Problem A. Magma Cave

Input file:	`standard input`
Output file:	`standard output`
Time limit:	12 seconds
Memory limit:	512 megabytes

Little Q is researching an active volcano. There are n caves inside the volcano, labeled by $1, 2, \ldots, n$. At the very beginning, before the first volcanic activity event, there is no magma path between these caves. You will be given q operations, each operation is one of the following:

- "1 u v w" ($1 \leq u, v \leq n$, $u \neq v$, $1 \leq w \leq q$): A volcanic activity event comes such that a new magma path between the u-th cave and the v-th cave occurs, whose length is w. Here w is used for identifying the magma path, so w will always be pairwise different.
- "2 k w" ($1 \leq k < n$, $1 \leq w \leq q$): Assume it is a undirected graph with n vertices, each magma path denoting an edge, Little Q is wondering whether there exists a spanning tree whose k-th shortest edge is of length w. You are the partner of Little Q, please write a program to answer his question.

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 ($2 \leq n \leq 50\,000$, $1 \leq q \leq 200\,000$), denoting the number of caves and the number of operations.

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

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

Output

For each question, print a single line. If it is possible, print "`YES`", otherwise print "`NO`".

Example

standard input	standard output
2 3 7 1 1 2 1 2 1 1 1 2 3 5 1 1 3 4 2 2 4 2 2 5 2 2 3 2 4 1 1 2 1 1 1 2 2 2 1 1 2 1 2	NO YES YES NO YES YES

Problem B. King's Ruins

Input file:	`standard input`
Output file:	`standard output`
Time limit:	15 seconds
Memory limit:	512 megabytes

The first king of Byteland was buried with his n knights a thousand years ago. In the ruins, the archaeologists have found n stone tablets in a row, labeled by $1, 2, \ldots, n$ from left to right. Each stone tablet belongs to a knight. On the surface of the k-th stone tablet, five numbers are describing the ability of the k-th knight in five aspects:

- The wind ability w, denoting how fast the knight is.
- The guard ability g, denoting how many attacks can be defended by the knight.
- The ice ability i, denoting the power of ice crystal cast by the knight.
- The flame ability f, denoting the power of fire cast by the knight.
- The light ability l, denoting the power of thunder cast by the knight.

Little Q is visiting the stone tablets from left to right, according to the labels from 1 to n. After visiting a stone tablet, before moving right to the next one, he can choose to take a photo with it or do nothing. Little Q estimates the value of each stone tablet, he wants to maximize the total value that he takes photos with, and the knight corresponding to the next photo is always not weaker than the current one. Here the x-th knight is considered not to be weaker than the y-th knight if and only if $w_x \geq w_y$, $g_x \geq g_y$, $i_x \geq i_y$, $f_x \geq f_y$ and $l_x \geq l_y$.

There are so many stone tablets that Little Q can not determine which to take photos with. Please write a program to help him.

Input

The first line contains a single integer T ($1 \leq T \leq 5$), the number of test cases. For each test case:

The first line contains a single integer n ($1 \leq n \leq 50\,000$), denoting the number of stone tablets.

In the next n lines, the k-th line contains six integers w_k, g_k, i_k, f_k, l_k and v_k ($1 \leq w_k, g_k, i_k, f_k, l_k \leq n$, $1 \leq v_k \leq 10\,000$), describing the k-th stone tablet, v_k denoting the value of the photo with it.

Output

For each test case, output n lines, the k-th ($1 \leq k \leq n$) of which containing an integer, denoting the maximum total value when the last photo is taken with the k-th stone tablet.

Example

standard input	standard output
1 4 1 2 1 2 1 30 2 1 2 1 2 40 3 3 3 3 3 50 2 3 3 2 4 100	30 40 90 140

Problem C. Leshphon

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	512 megabytes

Leshphon was a peaceful and beautiful village in Byteland until the nasty disease came here. There are n districts in Leshphon, connected by m directed roads such that you can reach every other district from any district via these directed roads. In other words, it is strongly connected.

