Problem A. Build a Computer

You want to build a computer to achieve a specific functionality: Given an integer x, determine whether x lies within the interval $[L, R]$. To accomplish this, you designed a directed acyclic graph (DAG) with edge weights of 0 and 1, which contains a starting node with an indegree of 0 and an ending node with an outdegree of 0. By starting from the starting node and following a path to the ending node, the sequence of the traversed edge weights forms a binary representation of an integer within the range $[L, R]$ without leading zeros. Every integer within the range $[L, R]$ must correspond to exactly one unique path in this graph. In this way, you can determine whether an integer lies within the range $[L, R]$ by checking if its binary representation can be constructed by traversing this DAG.

Clearly, you could separate the corresponding path for each integer into individual chains. However, you realized that for a large range, such a DAG would require too many nodes, and the computer you built with only 256 MiB of memory cannot store it. Therefore, you need to compress this DAG, allowing different paths to share nodes, in order to reduce the number of nodes and edges. Formally, you need to construct a DAG with no more than 100 nodes, where each node has an outdegree of at most 200. The DAG must have edge weights of 0 and 1, with exactly one starting node with an in-degree of 0 and one ending node with an out-degree of 0. Every integer in the range $[L, R]$ must correspond to **exactly** one unique path from the start to the end in this DAG, and no path should represent any integer outside the range $[L, R]$. Note that none of the binary sequences formed by any path in the graph should have leading zeros. There may be two edges with different weights between two nodes.

Input

A single line containing two positive integers L, R ($1 \le L \le R \le 10^6$).

Output

The first line should output the number of nodes n ($1 \le n \le 100$).

For the next n lines, the i-th line should start with an integer k ($0 \le k \le 200$), representing the number of outgoing edges from node i. Then output $2 \cdot k$ integers $a_{i,k}, v_{i,k}$ ($1 \le a_{i,k} \le n$, $a_{i,k} \ne i$, $v_{i,k} \in \{0, 1\}$), which means that node i has a directed edge with weight $v_{i,k}$ to node $a_{i,k}$. You must ensure that the output represents a directed acyclic graph that satisfies the requirements.

Example

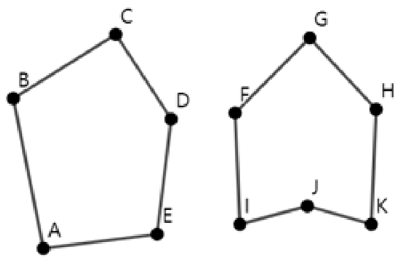
standard input	standard output

5 7	8
	3 2 1 3 1 4 1
	1 5 0
	1 6 1
	1 7 1
	1 8 1
	1 8 0
	1 8 1
	0

Problem B. Concave Hull

A simple polygon is a closed curve in the Euclidean plane consisting of straight line segments meeting end-to-end. Two line segments meet at every endpoint, and there are no other points of intersection between the line segments.

Simple polygons can be categorized into two types: convex and concave. A convex polygon is defined as a polygon where, for any two points inside it, all points on the line segment between these two points also lie inside the polygon, either within its interior or on its boundary. A simple polygon that is not convex is called a concave polygon. As shown in the figure below, the left one is a convex polygon, while the right one is a concave polygon.



Now, given n points such that all points are distinct and no three points are collinear, your task is to select some of these n points (maybe all of them) and connect them in any order to form a **concave polygon** with a strictly positive area. You need to determine the maximum possible area of the concave polygon that can be formed.

Input

The first line contains an integer T ($1 \le T \le 10^4$), indicating the number of test cases.

For each test case, the first line contains a positive integer n ($3 \le n \le 10^5$), indicating the number of points.

The next n lines each contain two integers x_i, y_i ($-10^9 \le x_i, y_i \le 10^9$), representing the coordinates of each point. It is guaranteed that all points are distinct, and no three points are collinear.

The sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, if it is not possible to form a concave polygon with a strictly positive area, output -1; otherwise, output a positive integer representing twice the maximum area of the concave polygon that can be formed. It can be proven that this answer is always a positive integer.

