Problem A. Equipment Upgrade

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

Little Q is playing an RPG game. In this game, the weapon can be upgraded. Initially, the weapon is at level 0, and the upper bound of the level is n. Assume the current level is i $(0 \le i < n)$, Little Q can pay c_i coins to upgrade the weapon, the next level will be $i+1$ with probability p_i, and will be $i-j$ $(1 \le j \le i)$ with probability $(1-p_i)\frac{w_j}{\sum_{k=1}^{i} w_k}$.

Though Little Q is very rich, he is still wondering the expected number of coins for him to upgrade the weapon from level 0 to level n. Please write a program to help him.

Input

The first line contains a single integer T $(1 \le T \le 300)$, the number of test cases. For each test case:

The first line contains a single integer n $(2 \le n \le 100\,000)$, denoting the upper bound of the level.

The i-th $(1 \le i \le n)$ of the following n lines contains two integers P_{i-1} and c_{i-1} $(1 \le P_{i-1}, c_{i-1} \le 100)$, describing the success probability and the cost for level $i-1$. Here $p_{i-1} = \frac{P_{i-1}}{100}$. It is guaranteed that $P_0 = 100$.

The next line contains $n-1$ integers $w_1, w_2, \ldots, w_{n-1}$ $(1 \le w_i \le 100)$.

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

Output

For each test case, output a single line containing an integer, denoting the expected number of coins to level n.

More precisely, if the answer is $\frac{p}{q}$, you should output the minimum non-negative integer r such that $q \cdot r \equiv p \pmod{998\,244\,353}$. You may safely assume that such r always exists in all test cases.

Example

standard input	standard output
2 2 100 1 50 5 1 3 100 1 70 2 50 3 2 3	12 228170152

Problem B. Boss Rush

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

Finally, Little Q gets his weapon at level 10^5 in the RPG game, now he is trying to beat the boss as soon as possible. The boss has H units of health point (HP), Little Q needs to cause at least H units of damage to beat the boss.

Little Q has learnt n skills, labeled by $1, 2, \dots, n$. Each skill can not be used multiple times, because there is not enough time for Little Q to wait for the skill to cool down. Assume Little Q uses the i-th skill at the x-th frame, the actor controlled by him will take t_i frames to perform, which means Little Q will not be allowed to use other skills until the $(x + t_i)$-th frame. The length of the damage sequence of the i-th skill is len_i, which means the skill will cause $d_{i,j}$ $(0 \le j < len_i)$ units of damage at the $(x + j)$-th frame if Little Q uses the i-th skill at the x-th frame. Note that len_i can be greater than t_i, for example, the burning skill can burn the boss for a long period, but takes a little time to cast the fire.

The game starts at the 0-th frame. Your task is to help Little Q beat the boss as soon as possible, or determine Little Q can't beat the boss using all the skills at most once.

Input

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

The first line contains two integers n and H $(1 \le n \le 18,\ 1 \le H \le 10^{18})$, denoting the number of skills and the HP of the boss.

For each skill, the first line contains two integers t_i and len_i $(1 \le t_i, len_i \le 100\,000)$, the second line contains len_i integers $d_{i,0}, d_{i,1}, \dots, d_{i,len_i-1}$ $(1 \le d_{i,j} \le 10^9)$.

It is guaranteed that the sum of all len_i is at most $3\,000\,000$, and there are at most 5 cases such that $n > 10$.

Output

For each test case, output a single line containing an integer, denoting the first frame to beat the boss. If it is impossible to beat the boss, please print "`-1`" instead.

Example

standard input	standard output
3 1 100 5 3 50 60 70 2 100 2 3 40 40 100 100 2 20 5 1 1000 1 1 999	1 2 -1

Problem C. Cyber Language

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

You may have ever seen the following words written in cyber language: "`KK`", "`DDW`", "`SMWY`", which means "`kan kan`", "`dai dai wo`", "`shen me wan yi`", respectively.

To translate a Chinese sentence into cyber language, you need to find the first letter of each Chinese character, and write it in upper-case.