To control the sources of infection and cut off the channels of transmission, the government of Byteland decides to close exactly three directed roads in Leshphon, such that the village is not strongly connected anymore. There may be many possible solutions, can you figure out the total number of them?

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 ($3 \leq n \leq 50$, $3 \leq m \leq n(n-1)$), denoting the number of districts and the number of roads.

Each of the following m lines contains two integers u_i and v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$), describing a directed road from the u_i-th district to the v_i-th district.

It is guaranteed that the village is strongly connected, and all the given m roads are pairwise different.

Output

For each test case, output a single line containing an integer, denoting the number of solutions.

Example

standard input	standard output
1 4 7 1 2 2 3 3 4 4 1 1 3 2 4 3 1	34

Problem D. Chaos Begin

Input file:	`standard input`
Output file:	`standard output`
Time limit:	15 seconds
Memory limit:	512 megabytes

Long long ago, there were n points $a_1, a_2, \ldots, a_n$ on the 2D plane. The world keeps stable for a long time. However, it begins to be chaotic recently when another n points $b_1, b_2, \ldots, b_n$ appeared, where $b_i = a_i + (\Delta x, \Delta y)$. And now, these $2n$ points have already lost their identifiers.

You are given these $2n$ points in an arbitrary order, you need to figure out all the possible $(\Delta x, \Delta y)$ to help the world recover from chaos.

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 of the input contains a single integer n ($1 \le n \le 50\,000$), denoting the number of initial points.

In the next $2n$ lines, the i-th line contains two integers x_i and y_i ($|x_i|, |y_i| \le 10^8$), describing the coordinate of a current point.

It is guaranteed that the x-coordinate and y-coordinate of each initial point are chosen uniformly at random from integers in $[-v, v]$, where v is chosen in $[10^7, 10^8]$. The randomness condition does not apply to the sample test case, but your solution must pass the sample as well.

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

Output

For each test case, first output a single line containing an integer k, denoting the number of possible $(\Delta x, \Delta y)$. Then output k lines, each line contains two integers Δx and Δy. It is guaranteed that $k \ge 1$, and when $k \ge 2$, you should print the answers in ascending order of Δx, and then in ascending order of Δy in case of a tie.

Example

standard input	standard output
1 3 1 2 3 4 8 9 7 8 6 7 2 3	2 -5 -5 5 5

Problem E. Out of Control

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

There is a cloud service API to help user store history timestamps. The data structure for each user is initially an empty stack. Every time you post a request to the API with an integer x, denoting the current timestamp, the service will append x to the end of the stack. When x is less than the previous timestamp stored in the stack, the service will think the input is wrong, and will append the previous timestamp instead of x.

You have posted the API for n times, your request timestamp is x_i in the i-th call. However, the network is out of control. The service may receive your requests in any arbitrary order, and may even miss some requests. Knowing this issue, you are asking for the on-call engineer to have a look at your stack in the database. Assume the service received exactly k requests, how many possible distinct stacks will it be?

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 of the input contains a single integer n ($1 \le n \le 3\,000$), denoting the total number of requests.

The second line contains n integers $x_1, x_2, \ldots, x_n$ ($1 \le x_i \le 10^9$), denoting the timestamp of each request.

It is guaranteed that the sum of all n is at most $30\,000$.

Output

For each test case, output n lines, the i-th ($1 \le i \le n$) of which containing an integer, denoting the number of distinct stacks when $k = i$. Note that the answer may be extremely large, so please print it modulo $10^9 + 7$ instead.

Example

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

Note

In the first example:

- When $k = 1$, the stack can be $[1]$, $[2]$ or $[3]$.
- When $k = 2$, the stack can be $[1, 2]$, $[1, 3]$, $[2, 2]$, $[2, 3]$ or $[3, 3]$.
- When $k = 3$, the stack can be $[1, 2, 3]$, $[1, 3, 3]$, $[2, 2, 3]$, $[2, 3, 3]$ or $[3, 3, 3]$.