Example

standard input	standard output

2
6
-2 0
1 -2
5 2
0 4
1 2
3 1
4
0 0
1 0
0 1
1 1
	40
	-1

Problem C. Giving Directions in Harbin

In some regions, people are more accustomed to giving directions using cardinal directions, such as: go south to the second intersection, then head east to the first intersection. However, due to the complex road network planning in Harbin, many streets do not align perfectly with cardinal directions. Thus, if you provide directions using absolute directions to someone who has lived in Harbin for a long time, they may struggle to understand your intended route.

In Harbin, people are more accustomed to using relative directions to give guidance. For the same location, a Harbin resident might first instruct you to face south, and then say: walk straight along the road to the second intersection, then turn left, and then straight to the first intersection.

To address this difference, you decide to write a program that converts the direction-giving style using cardinal directions into the style preferred by Harbin residents. Of course, using a real map of Harbin would be too complicated, so in this problem, you can assume the map is an infinitely large grid.

Input

The first line contains an integer T ($1 \le T \le 10^4$), indicating the number of test cases.

For each test case, the first line contains an integer n ($1 \le n \le 10$), indicating the number of direction instructions.

The next n lines each describe an instruction in absolute position, consisting of a character d ($d \in \{\texttt{N}, \texttt{S}, \texttt{W}, \texttt{E}\}$) and an integer x ($1 \le x \le 10$), indicating "go to the x-th intersection in the d direction." Here, `N` represents north, `S` represents south, `W` represents west, and `E` represents east.

It is guaranteed that two consecutive instructions will not have the same direction or opposite directions (north and south are opposite, as are west and east).

Output

For each test case, the first line outputs an integer m ($1 \le m \le 20$) and a character f ($f \in \{\texttt{N}, \texttt{S}, \texttt{W}, \texttt{E}\}$), representing the number of instructions in Harbin style and the initial facing direction, with the same meanings for directions as in the input.

Next, output m lines. Each line starts with a character $g \in \{\texttt{Z}, \texttt{L}, \texttt{R}\}$, where `Z` means to go straight, `L` means to turn left, and `R` means to turn right. If the character is `Z`, the line must also include an integer y ($1 \le y \le 100$), representing going straight to the y-th intersection. The first output instruction must start with `Z`. Consecutive instructions cannot have the same character g, and `L` and `R` instructions cannot be adjacent.

In this problem, you do not need to minimize m. If there are multiple ways to reach the same destination, any valid solution is acceptable.

Example

standard input	standard output
1 2 S 2 E 1	3 S Z 2 L Z 1

Problem D. A Simple String Problem

You are given a character grid with 2 rows and n columns, where each cell contains a lowercase letter. You can start at any position in the grid and move several steps, with each step either to the right or downward, stopping at any cell. Concatenating the characters from the cells visited in order forms a string.

A string S is called a double string if and only if there exists a non-empty string T such that $S = TT$. For example, `aa` and `xyzxyz` are double strings, while `a` and `xyzyz` are not.

Given the character grid, find the length of the longest double string you can obtain.

Input

The first line contains an integer n ($1 \leq n \leq 2 \times 10^5$), representing the number of columns in the character grid.

The next two lines contain two strings of length n consisting only of lowercase English letters, representing the character grid.

Output

Output a single integer, representing the length of the longest double string you can obtain.

Examples

standard input	standard output
5	6
abcab	
acabc	
6	6
babbaa	
babaaa	
2	0
ne	
fu	

Note

In the first example, the longest double string can be obtained as follows (not unique):

$$\underline{\texttt{abc}}\texttt{ab}$$
$$\texttt{ac}\underline{\texttt{abc}}$$

Problem E. Marble Race

Marble race is a fun way to play with marbles, and today you want to give it a try.

There are n starting points on the negative half of the x-axis, with the i-th point located at x_i. There are m marbles in total, where m is an odd number, and the i-th marble has a speed of v_i. In a race, each marble randomly chooses a starting point with equal probability, and different marbles can choose the same starting point. Then, all the marbles start moving simultaneously towards the positive direction of the x-axis. Let c_i be the starting point chosen by the i-th marble. At time t, the coordinate of the i-th marble is given by $x_{c_i} + v_i \cdot t$.

