Problem A. Ascending Rating

Input file:	`stdin`
Output file:	`stdout`
Time limit:	3 seconds
Memory limit:	512 megabytes

Before the start of contest, there are n ICPC contestants waiting in a long queue. They are labeled by 1 to n from left to right. It can be easily found that the i-th contestant's QodeForces rating is a_i.

Little Q, the coach of Quailty Normal University, is bored to just watch them waiting in the queue. He starts to compare the rating of the contestants. He will pick a continous interval with length m, say $[l, l+m-1]$, and then inspect each contestant from left to right. Initially, he will write down two numbers $maxrating = 0$ and $count = 0$. Everytime he meets a contestant k with strictly higher rating than $maxrating$, he will change $maxrating$ to a_k and $count$ to $count + 1$.

Little T is also a coach waiting for the contest. He knows Little Q is not good at counting, so he is wondering what are the correct final value of $maxrating$ and $count$. Please write a program to figure out the answer.

Input

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

In each test case, there are 7 integers $n, m, k, p, q, r, MOD(1 \leq m, k \leq n \leq 10^7, 5 \leq p, q, r, MOD \leq 10^9)$ in the first line, denoting the number of contestants, the length of interval, and the parameters k, p, q, r, MOD.

In the next line, there are k integers $a_1, a_2, ..., a_k (0 \leq a_i \leq 10^9)$, denoting the rating of the first k contestants.

To reduce the large input, we will use the following generator. The numbers p, q, r and MOD are given initially. The values $a_i (k < i \leq n)$ are then produced as follows :

$$a_i = (p \times a_{i-1} + q \times i + r) \bmod MOD$$

It is guaranteed that $\sum n \leq 7 \times 10^7$ and $\sum k \leq 2 \times 10^6$.

Output

Since the output file may be very large, let's denote $maxrating_i$ and $count_i$ as the result of interval $[i, i+m-1]$.

For each test case, you need to print a single line containing two integers A and B, where :

$$A = \sum_{i=1}^{n-m+1} (maxrating_i \oplus i)$$

$$B = \sum_{i=1}^{n-m+1} (count_i \oplus i)$$

Note that "$\oplus$" denotes binary XOR operation.

Examples

stdin	stdout
1 10 6 10 5 5 5 5 3 2 2 1 5 7 6 8 2 9	46 11

Problem B. Cut The String

Input file:	`stdin`
Output file:	`stdout`
Time limit:	3 seconds
Memory limit:	512 megabytes

A string is palindromic if it reads the same from left to right.

Given a string $S[1..n]$, Little Q will ask you m queries. For each query, Little Q will give you 2 integers l_i, r_i, you need to find the number of ways to cut the continous substring $S[l_i..r_i]$ into two non-empty palindromic strings. That is, find the number of $k(l_i \le k < r_i)$ satisfying $S[l_i..k]$ and $S[k+1..r_i]$ are both palindromic strings.

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 2 integers $n, m(2 \le n \le 100000, 1 \le m \le 100000)$ in the first line, denoting the length of S and the number of queries.

In the next line, there is a string S consists of n lower-case English letters.

Then in the following m lines, there are 2 integers $l_i, r_i(1 \le l_i < r_i \le n)$ in each line, denoting a query.

Output

For each query, print a single line containing an integer, denoting the answer.

Examples

stdin	stdout
1 10 5 aaaabababb 1 4 1 5 4 9 6 10 1 10	3 1 3 1 0

Problem C. Dynamic Graph Matching

Input file:	stdin
Output file:	stdout
Time limit:	4 seconds
Memory limit:	512 megabytes

In the mathematical discipline of graph theory, a matching in a graph is a set of edges without common vertices.

You are given an undirected graph with n vertices, labeled by $1, 2, ..., n$. Initially the graph has no edges.

There are 2 kinds of operations :

- "`+ u v`", add an edge (u, v) into the graph, multiple edges between same pair of vertices are allowed.
- "`- u v`", remove an edge (u, v), it is guaranteed that there are at least one such edge in the graph.

