Problem A. Bookshop

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

DreamGrid will go to the bookshop tomorrow. There are n books in the bookshop in total, labeled by $1, 2, \ldots, n$, connected by $n-1$ bidirectional roads like a tree. Because DreamGrid is very rich, he will buy the books according to the strategy below:

- Select two books x and y.
- Walk from x to y along the shortest path on the tree, and check each book one by one, including the x-th book and the y-th book.
- For each book being checked now, if DreamGrid has enough money (not less than the book price), he'll buy the book and his money will be reduced by the price of the book. In case that his money is less than the price of the book being checked now, he will skip that book.

BaoBao is curious about how rich DreamGrid is. You will be given q queries, in each query you will be given three integers x_i, y_i and z_i. Assume DreamGrid will bring z_i dollars before buying books, and will walk from x_i to y_i, you are asked to tell BaoBao the amount of money DreamGrid will have after he checking all the visited books.

Input

The first line contains a single integer T ($1 \le T \le 500$), the number of test cases. For each test case:

The first line of the input contains two integers n and q ($1 \le n, q \le 100\,000$), denoting the number of books in the bookshop and the number of queries.

The second line contains n integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le 10^9$), denoting the price of each book.

Each of the next $n-1$ lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$), denoting a bidirectional road between u_i and v_i. It is guaranteed that the roads form a tree.

In the next q lines, the i-th line contains three integers x_i, y_i and z_i ($1 \le x_i, y_i \le n$, $1 \le z_i \le 10^9$), describing the i-th query.

It is guaranteed that the sum of all n is at most $800\,000$, and the sum of all q is at most $800\,000$.

Output

For each query, print a single line containing an integer, denoting the amount of money DreamGrid will have after he checking all the visited books.

Example

standard input	standard output
1 5 5 5 2 7 4 6 1 2 1 3 2 4 2 5 4 5 10 5 4 10 4 5 12 5 3 12 3 5 11	4 2 0 4 2

Problem B. Destinations

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

There are n towns in Byteland, labeled by $1, 2, \ldots, n$, connected by $n-1$ bidirectional roads like a tree. m tourists are going to visit Byteland, the i-th of which will start his journey at the s_i-th town. However, none of them has decided where to end his journey. Formally, for the i-th tourist, he has made 3 plans, the j-th of which will end at the $e_{i,j}$-th town, and will cost him $c_{i,j}$ dollars. When a tourist makes his decision, he will walk along the shortest path to visit all the towns from s_i to his destination. Note that $e_{i,j}$ can coincide with s_i, the s_i-th town and the destination town will also be visited by the i-th tourist. Two plans may share the same destination but may differ in cost because the tourist may do extremely different things in the same town.

They will share photos of their journeys in Byteland. Nobody is willing to visit the same place with others. Your task is to help each tourist choose his destination from his plans such that each town will be visited by at most one tourist, and the total cost for all the tourists is minimized, or determine it is impossible.

Input

The first line contains a single integer T ($1 \le T \le 500$), the number of test cases. For each test case:

The first line of the input contains two integers n and m ($1 \le n \le 200\,000$, $1 \le m \le 100\,000$), denoting the number of towns and the number of tourists.

Each of the next $n-1$ lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$), denoting a bidirectional road between u_i and v_i. It is guaranteed that the roads form a tree.

In the next m lines, the i-th line contains seven integers s_i, $e_{i,1}$, $c_{i,1}$, $e_{i,2}$, $c_{i,2}$, $e_{i,3}$ and $c_{i,3}$ ($1 \le s_i, e_{i,j} \le n$, $1 \le c_{i,j} \le 10^6$), describing the i-th tourist.

It is guaranteed that the sum of all n is at most $1\,000\,000$, and the sum of all m is at most $300\,000$.

Output

For each test case, print a single line containing an integer denoting the minimum total cost. If there is no solution, print "`-1`" instead.

Example

standard input	standard output
2 7 2 1 2 1 3 2 4 2 5 3 6 3 7 2 1 1 3 100 4 200 3 2 1 4 2 7 50 4 2 1 2 2 3 3 4 1 2 1 3 1 4 1 2 1 1 3 1 4 1	51 -1

Problem C. Forgiving Matching

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

Little Q is now checking whether string A matches B. Two strings are considered matched if they have the same length, and there is no position i that A_i is different from B_i.

