Problem A. Salty Fish

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

Little Q has an apple tree with n nodes, labeled by $1, 2, \ldots, n$. The root of the tree is node 1, and the length of each edge is one unit. There are a_i apples on the i-th node. The price of each apple is one dollar, so if you sell t apples, you will gain t dollars.

Skywalkert, a close friend of Little Q, lost most of his money betting on programming contests, so he wants to steal some apples from this apple tree and sell them to make money.

The security system takes pictures of the nodes once per hour using m cameras. Let us denote $d(x, y)$ as the number of edges on the shortest path from node x to node y, and denote set $p(x, k)$ as $\{y \mid y \text{ is in subtree of } x \text{ and } d(x, y) \le k\}$. Note that $x \in p(x, k)$. The image from the i-th camera shows the picture of all the nodes in $p(x_i, k_i)$. If the security system detects a change in any of these images, it sounds an alarm, and the thief will be caught by Little Q.

Skywalkert is also a gifted hacker. He can lock some cameras so that images from these cameras will never change. Specifically, if he wants to lock the i-th camera, he needs to pay c_i dollars to do such a hack. Skywalkert will pay for all hacks after he steals the apples and sells them.

Please write a program to help Skywalkert earn the most money possible without being caught.

Input

The first line of the input contains an integer T ($1 \le T \le 10\,000$), denoting the number of test cases.

Each test case starts by a line with two integers n and m ($1 \le n, m \le 300\,000$), denoting the number of nodes and cameras.

The second line of a test case contains $n - 1$ integers $f_2, f_3, \ldots, f_n$ ($1 \le f_i < i$), denoting the parents of nodes $2, 3, \ldots, n$.

The third line of a test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$), denoting the number of apples on nodes $1, 2, \ldots, n$.

Each of the next m lines of a test case contains three integers x_i, k_i, and c_i ($1 \le x_i \le n$, $0 \le k_i \le n$, $1 \le c_i \le 10^9$), denoting the parameters of each camera.

It is guaranteed that the sum of all n is at most 10^6, and the sum of all m is at most 10^6.

Output

For each test case, print a single line containing an integer denoting the maximum amount of dollars Skywalkert can earn.

Example

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

Problem B. Nonsense Time

Input file:	*standard input*
Output file:	*standard output*
Time limit:	12 seconds
Memory limit:	512 mebibytes

You a given a permutation $p_1, p_2, \ldots, p_n$ of size n. Initially, all elements in p are frozen. There will be n stages that these elements will become available one by one. On stage i, the element p_{k_i} will become available.

For each i, find the longest increasing subsequence among available elements after the first i stages.

Input

The first line of the input contains an integer T ($1 \le T \le 3$), denoting the number of test cases.

In each test case, there is one integer n ($1 \le n \le 50\,000$) on the first line, denoting the size of permutation.

In the second line of each test case, there are n distinct integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le n$), denoting the permutation.

In the third line of each test case, there are n distinct integers $k_1, k_2, \ldots, k_n$ ($1 \le k_i \le n$), describing each stage.

It is guaranteed that $p_1, p_2, \ldots, p_n$ and $k_1, k_2, \ldots, k_n$ are generated uniformly at random among all possible permutations of the given size.

Output

For each test case, print a single line containing n integers, where the i-th integer denotes the length of the longest increasing subsequence among available elements after the first i stages.

Example

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

Problem C. Milk Candy

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Calabash is now playing an RPG game on his computer. In this game, there are n unknown numbers $x_1, x_2, \ldots, x_n$ and m NPCs selling hints. The i-th NPC is selling c_i hints. Each hint contains three integers, l_j, r_j, and w_j, which means Calabash can pay w_j coins to buy this hint, and this hint can tell Calabash the value of $x_{l_j} + x_{l_j+1} + \ldots + x_{r_j-1} + x_{r_j}$.

