case, output Q lines. Each line should contain a string indicating whether the current query strings s_x and s_y are cyclically isomorphic. If they are cyclically isomorphic, output 'Yes'; otherwise, output 'No'.

Example

standard input	standard output
2 2 2 ab ba 1 1 2 4 3 aab baa bba bab 6 1 2 1 3 1 4 2 3 2 4 3 4	Yes Yes No No No No Yes

Problem F. Escape The Maze

Input file:	`standard input`
Output file:	`standard output`
Time limit:	8 seconds
Memory limit:	512 megabytes

Alice is currently trapped in a maze, which can be seen as a tree. Each edge in the tree has a weight representing the length of that edge. The leaves of the tree represent the exits, and when Alice reaches a leaf, it means she has successfully escaped from the maze.

A leaf is defined as a node with degree 1 that is not the root.

Each maze has a difficulty level, denoted as L. When Alice is at a node x in the tree, she can choose to jump to a node y in her subtree. Let s be the sum of the edge weights along the path from x to y. The energy spent when jumping from x to y is $(s - L)^2$.

Alice wants to know the minimum amount of energy required to escape the maze if the tree has p as the root and she starts from p. Alice will ask this question a total of Q times.

The data guarantees that for any given pair of points x and y, the absolute value of the sum of edge weights s along the path between them does not exceed 10^9.

Input

The input consists of multiple test cases. The first line contains a single integer $T (1 \le T \le 5)$ — the number of test cases. Description of the test cases follows.

The first line of each test case contains two integers n, L $(3 \le n \le 10^5, -10^5 \le L \le 10^5)$— the number of nodes in the tree.

Each of the next $n - 1$ lines contains three integers u, v, w $(1 \le u, v \le n, u \ne v, -10^5 \le w \le 10^5)$.

The next line contains a positive integer Q $(1 \le Q \le 10)$.

Each of the next Q lines contains one integer p $(1 \le p \le n)$ asks the minimum amount of energy required to escape the maze if the tree has p as the root and she starts from p.

It is guaranteed that the given graph is a tree.

Output

For each test case, output Q lines. Each line should contain a integer indicating the minimum amount of energy required.

The data guarantees that the answer will not exceed the range that can be represented by a 64-bit signed integer.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>1
4 2
1 2 5
1 3 -4
1 4 6
4
1
2
3
4</pre></td><td><pre>9
1
0
0</pre></td></tr>
</table>

Problem G. Travel

Input file:	`standard input`
Output file:	`standard output`
Time limit:	12 seconds
Memory limit:	512 megabytes

Alice is currently traveling in a k-dimensional space. She starts at $(0, 0, \ldots, 0)$, and she has a total of n teleportation skills. When she uses a teleportation skill, if she is at $(x_1, x_2, \ldots, x_k)$ and the teleportation skill is $(y_1, y_2, \ldots, y_k)$, she will be teleported to $(x_1 + y_1, \ldots, x_k + y_k)$. Teleportation skills can be used in any way.

This k-dimensional space is finite, and the size of the i-th dimension is N_i. In other words, for the i-th dimension, the coordinate range is $0 \leq x_i \leq N_i$.

Alice's destination is $(N_1, \ldots, N_k)$. There are a total of m black holes in the k-dimensional space, which means Alice cannot reach those positions in any way. Alice wants to know how many ways she can use teleportation skills to reach her destination without passing through any black holes.

We consider two different teleportation plans to be different if and only if there exists a step where the two plans use different teleportation skills.

Two teleportation skills are considered different if and only if their skill numbers are different. Even if the effects of these two skills are exactly the same, they will still be considered as two separate skills.

Since the answer can be large, please output it modulo 998244353.

Input

The input consists of multiple test cases. The first line contains a single integer $T (1 \leq T \leq 5)$ — the number of test cases. Description of the test cases follows.

The first line of each test case contains one integers k $(2 \leq k \leq 6)$—This means that the space has a total of $k$dimensions.

