Problem A. Total Eclipse

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

There are n cities and m bidirectional roads in Byteland. These cities are labeled by $1, 2, \ldots, n$, the brightness of the i-th city is b_i.

Magician Sunset wants to play a joke on Byteland by making a total eclipse such that the brightness of every city becomes zero. Sunset can do the following operations for arbitrary number of times:

- Select an integer k $(1 \leq k \leq n)$.
- Select k distinct cities $c_1, c_2, \ldots, c_k$ $(1 \leq c_i \leq n)$ such that they are connected with each other. In other words, for every pair of distinct selected cities c_i and c_j $(1 \leq i < j \leq k)$, if you are at city c_i, you can reach city c_j without visiting cities not in $\{c_1, c_2, \ldots, c_k\}$.
- For every selected city c_i $(1 \leq i \leq k)$, decrease b_{c_i} by 1.

Now Sunset is wondering what is the minimum number of operations he needs to do, please write a program to help him.

Input

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

For each case, the first line of the input contains two integers n and m $(1 \leq n \leq 100\,000, 1 \leq m \leq 200\,000)$, denoting the number of cities and the number of roads.

The second line of the input contains n integers $b_1, b_2, \ldots, b_n$ $(1 \leq b_i \leq 10^9)$, denoting the brightness of each city.

Each of the following m lines contains two integers u_i and v_i $(1 \leq u_i, v_i \leq n, u_i \neq v_i)$, denoting an bidirectional road between the u_i-th city and the v_i-th city. Note that there may be multiple roads between the same pair of cities.

Output

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

Example

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

Problem B. Blood Pressure Game

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

Gugulu, a JBer, who was an ACMer two years ago, comes to HDOJ taking part in the 2020 Multi-University Training Contest again. However, every time when Gugulu came to regional, he always got an Iron Medal and would consider this competition JB-like. To release his pain, Gugulu would go to the Shanghai Disneyland Park to have fun in taking the Roller Coaster. Gugulu loves the feeling with high-level blood pressure so much which makes him feel like a happy flappy bird who forgets all the Wrong Answers and Time Limit Exceededs etc.

We can regard the path of the Roller Coaster as a list of turning points with different heights which can be represented as an array $\{a_1, a_2, a_3, \dots, a_n\}$ of size n, and Gugulu's final blood pressure after the game of the Roller Coaster is counted as the sum of all absolute values of the differences between the $n-1$ pairs of adjacent array numbers, i.e. $\sum_{i=1}^{n-1} |a_i - a_{i+1}|$.

Gugulu always got Iron Medals and is always getting Iron Medals, which makes him keep taking the Roller Coaster over and over again. However, as playing more games, his threshold on the value of blood pressure which can make himself happy is keeping increasing. As a result, the Roller Coaster of Shanghai Disneyland Park can hardly meet Gugulu's needs anymore.

Therefore, Gugulu decides to rearrange the array $\{a_1, a_2, a_3, \dots, a_n\}$ in any order to make his blood pressure as high as possible. Moreover, he wants to know the largest k distinct possible values of blood pressure that can be achieved and the number of the corresponding permutations.

You, another JBer, are sure that Gugulu is clever enough to get the highest blood pressure as he can. It is very important for you to calculate the correct answer to make an appointment with a proper cardiologist in advance to save Gugulu's life. You must solve this problem! Gugulu's blood pressure is becoming out of control!

Input

The first line of the input contains the number of test cases, T $(1 \le T \le 5)$. T test cases follow.

For each test case, the first line contains two integers, n $(2 \le n \le 600)$ and k $(1 \le k \le 20)$.

Then, in second line, there are n integers $\{a_1, a_2, a_3, \dots, a_n\}$ $(1 \le a_i \le 10^6)$, denoting the n original Roller Coaster turning points. As it is an exciting Roller Coaster, it is guaranteed that all n integers in the array are pairwise different.

Output

For each test case, output k lines. For the i-th $(1 \le i \le k)$ line, print two numbers w_i and c_i, where w_i

denotes the i-th largest possible value of blood pressure that can be achieved and c_i denotes the number of corresponding permutations modulo $10^9 + 7$. Note that when the i-th largest possible value doesn't exist, print "-1" instead.