In the second example:

- When $k = 1$, the stack can be $[2]$ or $[3]$.
- When $k = 2$, the stack can be $[2, 3]$ or $[3, 3]$.
- When $k = 3$, the stack can be $[2, 3, 3]$ or $[3, 3, 3]$.

Problem F. Dragon Seal

Input file:	`standard input`
Output file:	`standard output`
Time limit:	12 seconds
Memory limit:	512 megabytes

An immortal dragon attacked the Byteland. After days of battle, the heroes finally beat the dragon. However, the dragon is immortal such that it can not be killed, so the magicians in Byteland decide to seal the dragon using ancient black magic.

The dragon is scheduled to be sealed under an undirected magic tree with n vertices. In each vertice of the tree, there are two gems. Each gem has its magic value m and power value p. The black magic requires removing exactly one gem from each vertex such that there will be exactly one gem remaining at each vertex. The total power is the sum of the power values of all the gems that remained in the tree.

To escape from sealed, the dragon has a chance to attack. It will choose a target vertex on the tree, then select some vertices among it and its neighbor vertices, and finally start an attack on them. If at least one vertex is attacked, and the bitwise XOR sum among magic values of all the gems in attacked vertices is equal to zero, the tree will be broken, and the dragon escapes. The magicians must never let this happen.

Please write a program to help them determine how to choose gems such that the dragon can never escape, and the total power is maximized.

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 a single integer n ($2 \leq n \leq 60$), denoting the number of vertices.

In the next n lines, the i-th line contains four integers m_i, p_i, m'_i and p'_i ($1 \leq m_i, m'_i < 2^{64}$, $1 \leq p_i, p'_i \leq 10^7$), describing the two gems in the i-th vertex.

Each of the following $n-1$ lines contains two integers u_i and v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$), describing an undirected edge between the u_i-th vertex and the v_i-th vertex.

It is guarenteed that the input is a tree.

Output

For each test case, output a single line containing an integer, denoting the total power to seal the dragon. When it is impossible to seal the dragon, please print "`-1`" instead.

Example

standard input	standard output
2 2 3 1 3 1 3 2 3 2 1 2 3 1 3 2 1 2 2 4 3 1 3 3 2 1 2 2 3	-1 8

Problem G. Casino Royale

Input file:	`standard input`
Output file:	`standard output`
Time limit:	7 seconds
Memory limit:	512 megabytes

In Casino Royale, one of the famous games is played on a decimal string $s_1 s_2 \ldots s_n$, where $s_i \in \{\text{'1'}, \text{'2'}\}$. The two players move in turns, the first player moves first. In each move, the current player selects either the first bit or the last bit of the string, adds it to this player's score, and removes it from the string. The game ends when the string is empty. Both players want to maximize their scores, the one with the maximum score wins.

Before the first move of a game, the observers can see a portion of bits in the string, then make a bet on who will win in the end. Little Q is the owner of the Casino Royale. He is interested in how to predict the result of the above game such that he can earn a lot of money by predicting.

You will be given the initial string with some bits hidden. Little Q will then give you q queries. In each query, you will be given two integers A and B, denoting the final scores of the first player and the second player respectively, you need to report the number of distinct initial strings that will lead to this result if both players play optimally.

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 ($1 \leq n \leq 50$, $1 \leq q \leq (2n+1)(2n+1)$), denoting the length of the initial string and the number of queries.

The second line contains a string s of length n ($s_i \in \{\text{'1'}, \text{'2'}, \text{'?'}\}$), denoting the initial string that can be seen by the observers. Here '?' denotes the value of the corresponding bit is hidden.

Each of the following q lines contains two integers A and B ($0 \leq A, B \leq 2n$), denoting a query.

Output

For each query, print a single line containing an integer, denoting the number of possible initial strings.