Your task is to compute the number of matchings with exactly k edges after each operation for $k = 1, 2, 3, ..., \frac{n}{2}$. Note that multiple edges between same pair of vertices are considered different.

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 2 integers $n, m(2 \le n \le 10, n \bmod 2 = 0, 1 \le m \le 30000)$, denoting the number of vertices and operations.

For the next m lines, each line describes an operation, and it is guaranteed that $1 \le u < v \le n$.

Output

For each operation, print a single line containing $\frac{n}{2}$ integers, denoting the answer for $k = 1, 2, 3, ..., \frac{n}{2}$. Since the answer may be very large, please print the answer modulo $10^9 + 7$.

Examples

stdin	stdout
1	1 0
4 8	2 1
+ 1 2	3 1
+ 3 4	4 2
+ 1 3	3 1
+ 2 4	2 1
- 1 2	3 1
- 3 4	4 2
+ 1 2	
+ 3 4	

Problem D. Euler Function

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	512 megabytes

In number theory, Euler's totient function $\varphi(n)$ counts the positive integers up to a given integer n that are relatively prime to n. It can be defined more formally as the number of integers k in the range $1 \le k \le n$ for which the greatest common divisor $\gcd(n, k)$ is equal to 1.

For example, $\varphi(9) = 6$ because $1, 2, 4, 5, 7$ and 8 are coprime with 9. As another example, $\varphi(1) = 1$ since for $n = 1$ the only integer in the range from 1 to n is 1 itself, and $\gcd(1, 1) = 1$.

A composite number is a positive integer that can be formed by multiplying together two smaller positive integers. Equivalently, it is a positive integer that has at least one divisor other than 1 and itself. So obviously 1 and all prime numbers are not composite number.

In this problem, given integer k, your task is to find the k-th smallest positive integer n, that $\varphi(n)$ is a composite number.

Input

The first line of the input contains an integer $T(1 \le T \le 100000)$, denoting the number of test cases.

In each test case, there is only one integer $k(1 \le k \le 10^9)$.

Output

For each test case, print a single line containing an integer, denoting the answer.

Examples

stdin	stdout
2 1 2	5 7

Problem E. Find The Submatrix

Input file:	`stdin`
Output file:	`stdout`
Time limit:	3 seconds
Memory limit:	512 megabytes

Little Q is searching for the submatrix with maximum sum in a matrix of n rows and m columns. The standard algorithm is too hard for him to understand, so he (and you) only considers those submatrixes with exactly m columns.

It is much easier now. But Little Q always thinks the answer is too small. So he decides to reset no more than A cells' value to 0, and choose no more than B disjoint submatrixes to achieve the maximum sum. Two submatrix are considered disjoint only if they do not share any common cell.

Please write a program to help Little Q find the maximum sum. Note that he can choose nothing so the answer is always non-negative.

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 4 integers $n, m, A, B(1 \leq n \leq 100, 1 \leq m \leq 3000, 0 \leq A \leq 10000, 1 \leq B \leq 3)$.

Each of the following n lines contains m integers, the j-th number on the i-th of these lines is $w_{i,j}(|w_{i,j}| \leq 10^9)$, denoting the value of each cell.

Output

For each test case, print a single line containing an integer, denoting the maximum sum.

Examples

stdin	stdout
2 5 1 0 1 3 -1 5 -1 -2 5 1 1 1 3 -1 5 -1 -2	7 8

Problem F. Grab The Tree

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	512 megabytes

Little Q and Little T are playing a game on a tree. There are n vertices on the tree, labeled by $1, 2, ..., n$, connected by $n-1$ bidirectional edges. The i-th vertex has the value of w_i.

In this game, Little Q needs to grab some vertices on the tree. He can select any number of vertices to grab, but he is not allowed to grab both vertices that are adjacent on the tree. That is, if there is an edge between x and y, he can't grab both x and y. After Q's move, Little T will grab all of the rest vertices. So when the game finishes, every vertex will be occupied by either Q or T.

The final score of each player is the bitwise XOR sum of his choosen vertices' value. The one who has the higher score will win the game. It is also possible for the game to end in a draw. Assume they all will play optimally, please write a program to predict the result.