Example

standard input	standard output
1 4 8 1 3 7 9	20 2 18 4 16 2 14 8 12 2 10 4 8 2 -1

Problem C. Count on a Tree II Striking Back

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

You are given a tree with n nodes. The tree nodes are numbered from 1 to n. The color of the i-th node is col_i.

You need to perform the following operations for m times:

- "`1 x y`" ($1 \leq x, y \leq n$): Change the color of the x-th node into y.
- "`2 a b c d`" ($1 \leq a, b, c, d \leq n$): Let's denote $f(u, v)$ as the number of different colors occured on the path from u to v. You need to answer whether $f(a, b) > f(c, d)$ is true.

Input

The first line of the input contains a single integer T ($1 \leq T \leq 4$), the number of test cases.

For each case, the first line of the input contains two integers n and m ($1 \leq n \leq 500\,000$, $1 \leq m \leq 10\,000$), denoting the number of nodes and the number of operations.

The second line of the input contains n integers $col_1, col_2, \ldots, col_n$ ($1 \leq col_i \leq n$), denoting the initial color of each node.

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

Each of the next m lines describes an operation in formats described in the statement above, except that some parameters are encrypted in order to enforce online processing.

Let cnt be the number of queries that you answered "`Yes`" before in this test case. Note that cnt should be reset to 0 in each new test case. For each operation, x, y, a, b, c and d are encrypted. The actual values of x, y, a, b, c and d are $x \oplus cnt, y \oplus cnt, a \oplus cnt, b \oplus cnt, c \oplus cnt$ and $d \oplus cnt$. In the expressions above, the symbol "$\oplus$" denotes the bitwise exclusive-or operation. Also note that the constraints described in the statement above apply to the corresponding parameters only after decryption, the encrypted values are not subject to those constraints.

It is guaranteed that $f(a, b) \geq 2f(c, d)$ or $f(c, d) \geq 2f(a, b)$ always holds for each query.

Output

For each query, print a single line. If $f(a, b) > f(c, d)$ is true, print "`Yes`" else print "`No`".

Example

standard input	standard output
1 8 4 1 2 1 4 1 3 3 2 1 2 2 3 3 4 3 5 1 6 6 7 6 8 2 1 4 3 5 2 7 6 5 9 1 4 9 2 2 4 7 6	Yes No Yes

Problem D. Diamond Rush

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

There are $n \times n$ cells on a grid, the top-left cell is at $(1, 1)$ while the bottom-right cell is at (n, n). In the cell at (i, j), there are $\left(n^2\right)^{a_{i,j}}$ diamonds.

Initially, you are at $(1, 1)$, every time you can move to $(i + 1, j)$ or $(i, j + 1)$ from (i, j) without moving out of the grid. Your destination is at (n, n), so you will take exactly $2n - 2$ moves. When you are at a cell, you can take all the diamonds inside this cell, including the starting point $(1, 1)$ and the destination (n, n).

However, some cells are blocked but you don't know which cells are blocked. Please write a program to answer q queries. In each query, you will be given four integers xl, xr, yl, yr, you need to report the maximum number of diamonds that you can take without passing the cells (i, j) such that $xl \le i \le xr$ and $yl \le j \le yr$.

Input

The first line of the input contains a single integer T $(1 \le T \le 5)$, the number of test cases.

For each case, the first line of the input contains two integers n and q $(2 \le n \le 400, 1 \le q \le 200\,000)$, denoting the size of the grid and the number of queries.

Each of the following n lines contains n integers, the i-th line contains $a_{i,1}, a_{i,2}, \ldots, a_{i,n}$ $(1 \le a_{i,j} \le n^2)$, denoting the number of diamonds in each cell.

Each of the following q lines contains four integers xl, xr, yl and yr $(1 \le xl \le xr \le n, 1 \le yl \le yr \le n)$, denoting each query. It is guaranteed that you can find at least one valid path in each query.

Output

For each query, print a single line containing an integer, denoting the maximum number of diamonds that you can take. Note that the answer may be extremely large, so please print it modulo $10^9 + 7$ instead.