You are a unique marble race enthusiast and do not care which marble is the fastest. Instead, you want to find out the exact time when the **median** of all the m marble coordinates reaches the origin (i.e., $x = 0$). The median of a sequence of length m (where m is odd) is defined as the element at the position $\frac{m+1}{2}$ when sorted in ascending order (indexing starts from 1). Since the race has not yet started and the starting points are not yet determined, you are interested in the expected value of this time. To avoid floating-point errors, you only need to output the result modulo 10^9+7 (see the output format for details).

Input

The first line contains two positive integers n and m ($1 \le n, m \le 500$, and m is odd), representing the number of starting points and the number of marbles.

The second line contains n integers $x_1, x_2, \ldots, x_n$ ($-10^9 \le x_i < 0$), representing the coordinates of each starting point. It is guaranteed that all x_i are distinct.

The third line contains m integers $v_1, v_2, \ldots, v_m$ ($1 \le v_i \le 10^9$), representing the speed of each marble.

Output

Output a single integer, representing the expected time modulo $10^9 + 7$.

Formally, let $M = 10^9 + 7$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \pmod{M}$, where q^{-1} denotes the modular multiplicative inverse of q modulo M. In other words, output such an integer x that $0 \le x < M$ and $x \cdot q \equiv p \pmod{M}$. It can be proved that there is exactly one x which meets the condition.

Examples

standard input	standard output

2 3
-4 -5
1 2 3
	250000004

3 3
-4 -5 -6
1 2 3
	500000006

5 5
-4 -5 -6 -10 -2
1 2 3 2 4
	434986672

Note

For the first example, the speeds of the three marbles are $1, 2, 3$, respectively. Consider the initial positions of the three marbles:

- $-4, -4, -4$: At $t = 2$, the coordinates of the three marbles are $-2, 0, 2$, and the median is at the origin.

- $-4, -4, -5$: At $t = 2$, the coordinates are $-2, 0, 1$, and the median is at the origin.
- $-4, -5, -4$: At $t = 2.5$, the coordinates are $-1.5, 0, 3.5$, and the median is at the origin.
- For $(-4, -5, -5)$, $(-5, -4, -4)$, $(-5, -4, -5)$, $(-5, -5, -4)$, $(-5, -5, -5)$, the median is at the origin at times $t = 2.5$, $t = 2$, $t = 2$, $t = 2.5$, $t = 2.5$, respectively.

In summary, the expected time is $\frac{2+2+2.5+2.5+2+2+2.5+2.5}{8} = \frac{9}{4}$, so you need to output $9 \cdot 4^{-1} \bmod (10^9 + 7) = 250000004$.

Problem F. 1D Galaxy

In a magical one-dimensional space, there are n planets numbered from 1 to n. Initially ($t = 0$), the planet numbered i is located at position x_i with weight w_i (which can be negative). In the real world, planets move under the influence of gravity, and similarly, in this one-dimensional galaxy, planets move due to attraction. However, the movement in this galaxy does not follow conventional physical laws. Specifically, for any planet in this one-dimensional galaxy, if the total weight of planets to its left is greater than the total weight of planets to its right, it moves one unit to the left at the next time step. Conversely, if the total weight on the right is greater than that on the left, it moves one unit to the right. If both sides are equal, it remains in the same position. You can assume that the planets do not physically collide, meaning they can pass through each other.

Formally, let the position of the planet numbered i at time t ($t = 0, 1, 2, \ldots$) be $x_{i,t}$. The total weight of the planets to its left at this time is $w^l_{i,t} = \sum_{j\,:\,x_{j,t}<x_{i,t}} w_j$, and the total weight of the planets to its right is $w^r_{i,t} = \sum_{j\,:\,x_{j,t}>x_{i,t}} w_j$. The position of the planet at the next time step, $x_{i,t+1}$, is given by:

$$x_{i,t+1} = \begin{cases} x_{i,t} - 1, & w^l_{i,t} > w^r_{i,t} \\ x_{i,t} + 1, & w^l_{i,t} < w^r_{i,t} \\ x_{i,t}, & w^l_{i,t} = w^r_{i,t} \end{cases}$$

There are q queries, each asking for the position of the planet numbered i at a specific time t. Please answer these queries.