Input

The first line of the input contains an integer $T (1 \leq T \leq 20)$, 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 vertices.

In the next line, there are n integers $w_1, w_2, ..., w_n (1 \leq w_i \leq 10^9)$, denoting the value of each vertex.

For the next $n-1$ lines, each line contains two integers u and v, denoting a bidirectional edge between vertex u and v.

Output

For each test case, print a single line containing a word, denoting the result. If Q wins, please print "`Q`". If T wins, please print "`T`". And if the game ends in a draw, please print "`D`".

Examples

stdin	stdout
1 3 2 2 2 1 2 1 3	Q

Problem G. Interstellar Travel

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2 seconds
Memory limit:	512 megabytes

After trying hard for many years, Little Q has finally received an astronaut license. To celebrate the fact, he intends to buy himself a spaceship and make an interstellar travel.

Little Q knows the position of n planets in space, labeled by 1 to n. To his surprise, these planets are all coplanar. So to simplify, Little Q put these n planets on a plane coordinate system, and calculated the coordinate of each planet (x_i, y_i).

Little Q plans to start his journey at the 1-th planet, and end at the n-th planet. When he is at the i-th planet, he can next fly to the j-th planet only if $x_i < x_j$, which will cost his spaceship $x_i \times y_j - x_j \times y_i$ units of energy. Note that this cost can be negative, it means the flight will supply his spaceship.

Please write a program to help Little Q find the best route with minimum total cost.

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 is an integer $n(2 \leq n \leq 200000)$ in the first line, denoting the number of planets.

For the next n lines, each line contains 2 integers $x_i, y_i (0 \leq x_i, y_i \leq 10^9)$, denoting the coordinate of the i-th planet. Note that different planets may have the same coordinate because they are too close to each other. It is guaranteed that $y_1 = y_n = 0, 0 = x_1 < x_2, x_3, ..., x_{n-1} < x_n$.

Output

For each test case, print a single line containing several distinct integers $p_1, p_2, ..., p_m (1 \leq p_i \leq n)$, denoting the route you chosen is $p_1 \to p_2 \to ... \to p_{m-1} \to p_m$. Obviously p_1 should be 1 and p_m should be n. You should choose the route with minimum total cost. If there are multiple best routes, please choose the one with the smallest lexicographically.

A sequence of integers a is lexicographically smaller than a sequence of b if there exists such index j that $a_i = b_i$ for all $i < j$, but $a_j < b_j$.

Examples

stdin	stdout
1 3 0 0 3 0 4 0	1 2 3

Problem H. Monster Hunter

Input file:	`stdin`
Output file:	`stdout`
Time limit:	4 seconds
Memory limit:	512 megabytes

Little Q is fighting against scary monsters in the game "Monster Hunter". The battlefield consists of n intersections, labeled by $1, 2, ..., n$, connected by $n-1$ bidirectional roads. Little Q is now at the 1-th intersection, with X units of health point(HP).

There is a monster at each intersection except 1. When Little Q moves to the k-th intersection, he must battle with the monster at the k-th intersection. During the battle, he will lose a_i units of HP. And when he finally beats the monster, he will be awarded b_i units of HP. Note that when HP becomes negative(< 0), the game will over, so never let this happen. There is no need to have a battle at the same intersection twice because monsters do not have 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 $T(1 \leq T \leq 2000)$, denoting the number of test cases.

In each test case, there is one integer $n(2 \leq n \leq 100000)$ in the first line, denoting the number of intersections.

For the next $n-1$ lines, each line contains two integers $a_i, b_i(0 \leq a_i, b_i \leq 10^9)$, describing monsters at the $2, 3, ..., n$-th intersection.

For the next $n-1$ lines, each line contains two integers u and v, denoting a bidirectional road between the u-th intersection and the v-th intersection.

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

Output

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

Examples

stdin	stdout
1 4 2 6 5 4 6 2 1 2 2 3 3 4	3

Problem I. Random Sequence

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	512 megabytes

There is a positive integer sequence $a_1, a_2, ..., a_n$ with some unknown positions, denoted by "`0`". Little Q will replace each "`0`" by a random integer within the range $[1, m]$ equiprobably. After that, he will calculate the value of this sequence using the following formula :

$$\prod_{i=1}^{n-3} v[\gcd(a_i, a_{i+1}, a_{i+2}, a_{i+3})]$$

Little Q is wondering what is the expected value of this sequence. Please write a program to calculate the expected value.

