Problem A. Salty Fish

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

Little Q has an apple tree with n nodes, labeled by $1, 2, \ldots, n$. The root of the tree is the 1-th node 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's denote $d(x, y)$ as the number of edges on the shortest path from the x-th node to the y-th node, and denote set $p(x, k)$ as $\{y|\texttt{y in x's subtree and } d(x, y) \leq k\}$. 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, 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 want to lock the i-th camera, he needs to pay c_i dollars to do such a hack.

Please write a program to help Skywalkert make money optimally without being caught.

Input

The first line of the input contains an integer $T(1 \leq T \leq 10000)$, denoting the number of test cases.

In each test case, there are two integers $n, m(1 \leq n, m \leq 300000)$ in the first line, denoting the number of nodes and cameras.

In the second line, there are $n - 1$ integers $f_2, f_3, ..., f_n(1 \leq f_i < i)$, denoting the father of each node.

In the third line, there are n integers $a_1, a_2, ..., a_n(1 \leq a_i \leq 10^9)$, denoting the number of apples on each node.

For the next m lines, each line contains three integers $x_i, k_i, c_i(1 \leq x_i \leq n, 0 \leq k_i \leq n, 1 \leq c_i \leq 10^9)$, denoting each camera.

It is guaranteed that $\sum n \leq 10^6$ and $\sum m \leq 10^6$.

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:	14 seconds
Memory limit:	512 megabytes

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 $T (1 \leq T \leq 3)$, denoting the number of test cases.

In each test case, there is one integer $n (1 \leq n \leq 50000)$ in the first line, denoting the size of permutation.

In the second line, there are n distinct integers $p_1, p_2, ..., p_n (1 \leq p_i \leq n)$, denoting the permutation.

In the third line, there are n distinct integers $k_1, k_2, ..., k_n (1 \leq k_i \leq n)$, describing each stage.

It is guaranteed that $p_1, p_2, ..., p_n$ and $k_1, k_2, ..., k_n$ are generated randomly.

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:	5 seconds
Memory limit:	512 megabytes

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, w_j, which means Calabash can pay w_j coins to buy this hint, and this hint can tell Calabash the value of $x_{l_j} + x_{l_j+1} + \cdots + x_{r_j-1} + x_{r_j}$.

The target 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 each 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 cheapest way, or determine it is impossible.

Input

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

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

For the next m parts, there are two integers $c_i, k_i(1 \leq k_i \leq c_i)$ in the first line, denoting the number of hints the i-th NPC has and the limit for the i-th NPC.

For the next c_i lines, each line contains three integers $l_j, r_j, w_j(1 \leq l_j \leq r_j \leq n, 1 \leq w_j \leq 10^6)$, describing each hint offered by the i-th NPC.

It is guaranteed that $\sum c_i \leq 80$ in each test case.

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. Speed Dog

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

Sunset and Elephant formed a programming contest team named "Speed Dog". There are n problems in the online judge, labeled by $1, 2, \ldots, n$.

For the i-th problem, Sunset needs to code a_i bytes while Elephant needs to code b_i bytes. But they can work on problems together. Specifically, for the i-th problem, they can choose a real number $x_i (0 \leq x_i \leq 1)$, split this problem into two parts A and B, then assign Sunset to code part A and assign Elephant to code part B. As a result, Sunset will code $a_i \times x_i$ bytes, and Elephant will code $b_i \times (1 - x_i)$ bytes.

Now they want to solve all the problems whose labels are not larger than k. Assume Sunset codes X bytes in total and Elephant codes Y bytes in total. It is too tired for a coder to code too much code. Please help them find an assignment that $\max(X, Y)$ is minimized.

Input

The first line of the input contains an integer $T (1 \leq T \leq 10000)$, denoting the number of test cases.

In each test case, there is one integer $n (1 \leq n \leq 250000)$ in the first line, denoting the number of problems.

For the next n lines, each line contains two integers $a_i, b_i (1 \leq a_i, b_i \leq 1000)$, denoting each problem.

It is guaranteed that $\sum n \leq 10^6$.

Output

For each test case, print n lines in format `u/v`, where the i-th line denotes the minimum value of $\max(X, Y)$ when $k = i$. Note that you should guarantee that $\gcd(u, v) = 1$.