However, Little Q is a kind man, he forgives every person who hurt him. What's more, he even forgives strings! He gives the string k opportunities, if there are no more than k positions i that A_i is different from B_i, then Little Q will also consider the two strings matched. Note that both of the strings may contain the wildcard character '`*`', which can match exactly one any character, in such a case this pair won't consume the forgiveness opportunities.

Let's denote $occ(S,T)$ as the number of substrings in string S which matches T, two substrings are considered different if they start in different places. You will be given two strings S and T, write a program to compute the value of $occ(S,T)$ for $k = 0, 1, 2, \ldots, |T|$.

Input

The first line contains a single integer K $(1 \leq K \leq 100)$, the number of test cases. For each test case:

The first line of the input contains two integers n and m $(1 \leq m \leq n \leq 200\,000)$, denoting the length of S and the length of T.

The second line contains a string S of length n.

The third line contains a string T of length m.

It is guaranteed that the sum of all n is at most $1\,000\,000$. Both S and T can only contain the characters in {'`0`', '`1`', '`2`', '`3`', '`4`', '`5`', '`6`', '`7`', '`8`', '`9`', '`*`'}.

Output

For each test case, output $m + 1$ lines, the i-th $(1 \leq i \leq m + 1)$ of which containing an integer, denoting the value of $occ(S,T)$ when $k = i - 1$.

Example

standard input	standard output

`1`
`5 3`
`012*4`
`1*3`
	`1`
	`1`
	`3`
	`3`

Problem D. Game on Plane

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

Alice and Bob are playing a game. In this game, there are n straight lines on the 2D plane. Alice will select exactly k straight lines $l_1, l_2, \ldots, l_k$ among all the n straight lines first, then Bob will draw a straight line L. The penalty of Bob is defined as the number of lines in $\{l_1, l_2, \ldots, l_k\}$ that shares at least one common point with L. Note that two overlapping lines also share common points.

Alice wants to maximize the penalty of Bob while Bob wants to minimize it. You will be given these n lines, please write a program to predict the penalty of Bob for $k = 1, 2, 3, \ldots, n$ if both of the players play optimally.

Input

The first line contains a single integer T ($1 \le T \le 500$), the number of test cases. For each test case:

The first line contains a single integer n ($1 \le n \le 100\,000$), denoting the number of straight lines.

Each of the next n lines contains four integers xa_i, ya_i, xb_i and yb_i ($0 \le xa_i, ya_i, xb_i, yb_i \le 10^9$), denoting a straight line passes both (xa_i, ya_i) and (xb_i, yb_i). (xa_i, ya_i) will never be coincided with (xb_i, yb_i).

It is guaranteed that the sum of all n is at most $1\,000\,000$.

Output

For each test case, output n lines, the i-th ($1 \le i \le n$) of which containing an integer, denoting the penalty of Bob when $k = i$.

Example

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

Problem E. Kart Race

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

The annual kart race will soon be held in Byteland. The map of the race consists of n different intersections and m one-way streets. These intersections are labeled from 1 to n, the i-th of which is located at (x_i, y_i). There are no cycles in the street network, one can only drive to some intersections with the strictly larger value of x-coordinate. The streets may only intersect at the intersections.

The race will start at the 1-st intersection and will finish at the n-th intersection. The racers can pick their routes themselves, but they can only drive along the streets marked on the map. It is guaranteed that one can reach any place from 1, and any place can reach n, so any route is valid.

The kart race attracts so many sponsors. Each intersection has a slot to set a banner, if you choose to set a banner at the i-th intersection, the race company will get w_i profits. You are a middleman in the race company, your job is to choose some intersections to set banners such that the total profits are maximized. You know that no racer is willing to see more than a banner, so for every possible route from 1 to n, you should guarantee that at most one intersection is chosen.

Input

The first line contains a single integer T $(1 \le T \le 1\,000)$, the number of test cases. For each test case:

The first line of the input contains two integers n and m $(1 \le n \le 100\,000,\ 1 \le m \le 2n)$, denoting the number of intersections and the number of one-way streets.

In the next n lines, the i-th line contains three integers x_i, y_i and w_i $(0 \le x_i, y_i \le 10^9,\ 1 \le w_i \le 10^9)$, describing the i-th intersection. It is guaranteed that no two intersections share the same coordinator.

Each of the next m lines contains two integers u_i and v_i $(1 \le u_i, v_i \le n,\ x_{u_i} < x_{v_i})$, denoting a one way street from u_i to v_i. It is guaranteed that each pair of u_i and v_i will be described at most once.

It is guaranteed that the sum of all n is at most $1\,500\,000$.

