Problem A. Rikka with Lowbit

Input file:	`stdin`
Output file:	`stdout`
Time limit:	5s
Memory limit:	256MB

Today, Rikka is going to learn how to use BIT to solve some simple data structure tasks. While studying, She finds there is a magic expression $x\&(-x)$ in the template of BIT. After searching for some literature, Rikka realizes it is the implementation of the function lowbit(x).

lowbit(x) is defined on all positive integers. Let $a_1 \ldots a_m$ be the binary representation of x while a_1 is the least significant digit, k be the smallest index which satisfies $a_k = 1$. The value of lowbit(x) is equal to 2^{k-1}.

After getting some interesting properties of lowbit(x), Rikka sets a simple data structure task for you:

At first, Rikka defines an operator $f(x)$, it takes a non-negative integer x. If x is equal to 0, it will return 0. Otherwise it will return $x - \text{lowbit}(x)$ or $x + \text{lowbit}(x)$, each with the probability of $\frac{1}{2}$.

Then, Rikka shows a positive integer array A of length n, and she makes m operations on it.

There are two types of operations:

- 1 L R, for each index $i \in [L, R]$, change A_i to $f(A_i)$.
- 2 L R, query for the expectation value of $\sum_{i=L}^{R} A_i$. (You may assume that each time Rikka calls f, the random variable used by f is independent with others.)

Input

The first line contains a single integer $t (1 \leq t \leq 3)$, the number of the testcases.

The first line of each testcase contains two integers $n, m (1 \leq n, m \leq 10^5)$. The second line contains n integers $A_i (1 \leq A_i \leq 10^8)$.

And then m lines follow, each line contains three integers $t, L, R (t \in \{1, 2\}, 1 \leq L \leq R \leq n)$.

Output

For each query, let w be the expectation value of the interval sum, you need to output $(w \times 2^{nm})$ mod 998244353.

It is easy to find that $w \times 2^{nm}$ must be an integer.

Example

stdin	stdout
1 3 6 1 2 3 1 3 3 2 1 3 1 3 3 2 1 3 1 1 3 2 1 3	1572864 1572864 1572864

Problem B. Rikka with Burrows-Wheeler Transform

Input file:	`stdin`
Output file:	`stdout`
Time limit:	10s
Memory limit:	256MB

Burrows Wheeler-Transform(BWT) is a useful algorithm in text compression. It takes a string $s = s_1...s_n$ and returns another string with the same length. The algorithm has the following steps:

1. Generate all cyclic shift strings of s. Formally, for each $i \in [0, n)$, let $t_i = s_{i+1}...s_n s_1...s_i$
2. Sort all t_i in lexicographic order.
3. Generate the result string in which the ith char is equal to the last char of the ith smallest string.

For example, if $s = \text{abab}$, then $t_0 = t_2 = \text{abab}, t_1 = t_3 = \text{baba}$, the result of sorting will be $[\text{abab}, \text{abab}, \text{baba}, \text{baba}]$, and the result string will be bbaa.

Let $f(s)$ be the result string we can get when we do BWT on s.

Now, Rikka gets a **random** 01 string s of length n and m random queries $(l_{1,i}, r_{1,i}, l_{2,i}, r_{2,i})$. For each query, let $s_1 = s_{l_{1,i}}...s_{r_{1,i}}, s_2 = s_{l_{2,i}}...s_{r_{2,i}}$. Rikka knows it is easy to compare s_1 with s_2 in lexicographic order. To challenge you, Rikka wants you to compare $f(s_1), f(s_2)$ in lexicographic order.

In lexicographic order, $s = s_1...s_n$ is smaller than $t = t_1...t_m$ if and only if they meet at least one of the following two conditions:

1. s is a prefix of t and $n < m$.
2. there is an $i \in [1, \min(n, m)]$ which satisfies for all index $j \in [1, i), s_j = t_j$ and $s_i < t_i$.

Input

The first line contains a single integer $t (1 \leq t \leq 3)$, the numebr of testcases.

For each testcase, the first line contains three numbers $n, m, s (1 \leq n, m \leq 10^5, 0 \leq s \leq 10^9)$.