Input

The first line contains two integers n and q ($1 \le n, q \le 10^5$), representing the number of planets and the number of queries, respectively.

The i-th of the next n lines contains two integers x_i, w_i ($-10^9 \le x_i, w_i \le 10^9$), representing the initial position and weight of the planet numbered i.

The following q lines each contain two integers t and i ($0 \le t \le 10^9$, $1 \le i \le n$), representing a query.

Output

Output q lines, each representing the answer to the corresponding query.

Example

standard input	standard output

4 12
0 1
1 2
-1 3
2 2
0 1
0 2
0 3
0 4
1 1
1 2
1 3
1 4
2 1
2 2
2 3
2 4
	0
	1
	-1
	2
	1
	0
	0
	1
	0
	1
	1
	0

Problem G. Welcome to Join the Online Meeting!

You want to organize an online meeting on MeLink with n participants numbered form 1 to n. Each of these n participants knows at least one other participant besides themselves, and the acquaintance relationship is mutual.

The organization process of the meeting is as follows: First, one person creates the meeting and joins it. Then, members who have already joined the meeting can invite some of their acquaintances who are not yet in the meeting, until all n participants are present. However, there are k participants who are currently busy debugging code; these people can be invited to the meeting but cannot create the meeting or invite others.

You want to determine if it is possible to get all n participants into the meeting. If it is possible, determine an inviting plan.

Input

The first line contains three integers n, m, k ($2 \le n \le 2 \times 10^5$, $1 \le m \le \min\{5 \times 10^5, \frac{n(n-1)}{2}\}$, $0 \le k \le n$), representing the number of participants, the number of acquaintance relationships, and the number of participants currently busy.

The second line contains k integers $a_1, \ldots, a_k$ ($1 \le a_i \le n$), where the i-th integer represents that participant a_i is busy. These integers are all distinct. If $k = 0$, this line will be empty, but not omitted.

The next m lines each contain two integers p_i and q_i ($1 \le p_i, q_i \le n$, $p_i \ne q_i$), indicating that p_i and q_i know each other. The acquaintance relationships are mutual. It is guaranteed that the same acquaintance relationship will not appear more than once, and that every participant knows at least one other person.

Output

If it is impossible to organize a meeting with all n participants, output `No` in the first line.

If it is possible, output `Yes` in the first line. Then, in the second line, output an integer t ($1 \le t \le n$), representing the number of steps required to organize the meeting.

In the following t lines, each line describes one step of organizing the meeting. In the j-th line, first output an integer x_j ($1 \le x_j \le n$). If $j = 1$, x_j represents the participant who creates the meeting; otherwise, x_j must be a participant who has already joined the meeting. All x_j must be distinct. Next, output an integer y_j ($1 \le y_j \le n$), representing the number of participants invited by x_j in this step. Finally, output y_j integers z_l ($1 \le z_l \le n$), representing the participants invited by x_j. All z_l must be distinct, and no participant can be invited more than once during the entire process.

You do not need to minimize t; any valid plan is acceptable.

Examples

standard input	standard output

`4 5 2`
`3 4`
`1 2`
`1 3`
`2 3`
`3 4`
`2 4`
	`Yes`
	`2`
	`1 2 2 3`
	`2 1 4`

`4 5 3`
`2 4 3`
`1 2`
`1 3`
`2 3`
`3 4`
`2 4`
	`No`

Problem H. Subsequence Counting

Given a sequence $\{t\}$ of length m and a sequence $\{s\}$ of length L, where $\{s\}$ is composed of n consecutive segments from left to right. The i-th segment contains l_i identical elements, each with a value of v_i.

The sequence $\{s'\}$ is formed by shuffling the sequence $\{s\}$ according to a certain rule. Specifically, the sequence $\{s'\}$ satisfies $s'_{i \cdot k \bmod L} = s_i$ (indices start from 0). Here, k is a given positive integer constant, and it is guaranteed that $\gcd(k, L) = 1$.

Find the number of times $\{t\}$ appears as a subsequence in $\{s'\}$. Formally, if there is a strictly increasing sequence of indices $0 \le i_1 < i_2 < \cdots < i_m < L$ such that for each $j = 1, 2, \ldots, m$, $t_j = s'_{i_j}$, then $\{t\}$ is considered a subsequence of $\{s'\}$ at these indices. You need to determine how many different index groups satisfy this condition. Since the answer may be large, output the result modulo 998244353.