The goal of the game is to figure out all the n unknown numbers. Clever Calabash knows how to buy hints optimally, but NPCs are greedy: for the i-th NPC, Calabash must buy exactly k_i hints from him. Note that a single hint can't be bought more than once.

This problem is much more difficult for Calabash. Please write a program to help Calabash find the minimum number of coins he needs to pay to figure out all numbers, or determine that it is impossible.

Input

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

In each test case, there are two integers n and m ($1 \le n, m \le 80$) on the first line, denoting the number of unknown numbers and NPCs.

Then follow m parts. Each part starts with a line with two integers c_i and k_i ($1 \le k_i \le c_i$), denoting the number of hints the i-th NPC has and the limit for the i-th NPC.

Each of the next c_i lines contains three integers, l_j, r_j, and w_j ($1 \le l_j \le r_j \le n$, $1 \le w_j \le 10^6$), describing the hints offered by the i-th NPC.

It is guaranteed that, in each test case, the sum of all c_i is at most 80.

Output

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

Example

standard input	standard output
2 2 2 1 1 1 2 1 3 2 1 1 10 2 2 100 1 2 1000 2 2 1 1 1 1 1 1 1 1 1 2	111 -1

Problem D. Radar Scanner

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

There are n rectangle radar scanners on the ground. Their sides are all parallel to the coordinate axes. Each scanner covers some grid squares on the ground. The i-th scanner covers all the squares (x, y) satisfying $x_{i,1} \le x \le x_{i,2}$ and $y_{i,1} \le y \le y_{i,2}$.

Today, the radar system is facing a critical low-power problem. You need to choose exactly three scanners such that there exists a square covered by all scanners.

Your task is to count how many tuples (i, j, k) you can choose so that $1 \le i < j < k \le n$ and there exists a square covered by all three scanners i, j, and k.

Input

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

Each test case starts with a line containing an integer n ($3 \le n \le 100\,000$), denoting the number of radar scanners.

Each of the next n lines contains four integers, $x_{i,1}$, $y_{i,1}$, $x_{i,2}$, and $y_{i,2}$ ($1 \le x_{i,1} \le x_{i,2} \le 1000$, $1 \le y_{i,1} \le y_{i,2} \le 1000$), describing the i-th radar scanner.

Output

For each test case, print a single line containing a single integer: the number of possible tuples.

Example

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

Problem E. Snowy Smile

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

There are n pirate chests buried in Byteland, labeled by $1, 2, \ldots, n$. The i-th chest is located at point (x_i, y_i) on the plane, and its value is w_i. Note that w_i can be negative since the pirate can put some poisonous gases into the chest. When you open the i-th pirate chest, you will get w_i value.

You want to make money from these pirate chests. You can select a rectangle such that its sides are all parallel to the coordinate axes, and then open all the chests inside this rectangle or on its border. Note that you must open all the chests within that range regardless of whether their values are positive or negative. But you can choose a rectangle with nothing in it to get a zero sum.

Please write a program to find the maximum total value of a rectangle.

Input

The first line of the input contains an integer T $(1 \le T \le 100)$, denoting the number of test cases.

Each test case starts with a line containing an integer n $(1 \le n \le 2000)$, denoting the number of pirate chests.

Each of the next n lines contains three integers, x_i, y_i, and w_i $(-10^9 \le x_i, y_i, w_i \le 10^9)$, describing the i-th pirate chest.

It is guaranteed that the sum of n in all test cases is at most $10\,000$.

Output

For each test case, print a single line containing a single integer: the maximum total value.

Example

standard input	standard output
2 4 1 1 50 2 1 50 1 2 50 2 2 -500 2 -1 1 5 -1 1 1	100 6

Problem F. Faraway

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

A squad of n soldiers is dispatched to somewhere in Byteland. Currently, i-th soldier is at location (x_i, y_i). The soldiers are going to set off now, but the target location is not so clear.