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 2 integers $n, m(4 \leq n \leq 100, 1 \leq m \leq 100)$ in the first line, denoting the length of the sequence and the bound of each number.

In the second line, there are n integers $a_1, a_2, ..., a_n(0 \leq a_i \leq m)$, denoting the sequence.

In the third line, there are m integers $v_1, v_2, ...v_m(1 \leq v_i \leq 10^9)$, denoting the array v.

Output

For each test case, print a single line containing an integer, denoting the expected value. If the answer is $\frac{A}{B}$, please print $C(0 \leq C < 10^9 + 7)$ where $A \equiv C \times B \pmod{10^9 + 7}$.

Examples

stdin	stdout
2 6 8 4 8 8 4 6 5 10 20 30 40 50 60 70 80 4 3 0 0 0 0 3 2 4	8000 3

Problem J. Rectangle Radar Scanner

Input file:	`stdin`
Output file:	`stdout`
Time limit:	10 seconds
Memory limit:	512 megabytes

There are n houses on the ground, labeled by 1 to n. The i-th house is located at (i, y_i), and there is a spy transmitter with energy w_i inside the i-th house.

Little Q has a rectangle radar scanner, which can find all the transmitters within the range $[xl, xr] \times [yl, yr]$. That means a transmitter located at (x, y) can be found if $xl \leq x \leq xr$ and $yl \leq y \leq yr$.

Your task is to achieve the scanner efficiently.

Given m queries xl_i, xr_i, yl_i, yr_i, for each query, please find all the transmitters within the range, then report the product of their energy and the maximum/minimum energy among them.

To reduce the large input, we will use the following generator. The numbers $a_0, b_0, c_0, d_0, p, q, r$ and MOD are given initially. The values $a_i, b_i, c_i, d_i, xl_i, xr_i, yl_i, yr_i$ are then produced as follows :

$$
\begin{aligned}
a_i &= (p \times a_{i-1} + q \times b_{i-1} + r) \bmod MOD \\
b_i &= (p \times b_{i-1} + q \times a_{i-1} + r) \bmod MOD \\
c_i &= (p \times c_{i-1} + q \times d_{i-1} + r) \bmod MOD \\
d_i &= (p \times d_{i-1} + q \times c_{i-1} + r) \bmod MOD \\
xl_i &= \min(a_i \bmod n, b_i \bmod n) + 1 \\
xr_i &= \max(a_i \bmod n, b_i \bmod n) + 1 \\
yl_i &= \min(c_i \bmod n, d_i \bmod n) + 1 \\
yr_i &= \max(c_i \bmod n, d_i \bmod n) + 1
\end{aligned}
$$

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 an integer $n(1 \leq n \leq 100000)$ in the first line, denoting the number of houses.

In the next n lines, each line contains 2 integers $y_i, w_i(1 \leq y_i \leq n, 1 \leq w_i \leq 10^9)$, describing a house.

Then in the next line, there are 10 integers $m, a_0, b_0, c_0, d_0, p, q, r, MOD, k$, describing the queries. It is guaranteed that $1 \leq m \leq 10^6$ and $5 \leq a_0, b_0, c_0, d_0, p, q, r, MOD, k \leq 10^9$.

Output

Since the output file may be very large, let's denote $prod_i$ as the product of of the i-th query, max_i as the maximum energy of the i-th query, and denote min_i as the minimum energy of the i-th query. Note that when there are no avaliable transmitters, then $prod_i = max_i = min_i = 0$.

For each test case, you need to print a single line containing an integer $answer$, where :

$$answer \quad = \quad \sum_{i=1}^{m}((prod_i \bmod k) \oplus max_i \oplus min_i)$$

Note that "$\oplus$" denotes binary XOR operation.