You will be given several Chinese sentences, please translate them into the cyber language described above.

Input

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

The only line contains several Chinese characters written in lower-case, separated by exactly one space. It is guaranteed that the length of each line is at least 1 and at most 50.

Output

For each test case, output a single line containing a string, denoting the corresponding word written in the cyber language.

Example

standard input	standard output
3 kan kan dai dai wo shen me wan yi	KK DDW SMWY

Problem D. Divide the Sweets

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

To celebrate Children's Day, a mother of k kids comes to the shop to buy sweets. There are n boxes of sweets in the shop, the i-th box has w_i sweets. The mother will choose which boxes to buy, and will divide all the bought boxes among the k kids. Each kid will receive several (maybe zero) boxes, and will count the total number of sweets he receives. Assume the i-th kid receives c_i sweets, the unfairness is defined as $\sum_{i=1}^{k} c_i^2$. To make all the kids happy, the mother will always divide boxes to minimize the unfairness. Note that she can not open any box to adjust the number of sweets inside it.

The mother is wondering which boxes to buy. She can not make her final decision. Please write a program to help the mother try all possible $2^n - 1$ non-empty subsets of boxes to buy, and figure out the minimum unfairness for each subset. Note that you only need to report the sum of the minimum unfairness among all $2^n - 1$ non-empty subsets.

Input

The first line contains a single integer T $(1 \le T \le 50)$, the number of test cases. For each test case:

The first line contains two integers n and m $(1 \le m \le n \le 20,\ n + m \le 23)$, denoting the number of boxes and the parameter m.

The second line contains n integers $w_1, w_2, \ldots, w_n$ $(1 \le w_i \le 50\,000)$, denoting the number of sweets in each box.

It is guaranteed that there are at most 2 cases such that $n > 12$.

Output

For each test case, output m lines, the i-th $(1 \le i \le m)$ of which containing an integer, denoting the answer when $k = i$. Note that the answer may be extremely large, so please print it modulo $998\,244\,353$ instead.

Example

standard input	standard output
1 5 4 1 2 3 4 5	2240 1180 930 884

Problem E. Spanning Tree Game

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

Alice and Bob are playing a game on an undirected graph with n vertices and m edges. The vertices are labeled by $1, 2, \ldots, n$. The edges are labeled by $1, 2, \ldots, m$. The i-th edge connects the u_i-th vertex and the v_i-th vertex directly, and its weight will be chosen from the given two values a_i and b_i.

First, Alice needs to assign weights to all the m edges such that there are exactly k edges whose weights are taken from a while the weights of other $m - k$ edges are taken from b. Then, Bob needs to choose exactly $n - 1$ edges from the graph such that every pair of different vertices are connected by these $n - 1$ edges directly or indirectly.

The final score of the game is equal to the total weights of the $n - 1$ edges chosen by Bob. Alice wants to maximize the score while Bob wants to minimize it. Please write a program to predict the final score for $k = 0, 1, 2, \ldots, m$ if both of the players play optimally.

Input

The first line contains a single integer T ($1 \leq T \leq 20$), the number of test cases. For each test case:

The first line contains two integers n and m ($2 \leq n \leq 9$, $n-1 \leq m \leq 30$), denoting the number of vertices and the number of edges.

Each of the following m lines contains four integers u_i, v_i, a_i and b_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $1 \leq a_i, b_i \leq 1\,000\,000$), describing an edge.

It is guaranteed that the graph is connected.

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 final score when $k = i - 1$.

Example

standard input	standard output
1 3 3 1 2 4 6 1 3 2 7 2 3 3 5	11 9 7 5

Problem F. Dusk Moon

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

You are given n points $p_1, p_2, \ldots, p_n$ on the 2D plane. You need to perform q operations. Each operation is one of the following:

- "`1` k x y" ($1 \le k \le n$, $1 \le x, y \le 10^8$): Change the coordinate of the point p_k into (x, y).
- "`2` l r" ($1 \le l \le r \le n$): Find the minimum non-negative integer R such that you can cover all the points $p_l, p_{l+1}, \ldots, p_r$ using a single circle whose radius is R. Note that a point is considered to be covered if and only if it is inside the circle or on the border of the circle.