Output

For each test case, print a single integer in the first line, denoting the maximum total profits. Then print a sequence of integers in the second line, denoting the indexes of intersections you choose to set a banner. If there are multiple optimal solutions, you should print the lexicographically smallest one.

Example

standard input	standard output
2 6 6 0 1 1 2 2 1 1 0 1 1 2 1 2 0 1 3 1 1 1 4 3 5 2 6 5 6 1 3 4 2 2 1 0 0 8 1 1 9 1 2	2 2 3 9 2

Problem F. New Equipments II

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

Little Q's factory recently purchased n pieces of new equipment, labeled by $1, 2, \ldots, n$.

There are n workers in the factory, labeled by $1, 2, \ldots, n$. Each worker can be assigned to no more than one piece of equipment, and no piece of equipment can be assigned to multiple workers. If Little Q assigns the i-th worker to the j-th piece of equipment, they will bring $a_i + b_j$ profits. However, these workers are not so experienced, there are m extra constraints. In each constraint, you will be given two integers u_i and v_i, denoting the u_i-th worker can't be assigned to the v_i-th piece of equipment.

Now please for every k ($1 \leq k \leq n$) find k pairs of workers and pieces of equipment, then assign workers to these pieces of equipment, such that the total profits for these k pairs are maximized, or determine it is impossible.

Input

The first line contains a single integer T ($1 \leq T \leq 200$), the number of test cases. For each test case:

The first line contains two integers n and m ($1 \leq n \leq 4\,000$, $1 \leq m \leq 10\,000$), denoting the number of workers/pieces of new equipment and the number of constraints.

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

The third line contains n integers $b_1, b_2, \ldots, b_n$ ($1 \leq b_i \leq 10^9$).

Each of the following m lines contains two integers u_i and v_i ($1 \leq u_i, v_i \leq n$), denoting the u_i-th worker can't be assigned to the v_i-th piece of equipment. Each pair of u_i and v_i will be described at most once.

It is guaranteed that there will be at most 10 test cases such that $n > 100$.

Output

For each test case, output n lines, the k-th ($1 \leq k \leq n$) of which containing an integer, denoting the maximum possible total profits for k pairs of workers and pieces of equipment. Note that if it is impossible to find such k pairs, print "`-1`" at this line.

Example

standard input	standard output
2 4 4 10 20 30 40 5 10 7 8 4 2 3 2 1 4 1 2 2 2 1 1 1 1 1 1 1 2	48 85 115 130 2 -1

Problem G. Photoshop Layers

Input file: standard input
Output file: standard output
Memory limit: 512 megabytes

Pixels in a digital picture can be represented with three integers (R, G, B) in the range 0 to 255 that indicate the intensity of the red, green, and blue colors. The color of a pixel can be expressed as a six-digit hexadecimal capital string. For example, $(R = 100, G = 255, B = 50)$ can be expressed as "`64FF32`".

There are n layers in Photoshop workstation, labeled by $1, 2, \ldots, n$ from bottom to top. The screen will display these layers from bottom to top. In this problem, you only need to handle the case that the color of all the pixels in a layer are the same. The color of the i-th layer is $c_i = (R_i, G_i, B_i)$, the blending mode of the i-th layer is m_i ($m_i \in \{1, 2\}$):

- If $m_i = 1$, the blending mode of this layer is "Normal". Assume the previous color displayed on the screen is (R_p, G_p, B_p), now the new color will be (R_i, G_i, B_i).
- If $m_i = 2$, the blending mode of this layer is "Linear Dodge". Assume the previous color displayed on the screen is (R_p, G_p, B_p), now the new color will be $(\min(R_p + R_i, 255)$, $\min(G_p + G_i, 255)$, $\min(B_p + B_i, 255))$.

You will be given q queries. In the i-th query, you will be given two integers l_i and r_i ($1 \le l_i \le r_i \le n$). Please write a program to compute the final color displayed on the screen if we only keep all the layers indexed within $[l_i, r_i]$ without changing their order. Note that the color of the background is $(R = 0, G = 0, B = 0)$.

Input

The first line contains a single integer T ($1 \le T \le 10$), the number of test cases. For each test case:

The first line of the input contains two integers n and q ($1 \le n, q \le 100\,000$), denoting the number of layers and the number of queries.

In the next n lines, the i-th line contains an integer m_i and a six-digit hexadecimal capital string c_i, describing the i-th layer.

In the next q lines, the i-th line contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$), describing the i-th query.

Output

For each query, print a single line containing a six-digit hexadecimal capital string, denoting the final displayed color.