Examples

stdin	stdout
1 5 2 6 1 8 5 2 4 9 2 4 3 5 6 7 8 9 8 7 998244353 10007	68

Problem K. Transport Construction

Input file:	`stdin`
Output file:	`stdout`
Time limit:	5 seconds
Memory limit:	512 megabytes

There are n cities in Byteland, labeled by 1 to n. The i-th city is located at (x_i, y_i).

The Transport Construction Authority of Byteland is planning to open several bidirectional flights. Opening flight between the i-th city and the j-th city will cost $x_i \times x_j + y_i \times y_j$ dollars.

The Transport Construction Authority is now searching for the cheapest way to connect all of n cities, so that every pair of different cities are connected by these flights directly or indirectly. Please write a program to find the cheapest way.

Input

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

In each test case, there is an integer $n(2 \leq n \leq 100000)$ in the first line, denoting the number of cities in Byteland.

For the next n lines, each line contains 2 integers $x_i, y_i(1 \leq x_i, y_i \leq 10^6)$, denoting the coordinate of the i-th city. Note that different cities may have the same coordinate because they are too close to each other.

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

Output

For each test case, print a single line containing an integer, denoting the minimum total cost.

Examples

stdin	stdout
1 3 2 4 3 1 5 2	27

Problem L. Visual Cube

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 second
Memory limit:	512 megabytes

Little Q likes solving math problems very much. Unluckily, however, he does not have good spatial ability. Everytime he meets a 3D geometry problem, he will struggle to draw a picture.

Now he meets a 3D geometry problem again. This time, he doesn't want to struggle any more. As a result, he turns to you for help.

Given a cube with length a, width b and height c, please write a program to display the cube.

Input

The first line of the input contains an integer $T(1 \le T \le 50)$, denoting the number of test cases.

In each test case, there are 3 integers $a, b, c(1 \le a, b, c \le 20)$, denoting the size of the cube.

Output

For each test case, print several lines to display the cube. See the sample output for details.

Examples

<table>
<tr><th>stdin</th><th>stdout</th></tr>
<tr><td>2
1 1 1
6 2 4</td><td>..+-+
././|
+-+.+
|.|/.
+-+..
....+-+-+-+-+-+-+
.../././././././|
..+-+-+-+-+-+-+.+
./././././././|/|
+-+-+-+-+-+-+.+.+
|.|.|.|.|.|.|/|/|
+-+-+-+-+-+-+.+.+
|.|.|.|.|.|.|/|/|
+-+-+-+-+-+-+.+.+
|.|.|.|.|.|.|/|/.
+-+-+-+-+-+-+.+..
|.|.|.|.|.|.|/...
+-+-+-+-+-+-+....</td></tr>
</table>

Problem M. Walking Plan

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2 seconds
Memory limit:	512 megabytes

There are n intersections in Bytetown, connected with m one way streets. Little Q likes sport walking very much, he plans to walk for q days. On the i-th day, Little Q plans to start walking at the s_i-th intersection, walk through at least k_i streets and finally return to the t_i-th intersection.

Little Q's smart phone will record his walking route. Compared to stay healthy, Little Q cares the statistics more. So he wants to minimize the total walking length of each day. Please write a program to help him find the best route.

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 2 integers $n, m(2 \leq n \leq 50, 1 \leq m \leq 10000)$ in the first line, denoting the number of intersections and one way streets.

In the next m lines, each line contains 3 integers $u_i, v_i, w_i(1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq w_i \leq 10000)$, denoting a one way street from the intersection u_i to v_i, and the length of it is w_i.

Then in the next line, there is an integer $q(1 \leq q \leq 100000)$, denoting the number of days.

In the next q lines, each line contains 3 integers $s_i, t_i, k_i(1 \leq s_i, t_i \leq n, 1 \leq k_i \leq 10000)$, describing the walking plan.

Output

For each walking plan, print a single line containing an integer, denoting the minimum total walking length. If there is no solution, please print "`-1`".

Examples

stdin	stdout
2 3 3 1 2 1 2 3 10 3 1 100 3 1 1 1 1 2 1 1 3 1 2 1 1 2 1 1 2 1 1	111 1 11 -1