Input

The first line contains a single integer T ($1 \le T \le 3$), 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 points and the number of operations.

In the next n lines, the i-th line contains two integers x_i and y_i ($1 \le x_i, y_i \le 10^8$), describing the coordinate of p_i.

Each of the next q lines describes an operation in formats described in the statement above.

It is guaranteed that all the values of x_i, y_i, x, y 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 minimum radius.

Example

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

Problem G. Shallow Moon

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

There are $m \times m$ cells on a grid, the top-left cell is at $(1,1)$ while the bottom-right cell is at (m,m). Initially, all the cells were colored white. Little Q has drawn n black $w \times h$ rectangles on the grid. For the i-th rectangle, Little Q chose a cell at (a_i, b_i), and painted all the cells (x, y) black, where $a_i \leq x \leq a_i + w - 1$ and $b_i \leq y \leq b_i + h - 1$.

After Little Q finished all of his work, he is now wondering how many pairs of white cells are 4-connected. Please write a program to calculate:

$$\sum_{(i,j)|1 \leq i,j \leq m,\ (i,j)\ is\ white} f(i,j)$$

Here $f(i,j)$ is the number of white cells that are 4-connected with (i,j), including (i,j) itself.

Two cells are considered adjacent if and only if they share a common side. Two white cells (i,j), (x,y) are considered 4-connected if and only if there exists a sequence of white cells $c_1, c_2, \dots, c_k$ such that:

- $c_1 = (i,j)$.
- $c_k = (x,y)$.
- c_i and c_{i+1} are adjacent for all i $(1 \leq i < k)$.

Input

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

The first line contains four integers n, m, w and h $(1 \leq n \leq 100\,000,\ 1 \leq w, h \leq m \leq 10^9)$, denoting the number of rectangles, the size of the grid, and the size of each rectangle.

Each of the next n lines contains two integers a_i and b_i $(1 \leq a_i \leq m-w+1,\ 1 \leq b_i \leq m-h+1)$, denoting a rectangle.

It is guaranteed that the sum of all n is at most $2\,000\,000$.

Output

For each test case, print a single line containing an integer denoting the answer. Note that the answer may be extremely large, so please print it modulo 2^{64} instead.

Example

standard input	standard output
1 4 6 2 2 1 3 2 2 3 5 4 1	201

Problem H. Laser Alarm

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

The museum in Byteland has plenty of jewels on display, secured by n laser alarms. Each laser alarm can be considered as a segment in the 3D space. In this task, your job is to test the quality of the laser alarm system. You need to find a plane such that it touches the most laser alarms. Note that if the plane touches the endpoint of a segment, it should also be counted.

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 ($1 \le n \le 50$), denoting the number of laser alarms.

Each of the following n lines contains six integers x_i, y_i, z_i, x'_i, y'_i and z'_i ($1 \le x_i, y_i, z_i, x'_i, y'_i, z'_i \le 100$), describing a segment that connects (x_i, y_i, z_i) and (x'_i, y'_i, z'_i). It is guaranteed that the two endpoints of each segment do not coincide.

Output

For each test case, output a single line containing an integer, denoting the maximum possible number of laser alarms that can be touched.

Example

standard input	standard output
1 4 1 1 1 1 1 2 1 1 10 1 1 11 1 10 1 1 10 2 10 1 1 10 1 2	3

Problem I. Package Delivery

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

Little Q likes online shopping very much. In the next 10^9 days, there will be n packages delivered to the post office in total. Let's label the next 10^9 days as day 1, day 2, ..., day 10^9 respectively. For the i-th package, it will arrive at the post office at day l_i, and the deadline to take it back home is day r_i, which means Little Q can take it back home at day x if and only if $l_i \leq x \leq r_i$.

Every time Little Q comes to the post office, he can take at most k packages together back home at the same time. Note that Little Q can go to the post office multiple times during a single day. Please help Little Q determine how to take these n packages back home such that the number of times he will go to the post office is minimized.