Example

standard input	standard output
1 3 1 3 3 1 2 2	3/4 1/1 2/1

Problem E. Snowy Smile

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

There are n pirate chests buried in Byteland, labeled by $1, 2, \ldots, n$. The i-th chest's location is (x_i, y_i), and its value is w_i, w_i can be negative since the pirate can add 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, the sides of which are all paralleled to the axes, and then all the chests inside it or on its border will be opened. Note that you must open all the chests within that range regardless of 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 best rectangle with maximum total value.

Input

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

In each test case, there is one integer $n(1 \leq n \leq 2000)$ in the first line, denoting the number of pirate chests.

For the next n lines, each line contains three integers $x_i, y_i, w_i(-10^9 \leq x_i, y_i, w_i \leq 10^9)$, denoting each pirate chest.

It is guaranteed that $\sum n \leq 10000$.

Output

For each test case, print a single line containing an integer, denoting 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 megabytes

n soldiers are dispatched to somewhere in Byteland. These 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 that x_e, y_e are both non-negative integers within $[0, m]$. For the 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 $T(1 \leq T \leq 10)$, denoting the number of test cases.

In each test case, there are two integers $n, m(1 \leq n \leq 10, 1 \leq m \leq 10^9)$ in the first line, denoting the number of soldiers and the upper bound of x_e, y_e.

For the next n lines, each line contains four integers $x_i, y_i, k_i, t_i(0 \leq x_i, y_i \leq m, 2 \leq k_i \leq 5, 0 \leq t_i < k_i)$, denoting what each soldier knows.

Output

For each test case, print a single line containing an integer, denoting 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:	6 seconds
Memory limit:	512 megabytes

There are n spheres in 3D-space, labeled by $1, 2, \ldots, n$. The i-th sphere's center is at (x_i, y_i, z_i), and the radius of it is r_i.

Let's denote the distance between the i-th sphere and the j-th sphere $d(i, j)$ as

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

That means choosing two points P and Q, where P is on the i-th sphere's surface or inside it, Q is on the j-th sphere's surface or inside it, and minimize the Euclidean distance bewteen P and Q.

There are $\frac{n(n-1)}{2}$ pairs of $i, j (1 \leq i < j \leq n)$, please find the k-th smallest values among these $d(i, j)$.

Input

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

In each test case, there are two integers $n, k (2 \leq n \leq 100000, 1 \leq k \leq \min(300, \frac{n(n-1)}{2}))$ in the first line, denoting the number of spheres and the parameter k.

For the next n lines, each line contains four integers $x_i, y_i, z_i, r_i (0 \leq x_i, y_i, z_i \leq 10^6, 1 \leq r_i \leq 10^6)$, denoting each sphere.

Output

For each test case, print k lines, each line contains an integer, denoting the k-th smallest values among these $d(i, j)$. You should print them in non-decreasing order. To avoid precision error, print $\lceil d(i, j) \rceil$ instead. 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 megabytes

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

You are given the value 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 that $(f(n, m) - n) \oplus n = k$, or determine it is impossible.

Input

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

In each test case, there are two integers $k, m (1 \leq k \leq 10^{18}, 1 \leq m \leq 100)$.

Output

For each test case, print a single line containing an integer, denoting the smallest 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 megabytes

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

First, Chitanda will select a parameter k and remove $a_{k+1}, a_{k+2}, \dots, a_n$. 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 $a_{p_1}, a_{p_2}, \dots, a_{p_m}$, he should ensure $p_1 < p_2 < \dots < p_m$ and $a_{p_1} \leq a_{p_2} \leq \dots \leq a_{p_m}$.

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.

Given the parameter k selected by Chitanda, write a program to help Skywalkert maximize his score.

Input

The first line of the input contains an integer $T(1 \leq T \leq 10000)$, denoting the number of test cases.

In each test case, there is one integer $n(1 \leq n \leq 100000)$ in the first line, denoting the length of a.

In the second line, there are n integers $a_1, a_2, ..., a_n (1 \leq a_i \leq 10^9)$, denoting the sequence.

It is guaranteed that $\sum n \leq 500000$.

Output

For each test case, print a single line containing n integers $s_1, s_2, \dots, s_n$, 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. Ridiculous Netizens

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	512 megabytes

Mr. Bread has a tree T with n vertices, labeled by $1, 2, \dots, n$. Each vertex of the tree has a positive integer value w_i.