The next line contains k positive integers N_i $(1 \leq N_i \leq 10^5)$, representing the size of the i-th dimension.

The next line contains two integers n and m , $(1 \leq n \leq 10^5, 0 \leq m \leq 1000)$, representing the number of teleportation skills and the number of obstacles, respectively.

The next n lines contain k non-negative integers $(y_1, \ldots, y_k)$, where $(\forall i \in [1, k], 0 \leq y_i \leq N_i)$, representing the current teleportation skills.

The next m lines contain k non-negative integers $(x_1, \ldots, x_k)$, where $(\forall i \in [1, k], 0 \leq x_i \leq N_i)$, representing the coordinates of the black holes.

It is guaranteed that there is no leap skill corresponding to $(0, 0, \cdots, 0)$.

The data ensures that $\prod_{i=1}^{k}(N_i + 1) \leq 2 \times 10^5$.

Output

For each test case, output an integer representing the number of ways Alice can reach her destination modulo 998244353.

Example

standard input	standard output
1 2 100 1000 2 0 1 0 0 1	555294450

Problem H. Umamusume

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

"Makea newtrack"is a new mode of game "Umamusume".During the game,there will be n rounds to improve attributes,each round can choose between rest,train or race.

rest:add 50 TP

train:spend 50 TP and add 15 speed points(TP less than 50 will fail)

race:spend 50 TP and add 100 G(TP less than 50 will fail)

At first round,you have 100 TP.

You can spend G in a special store to obtain items,special store will refresh the items that can be buy every 6 rounds(Round 6 will be the earliest time to buy items).The probability of each item appearing in the store is p(It can exist in a store and not sell anything and the number of one type of item is only one).Different item have different price and features:

item name	price	function
TP Medicine(L)	100G	add 100 TP
TP Medicine(M)	50G	add 50 TP
TP Medicine(S)	25G	add 25 TP
Magic Book(L)	100G	add 15 speed points
Magic Book(M)	50G	add 7 speed points
Magic Book(S)	25G	add 3 speed points
Horn	100G	next training speed points will become 2x
Weight	200G	next training speed points will spend 100 TP but become 3x

(Weight can not be used together with Horn)

Each item can be bought more than one time and you can use the item on any round after you buy it.

In order to obtain the strongest and fastest umamusume in the game,you know all the items in the store at first,and you are smart.You want to know the expected speed points.

Please output it modulo $10^9 + 7$.

Input

The input consists of multiple test cases. The first line contains a single integer $T(1 \le T \le 10000)$ — the number of test cases.

The each test case contains contains two integers $n$and p $(0 \le n \le 10^9, 0 \le p < 10^9 + 7)$—This means that the number of rounds and the probability of each item appearing in the store.

Output

For each test case, output an integer representing the expected speed points modulo $10^9 + 7$.

Example

standard input	standard output
3 2 1 5 500000004 27 500000004	30 45 857330545

Problem I. Assertion

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

Alice boldly asserts to you that if you divide m items into n groups, there will definitely be one group with a quantity of items greater than or equal to d.

Due to Alice's excessive self-confidence, she is unaware that some of her assertions are actually incorrect. Your task is to determine whether Alice's assertion is correct. If Alice's assertion is true, output 'Yes'; otherwise, output 'No'.

Input

The input consists of multiple test cases. The first line contains a single integer $T(1 \leq T \leq 10^5)$ — the number of test cases. Description of the test cases follows.

The first line of each test case contains three integers n, m, d $(2 \leq m \leq 10^9, 1 \leq n < m, 0 \leq d \leq 10^9)$, n and m represent the number of groups and the quantity of items, respectively, in Alice's assertion. The symbol d signifies Alice's claim that there will always be at least one group with a quantity of items greater than or equal to d.

Output

For each set of data, output a string. If Alice's assertion is correct, output 'Yes'; otherwise, output 'No'.