Input

The first line contains four integers n, m, k, L ($1 \le n \le 2 \times 10^3$, $1 \le m \le 10$, $1 \le k < L \le 10^9$, $\gcd(k, L) = 1$).

The second line contains m integers representing the sequence $\{t\}$ ($1 \le t_i \le 10^3$).

The next n lines describe the sequence $\{s\}$, each containing two integers l_i, v_i ($1 \le l_i \le 10^9$, $1 \le v_i \le 10^3$). It is guaranteed that $\sum_{i=1}^{n} l_i = L$.

Output

Output a single integer, representing the result modulo 998244353.

Examples

standard input	standard output

4 2 17 27
3 1
10 3
6 1
10 3
1 1
	76

5 3 1789 15150
555 718 726
72 555
1029 718
5807 726
1002 718
7240 555
	390415327

Problem I. A Brand New Geometric Problem

You are a magician in a high-dimensional space, and you have an initial n-dimensional hypercube with edge lengths $a_1, a_2, \ldots, a_n$. For a d-dimensional hypercube, the edge length sum is defined as $\sum_{i=1}^{d} a_i$, and its hypervolume is $\prod_{i=1}^{d} a_i$.

You want to obtain a hypercube with edge length sum S and hypervolume M. To achieve this, you can perform both dimensional reduction and dimensional expansion operations on the current hypercube.

- Dimensional Reduction: Remove a dimension.
- Dimensional Expansion: Add a new dimension, with its edge length being any positive integer.

Both operations are very exhausting, so you want to determine the minimum number of operations required to obtain a hypercube with edge length sum S and hypervolume M.

Input

The first line contains three integers n, S, M ($1 \le n \le 10^5$, $1 \le S, M \le 10^{10}$).

The second line contains n integers, representing the initial edge lengths a_i of the hypercube ($1 \le a_i \le 10^{10}$).

Output

Output a single integer representing the minimum number of operations required. If it is impossible to obtain a hypercube that meets the conditions, output -1.

Examples

standard input	standard output
2 5 6 1 2	2
3 6 5 1 2 3	3
2 114514 735134400 114 514	20
2 4 7 1 3	-1

Note

For the first sample, one possible approach: first delete the dimension with edge length 1, and then add a dimension with edge length 3.

Problem J. New Energy Vehicle

A new energy vehicle is equipped with n batteries, where the i-th battery has a capacity of a_i units. Each unit of electricity allows the vehicle to travel exactly 1 kilometer. The vehicle can only go forward, not in reverse. You can choose which battery to use for each kilometer driven.

Initially, all batteries are fully charged. During the journey, the vehicle will pass through m charging stations. The j-th charging station is located at x_j kilometers from the starting point and can only recharge the t_j-th battery. Each charging station provides an unlimited amount of electricity.

Your task is to determine the maximum distance the new energy vehicle can travel.

Input

The first line contains an integer T ($1 \le T \le 10^4$), representing the number of test cases.

For each test case, the first line contains two integers n, m ($1 \le n, m \le 10^5$), representing the number of batteries and the number of charging stations, respectively.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$), representing the capacity of each battery.

The next m lines each contain two integers x_j, t_j ($1 \le x_j \le 10^9$, $1 \le t_j \le n$), representing the position of each charging station and the battery it can recharge.

For each test case, it is guaranteed that $1 \le x_1 < x_2 < \ldots < x_m \le 10^9$. Either the sum of n or the sum of m over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output an integer in a single line, representing the maximum distance the vehicle can travel.

Example

standard input	standard output

2
3 1
3 3 3
8 1
2 2
5 2
1 2
2 1
	12
	9

Problem K. Farm Management

You have given up programming and moved to the Sanjiang Plain to start farming. During your time working in the fields, you have adopted a regular daily schedule, and now you work **exactly** m units of time each day. It is now harvest season, and you need to harvest and process n types of crops. For crop type i, processing it for one unit of time will yield a profit of w_i. To make your daily work less monotonous, for each crop type i, the time spent processing it each day can range between $[l_i, r_i]$ inclusive as an integer.