Example

standard input	standard output
1 5 5 1 64C832 2 000100 2 010001 1 323C21 2 32C8C8 1 2 1 3 2 3 2 4 2 5	64C932 65C933 010101 323C21 64FFE9

Problem H. Restore Atlantis II

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

There are n ancient Greek maps describing the fabled islands Atlantis. The maps are labeled by $1, 2, \ldots, n$. The i-th map shows the rectangle area R_i is a part of Atlantis. The sides of all rectangles are parallel to the axes. There may be multiple islands, and the rectangles may overlap.

Unfortunately, some maps are even unreliable so they will not be considered. You will be given q queries. In the i-th query, you will be given two integers s_i and t_i ($1 \leq s_i \leq t_i \leq n$). Please write a program to figure out the total area of Atlantis when only maps labeled by k ($s_i \leq k \leq t_i$) are reliable.

Input

The first line contains a single integer T ($1 \leq T \leq 3$), the number of test cases. For each test case:

The first line of the input contains two integers n and q ($1 \leq n, q \leq 100\,000$), denoting the number of maps and the number of queries.

In the next n lines, the i-th line contains four integers xa_i, ya_i, xb_i and yb_i ($0 \leq xa_i < xb_i \leq 10^9$, $0 \leq ya_i < yb_i \leq 10^9$), describing the i-th map R_i. (xa_i, ya_i) is the southwest corner of R_i, and (xb_i, yb_i) is the northeast corner of R_i.

In the next q lines, the i-th line contains two integers s_i and t_i ($1 \leq s_i \leq t_i \leq n$), describing the i-th query.

It is guaranteed that all the values of xa_i, ya_i, xb_i, yb_i, s_i and t_i are chosen uniformly at random from integers in their corresponding ranges. The randomness condition does not apply to the sample test case, but your solution must pass the sample as well.

Output

For each query, print a single line containing an integer, denoting the total area using the information of all reliable maps in this query.

Example

standard input	standard output
1 3 6 10 10 20 20 12 12 22 22 15 15 25 25 1 1 2 2 3 3 1 2 2 3 1 3	100 100 100 136 151 187

Problem I. Rise in Price

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

There are $n \times n$ cells on a grid, the top-left cell is at $(1, 1)$ while the bottom-right cell is at (n, n). You start at $(1, 1)$ and move to (n, n). At any cell (i, j), you can move to $(i + 1, j)$ or $(i, j + 1)$, provided that you don't move out of the grid. Clearly, you will make exactly $2n - 2$ steps.

When you are at cell (i, j), including the starting point $(1, 1)$ and the destination (n, n), you can take all the $a_{i,j}$ diamonds at this cell, and have a chance to raise the price of each diamond by $b_{i,j}$ dollars. You will sell all the diamonds you have with the final price in the end, and your goal is to choose the optimal path that will maximize your profits. Note that initially the price of each diamond is zero, and you have nothing to sell.

Input

The first line contains a single integer T $(1 \le T \le 10)$, the number of test cases. For each test case:

The first line contains a single integer n $(1 \le n \le 100)$, denoting the size of the grid.

Each of the following n lines contains n integers, the i-th line contains $a_{i,1}, a_{i,2}, \ldots, a_{i,n}$ $(1 \le a_{i,j} \le 10^6)$, denoting the number of diamonds in each cell.

Each of the following n lines contains n integers, the i-th line contains $b_{i,1}, b_{i,2}, \ldots, b_{i,n}$ $(1 \le b_{i,j} \le 10^6)$, denoting how much you can raise the price in each cell.

It is guaranteed that all the values of $a_{i,j}$ and $b_{i,j}$ are chosen uniformly at random from integers in $[1, 10^6]$. The randomness condition does not apply to the sample test case, but your solution must pass the sample as well.

Output

For each test case, output a single line containing an integer: the maximum number of dollars you can earn by selling diamonds.

Example

standard input	standard output
1 4 2 3 1 5 6 3 2 4 3 5 1 4 5 2 4 1 3 2 5 1 2 4 3 5 1 2 3 4 4 3 5 3	528

Problem J. Road Discount

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

There are n cities in Byteland, labeled by 1 to n. The Transport Construction Authority of Byteland is planning to construct $n-1$ bidirectional roads among these cities such that every pair of different cities are connected by these roads directly or indirectly.

The engineering company has offered m possible candidate roads to construct. The i-th candidate road will cost c_i dollars, and if it is finally constructed, there will be a road connecting the u_i-th city and the v_i-th city directly. Fortunately, each road has its discounted price, the i-th of which is d_i.