The value of a non-empty tree T is equal to $w_1 \times w_2 \times \cdots \times w_n$. A subtree of T is a connected subgraph of T that is also a tree.

Please write a program to calculate the number of non-empty subtrees of T whose values are not larger than a given number m.

Input

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

In each test case, there are two integers $n, m(1 \leq n \leq 2000, 1 \leq m \leq 10^6)$ in the first line, denoting the number of vertices and the upper bound.

In the second line, there are n integers $w_1, w_2, \dots, w_n (1 \leq w_i \leq m)$, denoting the value of each vertex.

Each of the following $n-1$ lines contains two integers $u_i, v_i (1 \leq u_i, v_i \leq n, u_i \neq v_i)$, denoting an bidirectional edge between vertices u_i and v_i.

Output

For each test case, print a single line containing an integer, denoting the number of valid non-empty subtrees. As the answer can be very large, output it modulo $10^9 + 7$.

Example

standard input	standard output
1 5 6 1 2 1 2 3 1 2 1 3 2 4 2 5	14

Problem K. 11 Dimensions

Input file:	`standard input`
Output file:	`standard output`
Time limit:	8 seconds
Memory limit:	512 megabytes

11 Dimensions is a cute contestant being talented in math. One day, 11 Dimensions came across a problem but didn't manage to solve it. Today you are taking training here, so 11 Dimensions turns to you for help.

You are given a decimal integer S with n bits $s_1s_2\dots s_n(0 \leq s_i \leq 9)$, but some bits can not be recognized now(replaced by "?"). The only thing you know is that S is a multiple of a given integer m.

There may be many possible values of the original S, please write a program to find the k-th smallest value among them. Note that you need to answer q queries efficiently.

Input

The first line of the input contains an integer $T(1 \leq T \leq 10000)$, denoting the number of test cases.

In each test case, there are three integers $n, m, q(1 \leq n \leq 50000, 2 \leq m \leq 20, 1 \leq q \leq 100000)$ in the first line, denoting the length of S, the parameter m, and the number of queries.

In the second line, there is a string s of length n, denoting the given decimal integer S. It is guaranteed that s_i is either an integer within $[0, 9]$ or "?", and s_1 is always an integer within $[1, 9]$.

For the next q lines, each line contains an integer $k_i(1 \leq k_i \leq 10^{18})$, denoting each query.

It is guaranteed that $\sum n \leq 500000$ and $\sum q \leq 10^6$.

Output

For each query, print a single line containing an integer, denoting the value of S. If the answer exists, print $S \bmod (10^9 + 7)$ instead, otherwise print "`-1`".

Example

standard input	standard output
1 5 5 5 2??3? 1 2 3 100 10000	20030 20035 20130 24935 -1

Problem L. Stay Real

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

In computer science, a heap is a specialized tree-based data structure which is essentially an almost complete tree that satisfies the heap property: in a min heap, for any given node C, if P is a parent node of C, then the key(the value) of P is less than or equal to the key of C. The node at the "top" of the heap(with no parents) is called the root node.

Usually, we may store a heap of size n in an array $h_1, h_2, \ldots, h_n$, where h_i denotes the key of the i-th node. The root node is the 1-th node, and the parent of the $i(2 \leq i \leq n)$-th node is the $\lfloor \frac{i}{2} \rfloor$-th node.

Sunset and Elephant is playing a game on a min heap. The two players move in turns, and Sunset moves first. In each move, the current player selects a node which has no children, adds its key to this player's score and removes the node from the heap.

The game ends when the heap is empty. Both players want to maximize their scores and will play optimally. Please write a program to figure out the final result of the game.

Input

The first line of the input contains an integer $T(1 \leq T \leq 10000)$, denoting the number of test cases.

In each test case, there is one integer $n(1 \leq n \leq 100000)$ in the first line, denoting the number of nodes.

In the second line, there are n integers $h_1, h_2, ..., h_n (1 \leq h_i \leq 10^9, h_{\lfloor \frac{i}{2} \rfloor} \leq h_i)$, denoting the key of each node.

It is guaranteed that $\sum n \leq 10^6$.

Output

For each test case, print a single line containing two integers S and E, denoting the final score of Sunset and Elephant.

Example

standard input	standard output
1 3 1 2 3	4 2