Assume the target location is at (x_e, y_e). It is clear for all soldiers that x_e and y_e are both non-negative integers within the range $[0, m]$. Apart from that, for i-th soldier, the only thing he knows is that $(|x_i - x_e| + |y_i - y_e|) \bmod k_i = t_i$.

To find the correct target location, these soldiers are working on the information they have now. Please write a program to figure out the number of possible target locations.

Input

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

Each test case starts with a line containing two integers n and m ($1 \le n \le 10$, $1 \le m \le 10^9$), denoting the number of soldiers and the upper bound for x_e and y_e.

Each of the next n lines contains four integers, x_i, y_i, k_i, and t_i ($0 \le x_i, y_i \le m$, $2 \le k_i \le 5$, $0 \le t_i < k_i$), denoting what i-th soldier knows.

Output

For each test case, print a single line containing a single integer: the number of possible target locations.

Example

standard input	standard output
2 2 5 1 2 4 2 3 1 2 1 2 5 1 2 4 2 1 2 4 3	10 0

Problem G. Support or Not

Input file:	*standard input*
Output file:	*standard output*
Time limit:	8 seconds
Memory limit:	512 mebibytes

There are n spheres in the three-dimensional space, labeled by $1, 2, \ldots, n$. The center of the i-th sphere is at point (x_i, y_i, z_i), and its radius is r_i.

Let us denote the distance between spheres i and j as

$$d(i,j) = \max\left(0, \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} - r_i - r_j\right).$$

That means choosing two points P and Q, where P is inside or on the surface of sphere i and Q is inside or on the surface of sphere j, and minimizing the Euclidean distance between P and Q.

There are $\frac{n(n-1)}{2}$ pairs (i,j) such that $1 \le i < j \le n$. Please find the smallest k values among the values of $d(i,j)$ for all these pairs.

Input

The first line of the input contains an integer T ($1 \le T \le 3$), denoting the number of test cases.

Each test case starts with a single line containing two integers n and k ($2 \le n \le 100\,000$, $1 \le k \le \min(300, \frac{n(n-1)}{2})$), denoting the number of spheres and the parameter k.

Each of the next n lines contains four integers, x_i, y_i, z_i, and r_i ($0 \le x_i, y_i, z_i \le 10^6$, $1 \le r_i \le 10^6$), describing i-th sphere.

Output

For each test case, print k lines, each line containing a single integer: the smallest k values among $d(i,j)$ in non-decreasing order. To avoid precision error, print the values of $\lceil d(i,j) \rceil$: the values of the respective $d(i,j)$ rounded up. For example, $\lceil 5 \rceil = 5$, and $\lceil 5.1 \rceil = 6$.

Example

standard input	standard output
1 4 6 0 0 0 1 0 3 2 2 3 2 1 1 1 1 2 2	0 0 0 1 1 2

Problem H. TDL

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

For a positive integer n, let us denote function $f(n, m)$ as the m-th smallest integer x such that $x > n$ and $\gcd(x, n) = 1$. For example, $f(5, 1) = 6$ and $f(5, 5) = 11$.

You are given the values of m and $(f(n, m) - n) \oplus n$, where "$\oplus$" denotes the bitwise XOR operation. Please write a program to find the smallest positive integer n such that $(f(n, m) - n) \oplus n = k$, or determine it is impossible.

Input

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

Each test case is denoted by a single line containing two integers k and m ($1 \le k \le 10^{18}$, $1 \le m \le 100$).

Output

For each test case, print a single line containing a single integer: the smallest value of n. If there is no solution, output "-1" instead.

Example

standard input	standard output
2 3 5 6 100	5 -1

Problem I. Three Investigators

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

Chitanda owns a sequence $a_1, a_2, \ldots, a_n$ with n integers, and she wants to play a game with Skywalkert.

First, Chitanda will select a parameter k and remove $a_{k+1}, a_{k+2}, \ldots, a_n$. Thus there will be exactly k integers in sequence a.