At some day, the weather forecast says that there will be a heavy rain tomorrow and you can't work, so you need to adjust your schedule to quickly gather your crops today. Specifically, you can choose at most one type of crop and remove its daily time range restriction, allowing the time spent processing this crop to be any integer in the range $[0, m]$. The time ranges for all other crops remain unchanged. You have to work **exactly** m units of time as well.

You want to determine the maximum profit you can earn today.

Input

The first line contains two integers n and m ($1 \le n \le 10^5$, $1 \le m \le 10^{11}$), representing the number of crop types and the length of the workday in units of time, respectively.

The next n lines each contain three integers w_i, l_i, and r_i ($1 \le w_i \le 10^6$, $1 \le l_i \le r_i \le 10^6$), indicating the profits and time constraints of the crops.

It is guaranteed that $\sum_{i=1}^{n} l_i \le m \le \sum_{i=1}^{n} r_i$.

Output

Output a single integer representing the maximum profit you can earn today.

Example

standard input	standard output

5 17
2 3 4
6 1 5
8 2 4
4 3 3
7 5 5
	109

Problem L. A Game On Tree

There is a tree (a connected undirected graph with n nodes and $n-1$ edges) consisting of n nodes, with nodes numbered from 1 to n. Clearly, there is a unique simple path between any two nodes in the tree.

Xiaohong and Xiaolan are playing a game on this tree. In each game, both players **independently and uniformly** select a random simple path from all $\frac{n(n-1)}{2}$ simple paths (regardless of direction) that exist in the tree. Note that they may choose the same path. Let X denote the number of edges that are common to both selected paths, and the score of the game is X^2.

Your task is to find the expected value of the score $E(X^2)$ when Xiaohong and Xiaolan play the game once, and output the result modulo 998244353 (see the output format for details).

Input

The first line contains a positive integer T $(1 \le T \le 10^4)$, representing the number of test cases.

For each test case, the first line contains a positive integer n $(2 \le n \le 10^5)$, representing the number of nodes in the tree.

The next $n-1$ lines each contain two positive integers u, v $(1 \le u, v \le n)$, indicating that there is an edge between nodes u and v. The input is guaranteed to be a tree.

The sum of all n over all test cases does not exceed 10^6.

Output

For each test case, output a single integer, representing the answer modulo 998244353.

Formally, let $M = 998244353$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \pmod{M}$, where q^{-1} denotes the modular multiplicative inverse of q modulo M. In other words, output such an integer x that $0 \le x < M$ and $x \cdot q \equiv p \pmod{M}$. It can be proved that there is exactly one x which meets the condition.

Example

standard input	standard output

2
3
1 2
2 3
5
1 2
1 5
3 2
4 2
	443664158
	918384806

Note

For the first test case in the example, the answer without taking the modulo is $\frac{10}{9}$.

Among the 9 possible cases:

- In 2 cases, the number of common edges between the two paths is 0;
- In 6 cases, the number of common edges between the two paths is 1;
- In 1 case, the number of common edges between the two paths is 2.

Therefore, the answer is $E(X^2) = \frac{2 \cdot 0^2 + 6 \cdot 1^2 + 1 \cdot 2^2}{9} = \frac{10}{9}$.

Problem M. Weird Ceiling

While learning about the ceiling function, a student wrote the following pseudocode:

```
function F(a, b)
    $i \leftarrow b$
    while $i \geq 2$ do
        if $a \bmod i = 0$ then
            return $\frac{a}{i}$
        end if
        $i \leftarrow i - 1$
    end while
    return $a$
end function
```

You know that this is incorrect, but you are curious about the characteristics of the function $f(a, b)$ defined by this student. Specifically, you want to calculate the value of $\sum_{i=1}^{n} f(n, i)$.

Input

The first line contains an integer T $(1 \leq T \leq 10^3)$, indicating the number of test cases.

For each test case, there is one line containing an integer n $(1 \leq n \leq 10^9)$.

Output

For each test case, output one line containing an integer representing the answer.

Example

standard input	standard output
3 5 451 114514	21 10251 7075858