The input is generated by the seed s in the following steps:

1. Let A be a random number array of length 10^6 in which A_0=s, $A_i = (A_{i-1} \times 100000005 + 20150609) \bmod 998244353$.
2. $\forall i \in [1, n]$, $s_i = A_i \mod 2$.
3. $\forall i \in [1, m]$, let $a = A_{n+4(i-1)+1} \mod n + 1, b = A_{n+4(i-1)+2} \mod n + 1, c = A_{n+4(i-1)+3} \mod n + 1, d = A_{n+4(i-1)+4} \mod n + 1$, then $l_{1,i} = \min(a, b)$, $r_{1,i} = \max(a, b)$, $l_{2,i} = \min(c, d)$, $r_{2,i} = \max(c, d)$.

Output

For each query, if $f(s_1)$ is smaller than $f(s_2)$, output -1, if $f(s_1)$ is larger than $f(s_2)$, output 1. Otherwise output 0.

Example

stdin	stdout
2 5 5 0 100000 20 0	-1 0 1 -1 -1 1 1 1 -1 1 -1 -1 1 -1 -1 1 1 1 -1 1 1 -1 1 1 -1

Explanation

In the first testcase, string s is equal to 10111.

The queries are $(3, 3, 2, 4)$,$(1, 2, 2, 3)$,$(1, 4, 3, 5)$,$(4, 5, 2, 4)$,$(3, 4, 3, 5)$.

Problem C. Rikka with Rotate

Input file:	stdin
Output file:	stdout
Time limit:	10s
Memory limit:	256MB

To improve Rikka's math, Yuta designs a simple game which requires a lot of calculating. The game has several steps:

1. At first, the terminal chooses n points with integer coordinates in the two-dimensional Cartesian coordinate system and an integer K. And then, the terminal shows them to Rikka.
2. Rikka needs to color the n points using K different colors. Different points may have the same color, and some colors may not be used.
3. Rikka needs to count the number of different colorings.

Two colorings c_1, c_2 are the same if and only if there are some point o(may have real value coordinates) and angle α which satisfies if we rotate the n points in c_1 α degrees counterclockwise with the center o, the result will look exactly the same (including colors and positions) as c_2.

For example, when the points are $(1,0),(0,1)$ and K is 2, coloring $c_1 = (1,2)$ and $c_2 = (2,1)$ are the same, because if we rotate the points in c_1 180 degrees counterclockwise with the center $(\frac{1}{2}, \frac{1}{2})$, the result will be exactly the same as c_2. And in this case, there are 3 different colorings: $(1,1),(1,2),(2,2)$.

After playing this game several times, Rikka finds that the terminal always chooses points from a fixed point set P. So there are exactly $2^{|P|} - 1$ different games (all P's subsets except $\emptyset$).

Let $f(S,K)$ be the answer of the game when the point set chosen by the terminal is S and the number of colors is K. Now, given P and K, Rikka wants to calculate $\sum_{S \subseteq P, |S|>0} f(S,K)$.

Input

The first line contains one single integer $t (1 \le t \le 3)$, the number of the testcases.

For each testcase, the first line contains two integers $n, K (1 \le n \le 3000, 1 \le K \le 10^9)$.

Then n lines follow, each line contains two integers $(x_i, y_i)(|x_i|, |y_i| \le 10^8)$, describe the points in P.

The input guarantees that no two points have the same coordinates, i.e., $\forall i \neq j$, $(x_i, y_i) \neq (x_j, y_j)$

Output

For each testcase, output a single integer, the answer modulo 998244353.

Example

stdin	stdout
2 2 2 1 0 0 1 5 3 0 0 2 0 0 2 2 2 1 1	7 747

Problem D. Rikka with Prefix Sum

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2s
Memory limit:	256MB

Prefix Sum is a useful trick in data structure problems.

For example, given an array A of length n and m queries. Each query gives an interval $[l, r]$ and you need to calculate $\sum_{i=l}^{r} A_i$. How to solve this problem in $O(n+m)$? We can calculate the prefix sum array B in which B_i is equal to $\sum_{j=1}^{i} A_i$. And for each query, the answer is $B_r - B_{l-1}$.

Since Rikka is interested in this powerful trick, she sets a simple task about Prefix Sum for you:

Given two integers n, m, Rikka constructs an array A of length n which is initialized by $A_i = 0$. And then she makes m operations on it.

There are three types of operations:

- 1 L R w, for each index $i \in [L, R]$, change A_i to $A_i + w$.
- 2, change A to its prefix sum array. i.e., let A' be a back-up of A, for each $i \in [1, n]$, change A_i to $\sum_{j=1}^{i} A'_j$.
- 3 L R, query for the interval sum $\sum_{i=L}^{R} A_i$.