The Transport Construction Authority of Byteland can buy at most k roads at their discounted prices. Please write a program to help the Transport Construction Authority find the cheapest solution for $k = 0, 1, 2, \ldots, n-1$.

Input

The first line contains a single integer T $(1 \leq T \leq 10)$, the number of test cases. For each test case:

The first line contains two integers n and m $(2 \leq n \leq 1\,000,\ n-1 \leq m \leq 200\,000)$, denoting the number of cities and the number of candidate roads.

Each of the following m lines contains four integers u_i, v_i, c_i and d_i $(1 \leq u_i, v_i \leq n,\ u_i \neq v_i,\ 1 \leq d_i \leq c_i \leq 1\,000)$, describing a candidate road.

Output

For each test case, output n lines, the i-th $(1 \leq i \leq n)$ of which containing an integer, denoting the cheapest total cost to construct $n-1$ roads when $k = i-1$.

It is guaranteed that the answer always exists.

Example

standard input	standard output
1 5 6 1 2 1 1 2 3 2 1 2 4 3 2 2 5 4 3 1 3 5 3 4 5 6 1	10 7 6 5 5

Problem K. Segment Tree with Pruning

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

Chenjb is struggling with data stucture now. He is trying to solve a problem using segment tree. Chenjb is a freshman in programming contest, and he wrote down the following C/C++ code and ran "`Node* root = build(1, n)`" to build a standard segment tree on range $[1, n]$:

```
Node* build(long long l, long long r) {
    Node* x = new(Node);
    if (l == r) return x;
    long long mid = (l + r) / 2;
    x -> lchild = build(l, mid);
    x -> rchild = build(mid + 1, r);
    return x;
}
```

Chenjb submitted his code, but unfortunately, got MLE (Memory Limit Exceeded). Soon Chenjb realized that his program will new a large quantity of nodes, and he decided to reduce the number of nodes by pruning:

```
Node* build(long long l, long long r) {
    Node* x = new(Node);
    if (r - l + 1 <= k) return x;
    long long mid = (l + r) / 2;
    x -> lchild = build(l, mid);
    x -> rchild = build(mid + 1, r);
    return x;
}
```

You know, Chenjb is a freshman, so he will try different values of k to find the optimal one. You will be given the values of n and k, please tell him the number of nodes that will be generated by his new program.

Input

The first line contains a single integer T $(1 \leq T \leq 10\,000)$, the number of test cases. For each test case:

The only line contains two integers n and k $(1 \leq k \leq n \leq 10^{18})$, denoting a query.

Output

For each query, print a single line containing an integer, denoting the number of segment tree nodes.

Example

standard input	standard output
3 100000 1 100000 50 1000000000000000000 1	199999 4095 1999999999999999999

Problem L. Tree Planting

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

There are n buildings on the side of Bytestreet, standing sequentially one next to the other, labeled by $1, 2, \ldots, n$ from east to west. The city planning investigation is going to plant some trees in front of these buildings. A tree planting plan is beautiful if and only if it satisfies all the conditions below:

- At least one tree is planted.
- A tree can only be planted in front of a building, and two trees can not be planted in front of the same building. In other words, let c_i denote the number of trees planted in front of the i-th building, $c_i \in \{0, 1\}$ holds for all $i = 1, 2, \ldots, n$.
- For every possible value of i ($1 \le i < n$), c_i and c_{i+1} shouldn't both be 1.
- For every possible value of i ($1 \le i \le n-k$), c_i and c_{i+k} shouldn't both be 1.

The beautifulness of a plan is defined as:

$$\prod_{1 \le i \le n, c_i = 1} w_i$$

You need to find the sum of beautifulness among all the beautiful plans. Two plans are considered different if there exists any i ($1 \le i \le n$) such that they differ in c_i.

Input

The first line contains a single integer T ($1 \le T \le 200$), the number of test cases. For each test case:

The first line contains two integers n and k ($2 \le k < n \le 300$), denoting the number of buildings and the value of k.

The second line contains n integers $w_1, w_2, \ldots, w_n$ ($1 \le w_i \le 10^9$).

It is guaranteed that there will be at most 10 test cases such that $n > 100$.

Output

For each test case, output a single line containing an integer, denoting the sum of beautifulness among all the beautiful plans. Note that the answer may be extremely large, so please print it modulo $10^9 + 7$ instead.

Example

standard input	standard output
2 5 3 1 1 1 1 1 5 4 1 2 3 4 5	10 55