Then Skywalkert can select a subsequence of a and remove it from a. Assume the selected subsequence is $a_{p_1}, a_{p_2}, \ldots, a_{p_m}$. He should ensure that $p_1 < p_2 < \ldots < p_m$ and $a_{p_1} \le a_{p_2} \le \ldots \le a_{p_m}$.

Skywalkert can do the above operation for no more than 5 times. His score is the sum of all the integers selected by him in these no more than 5 operations.

For each possible parameter k selected by Chitanda, write a program to help Skywalkert know the maximum score he can achieve.

Input

The first line of the input contains an integer T ($1 \le T \le 10\,000$), denoting the number of test cases.

In each test case, there is one integer n ($1 \le n \le 100\,000$) on the first line, denoting the length of a.

In the second line of a test case, there are n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$), denoting the sequence.

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

Output

For each test case, print a single line containing n integers $s_1, s_2, \ldots, s_n$, where s_i denotes the maximum score of Skywalkert when $k = i$.

Example

standard input	standard output
1 8 8 7 6 5 1 3 2 4	8 15 21 26 27 30 30 34

Problem J. Road Manager

Input file: *standard input*
Output file: *standard output*
Time limit: 4 seconds
Memory limit: 512 mebibytes

There are $n \times m$ cells on a grid, the top left cell is at $(1, 1)$ while the bottom right cell is at (n, m).

Two cells are considered adjacent if they share a common side. Note that the grid is circular: (i, m) is also adjacent to $(i, 1)$. However, there is no such rule for connecting the first row and the last row.

There is exactly one bidirectional road between two adjacent cells:

- For every cell (i, j) such that $1 \le i \le n$ and $1 \le j \le m$, there is a road between it and $(i, j \bmod m + 1)$. Note that the grid is circular, so when $j = m$, there is a road between (i, m) and $(i, 1)$.
- For every cell (i, j) such that $1 \le i < n$ and $1 \le j \le m$, there is a road between it and $(i + 1, j)$.

The roads form a weighted graph, every road has its weight. You will be given q queries. In the i-th query, you will be given two integers, l_i and r_i. Consider the graph that remains if we remove all cells (x, y) such that $1 \le x \le n$ and $l_i \le y \le r_i$, together with the roads linked to them. You need to find the total weight of the minimum spanning tree (MST) of the resulting graph.

Note that different queries are independent. In other words, the removed cells and roads come back after you answer a query.

Input

There is only one test case in each test.

The first line of this test case contains six integers n, m, SA, SB, SC, and lim ($1 \le n \le 100$, $4 \le m \le 10\,000$, and $1 \le SA, SB, SC, lim \le 10^9$).

Since the grid might be rather large, you have to generate the input using the following C/C++ code, where `addedge(a, b, c, d, w)` means that we add a road between (a, b) and (c, d) with weight w:

```
unsigned int SA, SB, SC; int lim;
int getweight() {
    SA ^= SA << 16;
    SA ^= SA >> 5;
    SA ^= SA << 1;
    unsigned int t = SA;
    SA = SB;
    SB = SC;
    SC ^= t ^ SA;
    return SC % lim + 1;
}
void gen() {
    scanf("%d%d%u%u%u%d", &n, &m, &SA, &SB, &SC, &lim);
    int i, j, w;
    for (i = 1; i <= n; i++)
        for (j = 1; j <= m; j++) {
            w = getweight();
            if (j < m) {
                addedge(i, j, i, j + 1, w);
            } else {
                addedge(i, j, i, 1, w);
            }
        }
    for (i = 1; i < n; i++)
        for (j = 1; j <= m; j++) {
            w = getweight();
            addedge(i, j, i + 1, j, w);
        }
}
```

A note for coders in other languages: `int` is a 32-bit signed integer type, whereas `unsigned int` is a 32-bit unsigned integer type. The result of every operation with `unsigned int` is truncated to the last 32 bits. This effectively means each result is taken modulo 2^{32}.