Input

The first line contains a single number $t (1 \le t \le 3)$, the number of the testcases.

For each testcase, the first line contains two integers $n, m (1 \le n, m \le 10^5)$.

And then m lines follow, each line describes an operation$(1 \le L \le R \le n, 0 \le w \le 10^9)$.

The input guarantees that for each testcase, there are at most 500 operations of type 3.

Output

For each query, output a single line with a single integer, the answer modulo 998244353.

Example

stdin	stdout
1 100000 7 1 1 3 1 2 3 2333 6666 2 3 2333 6666 2 3 2333 6666	13002 58489497 12043005

Problem E. Rikka with Equation

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2s
Memory limit:	256MB

Today, Yuta gives Rikka a simple math task: given a positive integer array A of length n and a positive integer m, does the equation $(\sum_{i=1}^{n} A_i x_i) \equiv 0 \mod m$ have any solutions?

Rikka solves this problem easily: $\forall i \in [1, n], x_i = 0$ is always a solution of this equation.

And then, Yuta shows a much harder version of this task:

For a positive integer array A of length n and a positive integer m, let $f(A, m)$ be the number of different integer vectors x which satisfy $x_i \in [0, m)$ and $(\sum_{i=1}^{n} A_i x_i) \equiv 0 \mod m$.

For example, when $A = [1, 1], m = 2$, there are 2 integer vectors $[0, 0], [1, 1]$ satisfy the previous conditions. So $f([1, 1], 2)$ is equal to 2.

Now Yuta shows a positive integer array B of length n and a positive integer M. B has $N = 2^n - 1$ non-empty subsequences $A_1, ..., A_N$. For each integer $m \in [1, M]$, Yuta wants to know $\sum_{i=1}^{N} f(A_i, m)$.

For example, when $B = [1, 1]$ and $M = 3$, Rikka needs to calculate $f([1], 1) \times 2 + f([1, 1], 1)$, $f([1], 2) \times 2 + f([1, 1], 2)$ and $f([1], 3) \times 2 + f([1, 1], 3)$.

This task is too hard for Rikka. So she wants you to help her.

Input

The first line contains a single integer $t (1 \leq t \leq 3)$, the numebr of the testcases.

For each testcase, the first line contains two integers $n, M (1 \leq n, M \leq 10^5)$. The second line contains n positive integers $B_i (1 \leq B_i \leq 10^5)$.

Output

For each testcase, let w_m be equal to $(\sum_{i=1}^{N} f(A_i, m)) \mod 998244353$. You only need to output a single line with a single integer, $\oplus_{i=1}^{M} w_i$, i.e., the exclusive OR sum of all w_i.

Example

stdin	stdout
2 5 5 1 2 3 4 5 10 10 1 2 3 4 5 6 7 8 9 10	1079 933958261

Problem F. Rikka with Line Graph

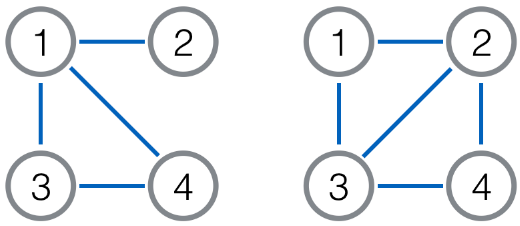
Input file:	`stdin`
Output file:	`stdout`
Time limit:	10s
Memory limit:	256MB

Line Graph $L(G)$ can be considered as an operator on an undirected graph G just like Complementary Graph and Dual Graph.

Rikka generalizes Line Graph to edge-weighted undirected graphs. For a graph $G = \langle V, E \rangle$, $L(G)$ is still an edge-weighted undirected graph which is constructed in the following way:

1. $L(G)$ has $|E|$ vertices and the ith vertex corresponds to the ith edge in G.
2. There is an edge between i, j in $L(G)$ if and only if edge i and j have at least one common vertices in G. And the edge weight is equal to the sum of the weights of edge i and j in G.

For example, in the following picture, the right graph is the line graph of the left one. Vertex $1, 2, 3, 4$ in $L(G)$ correspond to edge $(1, 2), (1, 4), (1, 3), (3, 4)$ in G. And if all edges in the left graph have weight 1, the edges in the right graph will have weight 2.



Now, Rikka has an edge-weighted undirected complete graph G with n vertices. And she constructs a graph $G' = L(G)$. It is clear that G' is connected.