Example

standard input	standard output
2 3 2 121 2 2 1 3 2 4 ?? 1 1 2 2 2 1 1 2	1 0 1 1 2 0

Problem H. Teyberrs

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

Teyberrs is a paradise for birds to live in. Assume you are a bird in Teyberrs, you are now flying somewhere like the game "Flappy Bird". You start flying at $(0, s)$, and every time when you are at $(x-1, y)$ $(1 \le x \le n)$, you must fly to either $(x, y-1)$ with cost a_x or $(x, y+1)$ with cost b_x. Like the map in "Flappy Bird", you can not hit obstacles at (x, y) where $y < l_x$ or $y > r_x$.

You will be given q queries. In each query, you will be given two integers x and y. Assume your target is at (x, y), can you find the path with the minimum cost, or determine it is impossible?

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 of the input contains three integers n, q and s $(1 \le n, q \le 200\,000,\ 1 \le s \le n)$, denoting the size of the map, the number of queries, and the start point.

In the next n lines, the i-th line contains four integers a_i, b_i, l_i and r_i $(1 \le a_i, b_i \le 10^9,\ 1 \le l_i \le r_i \le n)$.

In the next q lines, the i-th line contains two integers x and y $(1 \le x, y \le n)$, describing a target point.

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

Output

For each query, print a single line containing an integer, denoting the minimum total cost. When it is impossible to reach the target, please print "`-1`" instead.

Example

standard input	standard output
1 3 9 2 1 2 1 3 3 1 2 3 4 3 1 2 1 1 1 2 1 3 2 1 2 2 2 3 3 1 3 2 3 3	1 -1 2 -1 2 -1 6 -1 -1

Problem I. Operation Hope

Input file:	`standard input`
Output file:	`standard output`
Time limit:	10 seconds
Memory limit:	512 megabytes

Little Q is playing an RPG online game. In this game, there are n characters labeled by $1, 2, \ldots, n$. The i-th character has three types of quotas:

- a_i - The maximum points of damage he can achieve in 15 seconds.
- b_i - The maximum points of damage he can achieve in 40 seconds.
- c_i - The maximum points of damage he can achieve in 120 seconds.

You are the team leader working for the new balance between these n characters, aiming at bringing hope to the weak characters. For each character, your teammates have made a plan to strengthen some skills such that the three quotas may be increased as a result. Note that it is not allowed to weaken characters, because it will make their owners upset.

To make a perfect balance, you need to accept some plans and deny others such that the gap between all the n characters is minimized. Note that a plan can only be entirely accepted or entirely denied. Here, the gap is defined as

$$\max\{\max_{1\leq i\leq n} a_i - \min_{1\leq i\leq n} a_i, \max_{1\leq i\leq n} b_i - \min_{1\leq i\leq n} b_i, \max_{1\leq i\leq n} c_i - \min_{1\leq i\leq n} c_i\}$$

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 a single integer n $(1 \leq n \leq 100\,000)$, denoting the number of characters.

In the next n lines, the i-th line contains six integers a_i, b_i, c_i, a'_i, b'_i and c'_i $(1 \leq a_i \leq a'_i \leq 10^9$, $1 \leq b_i \leq b'_i \leq 10^9$, $1 \leq c_i \leq c'_i \leq 10^9)$, describing the quotas of the i-th character now and in plan.

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 optimal gap.

Example

standard input	standard output
1 2 1 1 1 2 3 5 2 4 3 7 5 8	2

Problem J. The Mine of Abyss

Input file:	`standard input`
Output file:	`standard output`
Time limit:	10 seconds
Memory limit:	512 megabytes

Recently, a huge mine with n crystals under deep ground has been detected in Byteland. The crystals are labeled by $1, 2, \ldots, n$. The weight of each crystal is not confirmed, but a range can be estimated. Specifically, the weight of the i-th crystal is an integer within the range $[a_i, b_i]$.

