

Problem A. Tree Game

Input file: **standard input**
Output file: **standard output**
Balloon Color: **Pink**

Zagha and Hiasat are playing a game on a tree with n nodes. At the beginning, both players are assigned a starting node on the tree, the two starting nodes are different, then both players take turns alternately. In each turn, the player chooses an **unassigned** adjacent node and merges it with his current node.

To merge node a with node b , we remove the edge (a, b) , remove the node b , and then any edge (b, x) for any node x becomes (a, x) .

Zagha will start first, and the player who can't make a move loses. In how many ways we can choose the starting positions for Zagha and Hiasat such that Zagha is guaranteed to win (if both play optimally)?

Input

The first line of input contains a single integer T ($1 \leq T \leq 15000$), the number of test cases.

The first line of each test case contains a single integer n ($1 \leq n \leq 10^6$), the size of the tree.

The second line contains $n - 1$ space-separated integers $a_2, a_3, \dots, a_n$ ($1 \leq a_i \leq n$), representing that there is an edge between node i and a_i .

The sum of n over all test cases doesn't exceed 4×10^6 .

Output

For each test case, output a single line with the number of ways to choose the starting positions for Zagha and Hiasat such that Zagha is guaranteed to win.

Example

standard input	standard output
3	0
2	4
1	13
3	
1 2	
5	
5 1 3 3	

Note

In the second sample, there are 6 ways to choose the starting positions. Zagha is guaranteed to win in 4 of them.

Problem B. Rectangles

Input file: standard input
Output file: standard output
Balloon Color: Light Green

You are given $2n$ points, you need to pair them to create n rectangles so that the intersection of all those rectangles forms a positive area. In how many ways you can do that? Find the number of ways modulo $10^9 + 7$.

Pairing two points $(x1, y1)$ and $(x2, y2)$ forms the rectangle with bottom-left corner $(\min(x1, x2), \min(y1, y2))$ and top-right corner $(\max(x1, x2), \max(y1, y2))$.

Input

The first line of input contains a single integer T ($1 \leq T \leq 3200$), the number of test cases.

The first line of each test case contains a single integer n ($1 \leq n \leq 10^5$), the number of rectangles you need to make.

The following $2n$ lines each contains two space-separated integers x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$), representing the coordinates of the i^{th} point. All points are distinct.

The sum of n over all test cases doesn't exceed 5×10^5 .

Output

For each test case output a single line with the number of ways to pair the $2n$ points modulo $10^9 + 7$.

Example

standard input	standard output
2	1
2	2
0 0	
0 1	
1 0	
1 1	
2	
0 1	
1 0	
2 3	
3 2	

Problem C. Function

Input file: **standard input**
Output file: **standard output**
Balloon Color: **Orange**

Usually in competitive programming you're given a problem in which you need to design an efficient algorithm that solves that problem and implement it, but for this problem we decided to take it easy on you and just give you the algorithm, so that your only task is to implement it.

The algorithm goes like this:

```
f(l,r):  
    if l is equal to r then return a[l]  
    s = 0  
    for i = l to r: s = s + a[i]  
    return s + f(l+1,r) + f(l,r-1)
```

You are given the elements a_i , print the value of $f(1, n)$ modulo $10^9 + 7$.

Input

The first line of input contains a single integer T ($1 \leq T \leq 16$), the number of test cases.

The first line of each test case contains a single integer n ($1 \leq n \leq 10^6$), the number of elements of array a .

The second line contains n space-separated integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$).

The sum of n over all test cases doesn't exceed 4×10^6 .

Output

For each test case, output a single line with the value of the function $f(1, n)$ modulo $10^9 + 7$.

Example

standard input	standard output
3	4
2	22
1 1	295
3	
1 2 3	
5	
1 3 5 7 11	

Problem D. Two Sequences

Input file: standard input
Output file: standard output
Balloon Color: Red

You are given two sequences of integers A and B , and an integer k .

Two sequences are equal if it is possible to rearrange one of the sequences such that each element in the sequence is equal to the corresponding element from the other sequence.

You can select one of the elements in sequence A , and add to it any integer from the range $[-k, k]$. Is it possible to make the two sequences equal by doing the specified operation **exactly** once?

Input

The first line of input contains a single integer T , the number of test cases.

The first line of each case contains two space-separated integers n and k ($1 \leq n \leq 10^5$)($1 \leq k \leq 10^9$), the length of each sequence and the number k specified in the problem statement.

The next line contains n space-separated integers ($1 \leq A_i \leq 10^9$), the elements in sequence A .

The next line contains n space-separated integers ($1 \leq B_i \leq 10^9$), the elements in sequence B .

Output

For each test case, output a single line with **YES** if it is possible to make the two sequences equal in a single operation. Otherwise output **NO**.