Let $d(i, j)$ be the length of the shortest path between vertex i, j in G'(the length of each edge is equal to its weight), K be the number of vertices in G', Rikka wants you to calculate $\sum_{i=1}^{K} \sum_{j=i+1}^{K} d(i, j)$.

Input

The first line contains a single number $t (1 \leq t \leq 3)$, the number of the testcases.

For each testcase, the first line contains one single integer $n (2 \leq n \leq 500)$.

Then n lines follow, each line contains n integers $w_{i,j} (0 \leq w_{i,j} \leq 10^9)$, the weight of each edge in G. Since there are no self circles in G, the value of $w_{i,i}$ is meaningless.

The input guarantees that for all $1 \leq i \neq j \leq n$, $d_{i,i} = 0$ and $d_{i,j} = d_{j,i}$.

Output

For each testcase, output a single line with a single number, the answer modulo 998244353.

Example

stdin	stdout
3 2 0 1 1 0 3 0 1 1 1 0 1 1 1 0 4 0 1 1 1 1 0 2 2 1 2 0 3 1 2 3 0	0 6 56

Problem G. Rikka with Shortest Path

Input file:	stdin
Output file:	stdout
Time limit:	3s
Memory limit:	256MB

Random Graph is an important model in CS, and Shortest Path is a basic topic in graph theory. This problem is a simple combination of them.

Rikka wants to generate an undirected graph with n vertices. It it clearly that there are $\frac{n(n-1)}{2}$ possible edges. For each edge, Rikka tosses a special coin which has the probability of $\frac{p}{10^6}$ to be the head side and probability of $1 - \frac{p}{10^6}$ to be the tail side. If the result is the head side, Rikka adds this edge to the graph. Otherwise, she does nothing.

As a result, Rikka gets an undirected graph G. And then she chooses a vertex from all n vertices randomly with equal probability and calculates the length of the shortest path from the chosen vertex to n(the length of each edge is 1.) If the chosen vertex can not reach n, Rikka will set the answer to 10^9.

Anyway, at last Rikka will get an integer w, it may be 10^9, or may represent the length of some path in G. Now, Rikka wants you to calculate the expected value of $w \times n \times 10^{6n^2}$. It is easy to find that the answer must be an integer.

Input

The first line contains a single integer $t (1 \leq t \leq 5)$, the number of the testcases.

For each testcase, the first line contains two integers $n, p (1 \leq n \leq 400, 0 \leq p \leq 10^6)$.

Output

For each testcase, output a single line with a single integer, the answer modulo 998244353.

Example

stdin	stdout
5 2 1000000 2 0 4 12345 10 12345 100 12345	276262510 466613154 86698890 508893397 860797923

Problem H. Rikka with Ants

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2s
Memory limit:	256MB

There are two small ants on Rikka's desk. If we consider Rikka's desk as a two-dimensional Cartesian coordinate system, both of them have coordinate $(1, 0)$.

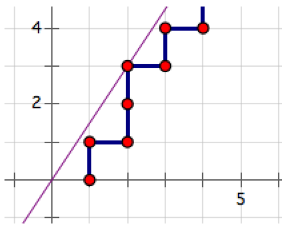
Now, Rikka places three obstacles on her desk:

1. $y = 0$, none of the two ants can walk cross this line.
2. $y = \frac{a}{b}x(a, b > 0)$, only the second ant can walk cross this line.
3. $y = \frac{c}{d}x(c, d > 0)$, only the first ant can walk cross this line.

It's remarkable that it is allowed for the ants to stand exactly on the obstacles. For example, if $a = b = c = d = 1$, then both of the ants can reach $(1, 0), (1, 1), (2, 1), (2, 2)$ and none of them can reach $(1, 2), (2, 3)$.

Now, the ants start to move. Their strategy is very simple: In each second, let (x, y) be the current coordinate of one ant, if it can reach $(x, y + 1)$, it will walk to this point, otherwise it will walk to $(x + 1, y)$.(Since $\frac{a}{b}, \frac{c}{d} > 0$, if the ant can reach (x, y), it can also reach $(x + 1, y)$).

The following image shows the first ant's path when $a = 3, b = 2$:



Now, given a, b, c, d, let p_1 be the first ant's path and p_2 be the second ant's path. Rikka wants you to calculate the number of the points with integer coordinates which are on both p_1 and p_2.

Input

The first line contains a single integer $t(1 \leq t \leq 10^5)$, the number of the testcases.