Example

standard input	standard output
1 2 2 2 3 1 4 1 1 2 2 2 2 1 1	276 336

Problem E. New Equipments

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

Little Q's factory recently purchased m pieces of new equipment, labeled by $1, 2, \ldots, m$.

There are n workers in the factory, labeled by $1, 2, \ldots, n$. Each worker can be assigned to no more than one piece of equipment, and no piece of equipment can be assigned to multiple workers. If Little Q assigns the i-th worker to the j-th piece of equipment, he will need to pay $a_i \times j^2 + b_i \times j + c_i$ dollars.

Now please for every k $(1 \leq k \leq n)$ find k pairs of workers and pieces of equipment, then assign workers to these pieces of equipment, such that the total cost for these k workers is minimized.

Input

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

For each case, the first line of the input contains two integers n and m $(1 \leq n \leq 50,\ n \leq m \leq 10^8)$, denoting the number of workers and the number of pieces of equipment.

Each of the following n lines contains three integers a_i, b_i and c_i $(1 \leq a_i \leq 10,\ -10^8 \leq b_i \leq 10^8,\ 0 \leq c_i \leq 10^{16},\ b_i^2 \leq 4a_ic_i)$, denoting a worker.

Output

For each test case, output a single line containing n integers, the k-th $(1 \leq k \leq n)$ of which denoting the minimum possible total cost for k pairs of workers and pieces of equipment.

Example

standard input	standard output
1 3 5 2 3 10 2 -3 10 1 -1 4	4 15 37

Problem F. The Oculus

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

Let's define the Fibonacci sequence $F_1, F_2, \dots$ as $F_1 = 1, F_2 = 2, F_i = F_{i-1} + F_{i-2}$ $(i \geq 3)$.

It's well known that every positive integer x has its unique Fibonacci representation $(b_1, b_2, \dots, b_n)$ such that:

- $b_1 \times F_1 + b_2 \times F_2 + \cdots + b_n \times F_n = x$.
- $b_n = 1$, and for each i $(1 \leq i < n)$, $b_i \in \{0, 1\}$ always holds.
- For each i $(1 \leq i < n)$, $b_i \times b_{i+1} = 0$ always holds.

For example, $4 = (1, 0, 1)$, $5 = (0, 0, 0, 1)$, and $20 = (0, 1, 0, 1, 0, 1)$ because $20 = F_2 + F_4 + F_6 = 2 + 5 + 13$.

There are two positive integers A and B written in Fibonacci representation, Skywalkert calculated the product of A and B and written the result C in Fibonacci representation. Assume the Fibonacci representation of C is $(b_1, b_2, \dots, b_n)$, Little Q then selected a bit k $(1 \leq k < n)$ such that $b_k = 1$ and modified b_k to 0.

It is so slow for Skywalkert to calculate the correct result again using Fast Fourier Transform and tedious reduction. Please help Skywalkert to find which bit k was modified.

Input

The first line of the input contains a single integer T $(1 \leq T \leq 10\,000)$, the number of test cases.

For each case, the first line of the input contains the Fibonacci representation of A, the second line contains the Fibonacci representation of B, and the third line contains the Fibonacci representation of modified C.

Each line starts with an integer n, denoting the length of the Fibonacci representation, followed by n integers $b_1, b_2, \dots, b_n$, denoting the value of each bit.

It is guaranteed that:

- $1 \leq |A|, |B| \leq 1\,000\,000$.
- $2 \leq |C| \leq |A| + |B| + 1$.
- $\sum |A|, \sum |B| \leq 5\,000\,000$.

Output

For each test case, output a single line containing an integer, the value of k.

Example

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

Problem G. In Search of Gold

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

Sunset got a map about an abandoned gold mine in the border town. The map shows that the gold mine consists of n rooms connected by $n-1$ bidirectional tunnels, forming a tree structure. The map is so strange that on the i-th tunnel, there are two numbers a_i and b_i. The only thing Sunset knows is that there are exactly k tunnels whose lengths are taken from a while the lengths of other $n-k-1$ tunnels are taken from b.

Tomorrow Sunset will explore that gold mine. He is afraid of getting lost in the gold mine, so can you please tell him the diameter of the gold mine if he is lucky enough? In other words, please calculate the minimum possible length of the diameter from the information Sunset has.