Example

standard input	standard output
3 1 2 1 2 3 2 3 10 4	Yes Yes Yes

Problem J. Easy problem I

Input file:	`standard input`
Output file:	`standard output`
Time limit:	9 seconds
Memory limit:	512 megabytes

note:The difference is that in this version,operation 1 is different,$n, m \le 2 \times 10^5$,$x_j \le x_{j+1}$.

For a given sequence of n intergers a.

There are two types of operations:

1 l r x_j $(1 \le l \le r \le n)$ — for each $i \in [l, r]$,change $a_i = |a_i - x_j|$.

2 l r $(1 \le l \le r \le n)$ — output ans $= \sum_{i=l}^{r} a_i$

tips:Due to the large input data, it may be necessary to FastIO.

Input

The input consists of multiple test cases. The first line contains a single integer $T (1 \le T \le 5)$ — the number of test cases.

The first line of each test case contains two integers n and $m, (1 \le n \le 2 \times 10^5, 1 \le m \le 2 \times 10^5)$ — the length of sequence and the number of operations.

The next line contains n integer a_i $(0 \le a_i \le 10^7)$

The next m line contains some integers opt, l, r, x $(1 \le opt \le 2, 1 \le l \le r \le n, 0 \le x \le 10^7)$ — indicating the operations.

Output

For each query, output an interger in a single line indicating the ans.

Example

standard input	standard output
1 5 5 1 2 3 4 5 1 1 5 3 2 1 2 2 2 4 1 2 3 5 2 1 5	3 2 14

Problem K. Easy problem II

Input file:	`standard input`
Output file:	`standard output`
Time limit:	8 seconds
Memory limit:	512 megabytes

note:The difference is that in this version,operation 1 is different,$n, m \leq 10^5$, x can take any possible value.

For a given sequence of n intergers a.

There are two types of operations:

$1 \quad l \quad r \quad x \quad (1 \leq l \leq r \leq n)$ — for each $i \in [l, r]$,change $a_i = \begin{cases} x - a_i & \text{if } a_i < x \\ \\ x + a_i & \text{if } a_i \geq x \end{cases}$.

$2 \quad l \quad r \quad (1 \leq l \leq r \leq n)$ — output $\text{ans} = \sum_{i=l}^{r} a_i$

Input

The input consists of multiple test cases. The first line contains a single integer $T(1 \leq T \leq 1)$ — the number of test cases.

The first line of each test case contains two integers n and $m, (1 \leq n \leq 10^5, 1 \leq m \leq 10^5)$— the length of sequence and the number of operations.

The next line contains n integer $a_i(0 \leq a_i \leq 10^7)$

The next m line contains some integers opt, l, r, x $(1 \leq \text{opt} \leq 2, 1 \leq l \leq r \leq n, 0 \leq x \leq 10^7)$ — indicating the operations.

Output

For each query, output an interger in a single line indicating the ans.

Example

standard input	standard output
1 5 5 1 2 3 4 5 1 1 5 3 2 1 2 2 2 4 1 2 3 5 2 1 5	3 14 32

Problem L. Play on Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Trees are very interesting, so playing on them is also very interesting.One day, Satono Diamond and Kitasan Black were playing a game on a tree.

First, Satono Diamond selected a point as the root of the tree.

Then Kitasan Black began the game. Each round selected a node x and deleted it and its subtree.

Whoever deleted the root node lost.

Kitasan Black wanted to know if she could win, but she didn't know which root node it was, so she assumed that the root node was randomly selected by Satono Diamond.Please tell her the probability of winning.

Please output it modulo $10^9 + 7$.

Input

The input consists of multiple test cases. The first line contains a single integer $T (1 \leq T \leq 3)$ — the number of test cases.

The first line of each test case contains one integer n $(1 \leq n \leq 2 \times 10^5)$ — the number of nodes in the tree.

Each of the next $n - 1$ lines contains two integers u, v $(1 \leq u, v \leq n, u \neq v)$

Output

For each query, output an integer in a single line indicating the probability of winning

Example

standard input	standard output
2 2 1 2 3 1 2 1 3	1 666666672