For each testcase, the first line contains four integers $a, b, c, d(1 \leq a, b, c, d \leq 10^9)$.

Output

For each testcase, output a single line with a single number, the answer modulo 998244353. If p_1, p_2 have infinite many common points, output -1.

Example

stdin	stdout
5 1 1 1 1 1 2 1 1 1 3 2 1 1 100 1 99 12 34 56 78	-1 2 1 5049 3

Problem I. Rikka with Zombies

Input file:	`stdin`
Output file:	`stdout`
Time limit:	3s
Memory limit:	256MB

Recently, Yuta designed and published a doomsday survival game.

The background of the game is a small town. There are n houses and several undirected roads connecting between them. As the game progresses, some zombies will appear in some houses, and they can walk to other houses through the roads. To survival from the zombies, players can build fences on the roads or even destroy some roads (But the map must be connected, i.e., for each pair of houses (i, j), there must be at least one path from i to j during the game).

Now Rikka is playing this game. To give her a better experience, Yuta tells her there will be m zombies, the ith zombie will appear at the a_ith house, and it can walk through the fences with height less than h_i units.

Using this information, Rikka finishes her defensse measures: she destroys as many roads as possible, and there are only $n - 1$ roads now(The map is still connected). On each of the remaining roads, Rikka builds a fence on it. Since the game has some kind of randomness, the height of the fence on the ith road is randomly selected from all integers in $[l_i, r_i]$ with equal probability.

A house is safe if and only if no zombie can reach this house. Now Rikka wants to calculate the probability of existing at least one safe houses.

Input

The first line contains a single integer $t(1 \le t \le 5)$, the number of the testcases.

For each testcase, the first line contains two integers $n, m(1 \le n, m \le 2000)$, the number of the houses and the zombies.

Then $n - 1$ lines follow, each line contains four integers $u_i, v_i, l_i, r_i(1 \le u \ne v \le n, 1 \le l_i \le r_i \le 10^9)$, describes a road and the fence on it.

Then m lines follow, each line contains two integers $a_i, h_i(1 \le a_i \le n, 1 \le h_i \le 10^9)$, describes a zombie.

Output

For each testcase, if the answer's simplest fraction representation is $\frac{x}{y}(\gcd(x, y) = 1)$, you need to output $x \times y^{-1} \mod 998244353$.

Example

stdin	stdout
2 4 2 1 2 1 2 2 3 1 2 1 4 1 2 1 2 3 2 5 2 1 2 1 10 2 3 2 9 1 4 3 12 2 5 4 6 1 7 5 5	374341633 888437475

Problem J. Rikka with Nickname

Input file:	stdin
Output file:	stdout
Time limit:	2s
Memory limit:	256MB

Sometimes you may want to write a sentence into your nickname like "lubenwei niubi". But how to change it into a single word? Connect them one by one like "lubenweiniubi" looks stupid.

To generate a better nickname, Rikka designs a non-trivial algorithm to merge a string sequence $s_1 ... s_n$ into a single string. The algorithm starts with $s = s_1$ and merges $s_2 ... s_n$ into s one by one. The result of merging t into s is the shortest string r which satisfies s is a prefix of r and t is a subsequence of r.(If there are still multiple candidates, take the lexicographic order smallest one.)

String s is a prefix of r if and only if $|s| \le |r|$ and for all index $i \in [1, |s|]$, $s_i = r_i$.

String s is a subsequence of r if and only if there is an sequence $a_1 ... a_{|s|}$ $(1 \le a_1 < a_2 < ... < a_{|s|} \le |r|)$ which satisfies for all index $i \in [1, |s|]$, $s_i = r_{a_i}$.

For example, if we want to generate a nickname from "lubenwei niubi", we will merge "niubi" into "lubenwei", and the result is "lubenweiubi".

Now, given a sentence $s_1 ... s_n$ with n words, Rikka wants you to calculate the resulting nickname generated by this algorithm.

Input

The first line contains a single number $t (1 \le t \le 3)$, the number of testcases.

For each testcase, the first line contains one single integer $n (1 \le n \le 10^6)$.

Then n lines follow, each line contains a lowercase string $s_i (1 \le |s_i| \le 10^6, \sum_{i=1}^{n} |s_i| \le 10^6)$.

Output

For each testcase, output a single line with a single string, the result nickname.

Example

stdin	stdout
2 2 lubenwei niubi 3 aa ab abb	lubenweiubi aabb