Input

The first line of the input contains a single integer T ($1 \leq T \leq 10\,000$), the number of test cases.

For each case, the first line of the input contains two integers n and k ($2 \leq n \leq 20\,000$, $0 \leq k \leq n-1$, $k \leq 20$), denoting the number of rooms and the parameter k.

Each of the following $n-1$ lines contains four integers u_i, v_i, a_i and b_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $1 \leq a_i, b_i \leq 10^9$), denoting an bidirectional tunnel between the u_i-th room and the v_i-th room, the length of which is either a_i or b_i.

It is guaranteed that the sum of all n is at most $200\,000$.

Output

For each test case, output a single line containing an integer, the minimum possible length of the diameter.

Example

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

Problem H. Dynamic Convex Hull

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

Let's first see a related classical algorithm to help you solve this problem: You will be given n functions $f_1(x), f_2(x), \ldots, f_n(x)$, where $f_i(x) = a_i x + b_i$. When you want to find the minimum value of $f_i(x)$ for a fixed parameter x, you just need to find the corresponding function on the convex hull.

Now you will be given n functions $f_1(x), f_2(x), \ldots, f_n(x)$, where $f_i(x) = (x - a_i)^4 + b_i$.

You need to perform the following operations for m times:

- "`1 a b`" ($1 \leq a \leq 50\,000, 1 \leq b \leq 10^{18}$): Add a new function $f_{n+1}(x) = (x - a)^4 + b$ and then change n into $n + 1$.
- "`2 t`" ($1 \leq t \leq n$): Delete the function $f_t(x)$. It is guaranteed that each function won't be deleted more than once.
- "`3 x`" ($1 \leq x \leq 50\,000$): Query for the minimum value of $f_i(x)$, where $1 \leq i \leq n$ and the function $f_i(x)$ has not been deleted yet.

Input

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

For each case, the first line of the input contains two integers n and m ($1 \leq n, m \leq 100\,000$), denoting the number of functions and the number of operations.

Each of the following n lines contains two integers a_i and b_i ($1 \leq a_i \leq 50\,000, 1 \leq b_i \leq 10^{18}$), denoting the i-th function $f_i(x)$.

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

Output

For each query, print a single line containing an integer, denoting the minimum value of $f_i(x)$. Note that when there is no functions, print "`-1`" instead.

Example

standard input	standard output
1 2 8 3 9 6 100 3 4 2 1 3 4 1 1 1 3 4 2 2 2 3 3 4	10 116 82 -1

Problem I. It's All Squares

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

One day when Little Q woke up, he found himself being inside a 2D pixel world. The world is a grid with $n \times m$ square cells. Little Q can only walk along the side of these cells, which means he can stay at a point (x, y) if and only if $0 \le x \le n$ and $0 \le y \le m$, where x and y are all integers. There is a number written at the center of each cell, number $w_{i,j}$ $(1 \le i \le n, 1 \le j \le m)$ is written at the point $(i - 0.5, j - 0.5)$.

Little Q had no idea about how to escape from the pixel world, so he started wandering. You will be given q queries, each query consists of two integers (x, y) and a string S, denoting the route of Little Q. Initially, Little Q will stand at (x, y), then he will do $|S|$ steps of movements $S_1, S_2, \dots, S_{|S|}$ one by one. Here is what he will do for each type of movement:

- "`L`" : Move from (x, y) to $(x - 1, y)$.
- "`R`" : Move from (x, y) to $(x + 1, y)$.
- "`D`" : Move from (x, y) to $(x, y - 1)$.
- "`U`" : Move from (x, y) to $(x, y + 1)$.

It is guaranteed that Little Q will never walk outside of the pixel world, and the route will form a simple polygon. For each query, please tell Little Q how many distinct numbers there are inside the polygon formed by the route.

Fortunately, after solving this problem, Little Q woke up on his bed. The pixel world had just been a dream!

Input

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

For each case, the first line of the input contains three integers n, m and q $(1 \le n, m \le 400, 1 \le q \le 200\,000)$, denoting the size of the pixel world and the number of queries.