You are the analyzer of this mine. You will be given q operations, each operation is one of the following:

- "`1` k a b" ($1 \leq k \leq n$, $1 \leq a \leq b \leq 10^9$): The k-th crystal is re-scanned. The new report shows its weight is an integer within the range $[a, b]$. The previous range is useless now.
- "`2` l r" ($l \leq l \leq r \leq n$): Assume some (maybe none or maybe all) crystals indexed in $[l, r]$ are unearthed, let's measure their total weight, how many possible total weights may we get?

Input

The first line contains a single integer T ($1 \leq T \leq 5$), the number of test cases. For each test case:

The first line contains two integers n and q ($1 \leq n, q \leq 50\,000$), denoting the number of crystals and the number of operations.

In the next n lines, the i-th line contains two integers a_i and b_i ($1 \leq a_i, b_i \leq 10^9$), denoting the weight range of the i-th crystal.

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

It is guaranteed that all the values of a_i, b_i, a, b 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 number of possible total weights.

Example

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

Note

In the first query, the total weight can be 0, 2 or 3.

In the second query, the total weight can be 0, 1, 2, 3 or 4.

Problem K. 8-bit Zoom

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

You are given a picture with size $n \times n$. You need to output the zoomed picture with the zooming rate $Z\%$ in 8-bit style, or determine the picture can not be zoomed. Here in 8-bit style, the size of the result picture is $\frac{nZ}{100} \times \frac{nZ}{100}$. A picture can not be zoomed in 8-bit style if and only if any of the following holds:

- $\frac{nZ}{100}$ is not an integer.
- In the zoomed picture, the color of some pixels can not be determined. Note that there aren't any interpolation algorithm applied in 8-bit style, so when at least two different colors are mapped into the same pixel in the resulting picture, the color of this pixel is undetermined.

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 two integers n and Z ($1 \le n \le 50$, $100 \le Z \le 200$, $Z \bmod 25 = 0$), denoting the size of the original picture and the zooming rate.

Each of the following n lines contains a string of length n, consisting of lowercase English letters. The j-th character in the i-th line denotes the color of the pixel located at (i, j).

Output

For each test case, if the picture can not be zoomed, print "`error`" in a line, otherwise print $\frac{nZ}{100}$ lines, each line contains a string of length $\frac{nZ}{100}$, denoting the resulting picture.

Example

standard input	standard output
5 2 100 ab cd 2 200 ab cd 2 125 aa aa 4 125 aaab aaaa aaaa aaaa 4 125 aaaa aaaa aaaa aaaa	ab cd aabb aabb ccdd ccdd error error aaaaa aaaaa aaaaa aaaaa aaaaa

Problem L. Noblesse Code

Input file:	`standard input`
Output file:	`standard output`
Time limit:	10 seconds
Memory limit:	512 megabytes

You will be given n noblesse code pairs $(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)$ and q queries. In each query, you will be given a pair (A, B), you need to figure out how many noblesse code pairs can be transformed from the given pair (A, B). Every time you can transform the current pair (A, B) into $(A + B, B)$ or $(A, A + B)$. You can do the transform operation for arbitrary times (or do nothing).

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 of the input contains two integers n and q $(1 \le n, q \le 50\,000)$, denoting the number of noblesse code pairs and the number of queries.

In the next n lines, the i-th line contains two integers a_i and b_i $(1 \le a_i, b_i \le 10^{18})$, describing the i-th noblesse code pair.

In the next q lines, the i-th line contains two integers A and B $(1 \le A, B \le 10^{18})$, describing the pair in the i-th 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 number of noblesse code pairs that can be transformed from the given pair. Note that two noblesse code pairs $(a_i, b_i), (a_j, b_j)$ are considered to be different if and only if $i \ne j$.

Example

standard input	standard output
2 3 4 6 9 5 3 1 1 6 3 1 2 2 1 5 3 2 2 7 14 7 14 7 7 7 14	1 0 1 1 2 2