Example

standard input	standard output
3	YES
3 2	NO
1 2 3	YES
3 2 1	
4 2	
1 9 1 1	
6 1 1 1	
2 4	
1 5	
1 1	

Problem E. Connecting Components

Input file: standard input
Output file: standard output
Balloon Color: Yellow

You are given an undirected graph with n nodes and $n - 1$ edges. The graph doesn't contain loops or multiple edges.

You are allowed to make the following operation **at most twice**:

- Choose an edge (u, v) from the graph, change it to (u, w) , the cost of this operation is $|w - v|$, where $|x|$ is the absolute value of x .

In other words, you are allowed to change one endpoint of an edge in one move, and the cost of this move is the absolute difference between the new ID and the old ID of the changed endpoint.

Find the minimum total cost needed to make the given graph connected in **at most two** moves, or determine that it is impossible.

Note that you do **not** have to minimize the number of moves.

Input

The first line of input contains a single integer T ($1 \leq T \leq 10^4$), the number of test cases.

The first line of each test case contains a single integer n ($1 \leq n \leq 10^5$), the number of nodes in the graph.

Each of the following $n - 1$ lines contains two space-separated integers u and v ($1 \leq u, v \leq n, u \neq v$), representing an edge that connects the nodes u and v . Each pair of nodes will be connected by at most one edge.

The sum of n over all test cases doesn't exceed 5×10^6 .

Output

For each test case, if it is impossible to make the given graph connected in **at most two** moves, output -1 . Otherwise, output the minimum total cost needed, in a single line.

Example

standard input	standard output
3	2
8	0
5 1	-1
1 3	
3 5	
4 2	
2 7	
7 6	
6 2	
2	
1 2	
7	
1 2	
1 3	
1 4	
2 3	
2 4	
3 4	

Problem F. Mirror

Input file: **standard input**
Output file: **standard output**
Balloon Color: **White**

There are n students in a classroom, this classroom can be seen as a Cuboid in which the students are looking in the direction of the negative z -axis. The positive x -axis is to their right and the positive y -axis points upward. The wall in front of the students is the plane $z = 0$, and the wall behind them is the plane $z = a$. The walls to their right and left are the planes $x = 10^9$ and $x = -10^9$, respectively.

You will be given q queries of the form (x_i, y_i, a) , meaning that there's a point on the wall behind the students at those coordinates. You need to put a rectangular mirror with minimum area on the front wall so that all n students can see that point through the mirror (The mirror needs to be **axially aligned**). Find that area for each query independently.

Input

The first line of input contains a single integer T ($1 \leq T \leq 8000$), the number of test cases.

The first line of each test case contains two space-separated integers n and a ($1 \leq n \leq 2 \times 10^5$) ($2 \leq a \leq 10^9$), where n is the number of students, and a represents that the wall behind the students is located at $z = a$.

The next n lines each contains two space-separated integers x_i and z_i ($-10^9 < x_i < 10^9$) ($0 < z_i < a$), representing that the location of the i^{th} student is $(x_i, 0, z_i)$. All locations are distinct.

The next line contains an integer q ($1 \leq q \leq 2 \times 10^5$), the number of queries.

The next q lines each contains two space-separated integers x_i and y_i ($-10^9 < x_i < 10^9$) ($1 \leq y_i \leq 10^9$), representing that the location of the point in the i^{th} query is (x_i, y_i, a) .

The sum of n over all test cases doesn't exceed 1.5×10^6 . Same goes for q .

Output

For each query, print a single line with the minimum area of the mirror needed. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6} .

Example

standard input	standard output
1	4.5000000000
3 3	3.2400000000
-2 1	10.3500000000
7 2	
3 1	
3	
2 5	
-2 4	
8 10	

Problem G. TeddyBearsDay

Input file: **standard input**
Output file: **standard output**
Balloon Color: **Black**

You are given the description of TeddyLand, an undirected tree of n nodes where each node represent a TeddyCity, and there's a TeddyPerson that lives in each of the TeddyCities.

As everyone knows, since the TeddyBearsDay is approaching, people from all around TeddyLand are planning to send TeddyBears to each other.

But we all know that the TeddyDelivery is very expensive these days (the TeddyCost to send a TeddyBear from a TeddyCity to another TeddyCity is the length of the TeddyPath between them!).

So you, the TeddyKing, will rearrange the TeddyDeliveries in such a way that each TeddySender will still send the same amount of TeddyBears, and each TeddyReceiver will still receive the same amount of TeddyBears, and the total TeddyCost is minimum.

Input

The first line will contain one integer T ($1 \leq T \leq 250$) - the number of TeddyTestCases.

The first line of each TeddyCase will contain one integer n ($1 \leq n \leq 10000$) - the number of TeddyCities in TeddyLand.

The following $n - 1$ lines will contain three integers each u_i , v_i and w_i ($1 \leq u_i, v_i \leq n$) ($1 \leq w_i \leq 10000$) - meaning that TeddyCity u_i is connected to TeddyCity v_i with a TeddyRoad of length w_i .