Input

The first line contains a single integer T ($1 \leq T \leq 3\,000$), the number of test cases. For each test case:

The first line contains two integers n and k ($1 \leq k \leq n \leq 100\,000$), denoting the number of packages and the number of packages Little Q can carry at the same time.

Each of the following n lines contains two integers l_i and r_i ($1 \leq l_i \leq r_i \leq 10^9$), describing a package.

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

Output

For each test case, output a single line containing an integer, denoting the minimum possible number of times that Little Q will go to the post office.

Example

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

Problem J. Range Reachability Query

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

You are given a directed acyclic graph with n vertices and m edges. The vertices are labeled by $1, 2, \ldots, n$, and the edges are labeled by $1, 2, \ldots, m$.

You will be given q queries. In the i-th query, you will be given four integers u_i, v_i, l_i and r_i ($1 \le l_i \le r_i \le m$). You need to answer whether vertex u_i can reach vertex v_i when only edges labeled by k ($l_i \le k \le r_i$) are available.

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 three integers n, m and q ($2 \le n \le 50\,000$, $1 \le m \le 100\,000$, $1 \le q \le 50\,000$), denoting the number of vertices, the number of edges, and the number of queries.

Each of the following m lines contains two integers u_i and v_i ($1 \le u_i < v_i \le n$), denoting a directed edge from vertex u_i to vertex v_i.

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

Output

For each query, print a single line. If vertex u_i can reach vertex v_i when only edges labeled by k ($l_i \le k \le r_i$) are available, print "YES". Otherwise, print "NO".

Example

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

Problem K. Taxi

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

There are n towns in Byteland, labeled by $1, 2, \ldots, n$. The i-th town's location is (x_i, y_i). Little Q got a taxi VIP card, he can use the VIP card to cut down the taxi fare. Formally, assume Little Q is at (x', y'), if he calls a taxi to drive him to the k-th town, the VIP card will reduce $\min(|x' - x_k| + |y' - y_k|, w_k)$ dollars.

Little Q wants to make full use of his VIP card. He will give you q queries, in each query you will be given his location, and you need to choose a town such that the VIP card will reduce the most taxi fare.

Input

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

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

Each of the following n lines contains three integers x_i, y_i and w_i $(1 \leq x_i, y_i, w_i \leq 10^9)$, describing a town.

Each of the following q lines contains two integers x' and y' $(1 \leq x', y' \leq 10^9)$, describing a query.

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

Output

For each query, print a single line containing an integer, denoting the maximum possible reduced taxi fare.

Example

standard input	standard output
1 3 4 1 5 7 5 1 6 2 3 9 1 5 2 2 4 3 10 10	6 4 5 9

Problem L. Two Permutations

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

There are two permutations $P_1, P_2, \ldots, P_n$, $Q_1, Q_2, \ldots, Q_n$ and a sequence R. Initially, R is empty. While at least one of P and Q is non-empty, you need to choose a non-empty array (P or Q), pop its leftmost element, and attach it to the right end of R. Finally, you will get a sequence R of length $2n$.

You will be given a sequence S of length $2n$, please count the number of possible ways to merge P and Q into R such that $R = S$. Two ways are considered different if and only if you choose the element from different arrays in a step.

Input

The first line contains a single integer T $(1 \leq T \leq 300)$, the number of test cases. For each test case:

The first line contains a single integer n $(1 \leq n \leq 300\,000)$, denoting the length of each permutation.

The second line contains n distinct integers $P_1, P_2, \ldots, P_n$ $(1 \leq P_i \leq n)$.

The third line contains n distinct integers $Q_1, Q_2, \ldots, Q_n$ $(1 \leq Q_i \leq n)$.

The fourth line contains $2n$ integers $S_1, S_2, \ldots, S_{2n}$ $(1 \leq S_i \leq n)$.

It is guaranteed that the sum of all n is at most $2\,000\,000$.

Output

For each test case, output a single line containing an integer, denoting the number of possible ways. Note that the answer may be extremely large, so please print it modulo $998\,244\,353$ instead.

Example

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