On the second line, there is one integer q ($1 \le q \le 10\,000$), denoting the number of queries.

Each of the next q lines contains two integers l_i and r_i ($1 < l_i \le r_i < m$) describing a query.

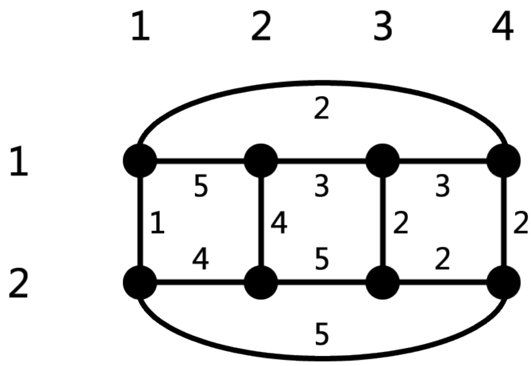
Output

For each query, print a single line containing an integer: the total weight of the MST of the respective graph.

Example

standard input	standard output
2 4 1 2 3 5 3 2 2 2 3 3 3	9 5 13

Note



The generated graph in the example.

Problem K. Monster Hunter

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	512 mebibytes

Little Q is fighting against scary monsters in the game "Monster Hunter". The battlefield consists of n intersections, labeled by $1, 2, \ldots, n$, connected by $n-1$ bidirectional roads like a tree. Little Q is now at intersection 1 and has X health points (HP).

There is a monster at each intersection except intersection 1. When Little Q moves to the k-th intersection for the first time, he must fight the monster at that intersection. During the fight, he will lose a_i HP. And when he finally beats the monster, he will be awarded b_i HP. Note that when HP becomes negative (< 0), the game will end, so never let this happen. If Little Q visits the same intersection more than once, the fight happens only on the first visit, as monsters do not have an extra life.

When all monsters are cleared, Little Q will win the game. Please write a program to compute the minimum initial HP that can lead to victory.

Input

The first line of the input contains an integer T ($1 \le T \le 2000$), denoting the number of test cases.

In each test case, there is one integer n ($2 \le n \le 100\,000$) on the first line, denoting the number of intersections.

Each of the next $n-1$ lines contains two integers a_i and b_i ($0 \le a_i, b_i \le 10^9$) describing monsters at intersections $2, 3, \ldots, n$.

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

It is guaranteed that the sum of all n is at most 10^6.

Output

For each test case, print a single line containing an integer, denoting the minimum initial HP required to win the game.

Example

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

Problem L. Game Prediction

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Sunset and Elephant are playing a game on a sequence $b_1, b_2, \ldots, b_m$. The two players move in turns, and Sunset moves first. In each move, the current player selects a value which is either at the beginning of the sequence or at the end of the sequence, adds it to this player's score and removes the value from the sequence. The game ends when the sequence is empty. Both players want to maximize their scores and will play optimally.

You are given a sequence $a_1, a_2, \ldots, a_n$ and q queries. In the i-th query, you will be given two integers l_i and r_i. Please write a program to figure out the final result of the game when they choose $a_{l_i}, a_{l_i+1}, \ldots, a_{r_i}$ as the initial sequence b. Here, $m = r_i - l_i + 1$ and $b_j = a_{l_i+j-1}$ for every j such that $1 \le j \le m$.

Input

There is only one test case in each test.

The test case starts with a line containing two integers n and q ($1 \le n \le 100\,000$, $1 \le q \le 200\,000$) on the first line, denoting the length of the sequence and the number of queries.

On the second line, there are n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

Each of the next q line contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$), denoting the queries.

It is guaranteed that all the values of a_i are chosen uniformly at random from integers in the range $[1, 10^9]$. 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 two integers S and E, denoting the final score of Sunset and the final score of Elephant.

Example

standard input	standard output
5 4 7 9 3 5 2 1 5 3 5 2 4 1 2	12 14 5 5 12 5 9 7