The next line will contain one integer q ($1 \leq q \leq 10000$) - the number of TeddyDeliveries planned.

The following q lines will contain two integers each u_i , v_i ($1 \leq u_i, v_i \leq n$) - meaning that the TeddyPerson in TeddyCity u_i wants to send a TeddyBear for the TeddyPerson that lives in the TeddyCity v_i .

It's possible that u_i is equal to v_i .

The total sum of $n \leq 5 * 10^5$

The total sum of $q \leq 5 * 10^5$

Output

For each test case, output a single line with the minimum total TeddyCost of all TeddyDeliveries after rearrangement.

Example

standard input	standard output
1 7 1 2 2 2 3 1 2 4 1 1 5 1 5 6 2 5 7 2 2 6 3 4 5	4

Problem H. Win Strategy

Input file: standard input
Output file: standard output
Balloon Color: Dark Blue

You are at a programming contest where there will be n problems. For each problem you know at what time it will be available for you to read (and solve). Also you know how much time you need to solve it.

To solve a problem, you need to spend the needed time continuously without switching between different problems.

Find the maximum number of problems you can solve in the given contest time.

Input

The first line of input contains a single integer T ($1 \leq T \leq 250$), the number of test cases.

The first line of each test case contains two space-separated integers n and L ($1 \leq n, L \leq 1000$), the number of problems and the length of the contest, respectively.

Each of the following n lines contains two space-separated integers a_i and b_i ($1 \leq a_i, b_i \leq L$), which means the i^{th} problem will be available at minute a_i and you need b_i minutes to solve it.

Problems will be given in a non-decreasing order of a_i .

Output

For each test case, output a single line with the maximum number of problems you can solve during the contest.

Example

standard input	standard output
1 4 10 1 3 1 2 1 6 2 2	3

Problem I. Tours

Input file: **standard input**
Output file: **standard output**
Balloon Color: **Dark Purple**

You own a company that organizes tours to Mount Fuji everyday.

Each day, a number of buses stationed at different cities pick up people in the morning and get them back to the station in the evening after the tour has ended.

People have already booked their tickets for the next n days, so you want to plan the number of buses that you want to put at each station such that there's always enough buses for the people attending the tour from each station, and the total number of buses used is minimum possible.

Note that you can make any bus go from one station to another between multiple days, but doing so requires a whole day for the bus to relocate to the other station, so you can't use it during that day in any station.

For example, if a bus was used at station x at day k , and we need it to relocate to station y , then it will only arrive and be usable there on day $k + 2$.

Input

The first line of input contains a single integer T ($1 \leq T \leq 250$), the number of test cases.

The first line of each test case contains three space-separated integers m , n , and R ($1 \leq m \times n \leq 10^5$) ($1 \leq R \leq 10^5$), the number of stations, the number of days, and the capacity of a single bus, respectively.

Each of the following m lines contains n integers, where $a_{i,j}$ ($1 \leq a_{i,j} \leq 10^5$) represents the number of people who booked a ticket to get in the j^{th} tour from the i^{th} station.

The total sum of $A_{i,j} \leq 5 * 10^5$

Output

For each test case, output a single line with the minimum number of buses needed to satisfy all the tours.

Example

standard input	standard output
2	2
2 3 1	8
1 0 2	
1 0 0	
1 5 1	
1 2 8 1 6	

Problem J. Restricted Vertex Cover

Input file: standard input
Output file: standard output
Balloon Color: Gold

You are given an undirected graph G consisting of n nodes and m edges, each edge is either marked or unmarked.

Your job is to find out if it's possible to find **any** Vertex Cover of the graph, such that the marked edges can have only one of their endpoints included in the set (but not both).

A Vertex Cover is a subset of the nodes such that each edge in the graph has at least one of its endpoints in this set.

Input

The first line of input contains a single integer T ($1 \leq T \leq 40$), the number of test cases.

The first line of each test case contains two space-separated integers n and m ($1 \leq n \leq 10^5$) ($0 \leq m \leq 10^5$), the number of nodes and edges in the graph, respectively.

Each of the following m lines contains three space-separated integers u_i , v_i and w_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$) ($0 \leq w_i \leq 1$), representing that the i^{th} edge connects the nodes u_i and v_i , and the edge is marked if w_i is equal to 1, otherwise it's unmarked.

Output

For each test case, output a single line with **YES** if it's possible to find such a Vertex Cover, otherwise output **NO**.

Example

standard input	standard output
2	YES
5 7	NO
1 2 1	
2 3 1	
3 4 0	
1 3 0	
2 4 1	
5 3 1	
5 4 1	
5 6	
1 2 1	
1 3 1	
2 4 1	
4 5 1	
3 5 1	
2 5 0	