Each of the following n lines contains m integers, the i-th line contains m integers $w_{i,1}, w_{i,2}, \dots, w_{i,m}$ $(1 \le w_{i,j} \le n \times m)$, denoting the number written in each cell.

Each of the following q lines contains two integers x, y $(0 \le x \le n,\ 0 \le y \le m)$ and a non-empty string S $(S_i \in \{\mathtt{L}, \mathtt{R}, \mathtt{D}, \mathtt{U}\})$, denoting each query.

It is guaranteed that $\sum |S| \le 4\,000\,000$.

Output

For each query, output a single line containing an integer, the number of distinct numbers inside the polygon.

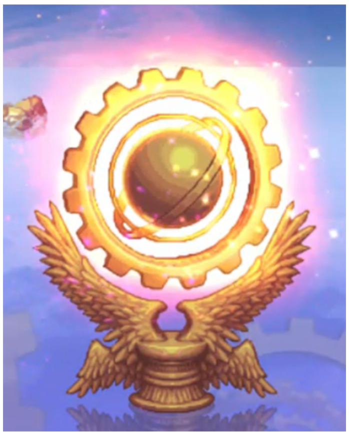
Example

standard input	standard output
1 3 3 2 1 2 3 1 1 2 7 8 9 0 0 RRRUUULLLDDD 0 0 RRUULLDD	6 2

Problem J. Lead of Wisdom

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

In an online game, "Lead of Wisdom" is a place where the lucky player can randomly get powerful items.



There are k types of items, a player can wear at most one item for each type. For the i-th item, it has four attributes a_i, b_i, c_i and d_i. Assume the set of items that the player wearing is S, the damage rate of the player DMG can be calculated by the formula:

$$DMG = \left(100 + \sum_{i \in S} a_i\right)\left(100 + \sum_{i \in S} b_i\right)\left(100 + \sum_{i \in S} c_i\right)\left(100 + \sum_{i \in S} d_i\right)$$

Little Q has got n items from "Lead of Wisdom", please write a program to help him select which items to wear such that the value of DMG is maximized.

Input

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

For each case, the first line of the input contains two integers n and k $(1 \leq n, k \leq 50)$, denoting the number of items and the number of item types.

Each of the following n lines contains five integers t_i, a_i, b_i, c_i and d_i $(1 \leq t_i \leq k,\ 0 \leq a_i, b_i, c_i, d_i \leq 100)$, denoting an item of type t_i whose attributes are a_i, b_i, c_i and d_i.

Output

For each test case, output a single line containing an integer, the maximum value of DMG.

Example

standard input	standard output
1 6 4 1 17 25 10 0 2 0 0 25 14 4 17 0 21 0 1 5 22 0 10 2 0 16 20 0 4 37 0 0 0	297882000

Problem K. King of Hot Pot

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

Little Q is enjoying hot pot together with Tangjz. There are n dishes of meat in the boiling water, labeled by $1, 2, \ldots, n$. The i-th dish of meat will be OK to eat at time a_i, and it will take Little Q b_i units of time to swallow it. You can assume swallowing a dish of meat won't cost any time, and it is impossible for Little Q to swallow more than one dish of meat at the same moment. Note that for the i-th dish of meat, Little Q can choose to ignore it for a while and swallow it later.

Little Q is called "King of Hot Pot", and he wants to show off before Tangjz by swallowing k dishes of meat as soon as possible. Please write a program to help Little Q find k dishes of meat and the order to swallow them such that the time he spends on reaching the target is minimized for every possible value of k ($1 \leq k \leq n$). Note that the time for waiting is also included in the answer.

Input

The first line of the input contains a single integer T ($1 \leq T \leq 10\,000$), the number of test cases.

For each case, the first line of the input contains an integer n ($1 \leq n \leq 300\,000$), denoting the number of dishes of meat.

Each of the following n lines contains two integers a_i and b_i ($1 \leq a_i, b_i \leq 10^9$), denoting each dish of meat.

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 n integers, the k-th ($1 \leq k \leq n$) of which denoting the minimum possible units of time needed for Little Q to swallow k dishes of meat.

Example

standard input	standard output
1 5 1 2 4 6 3 5 4 2 3 2	3 5 7 12 18

Problem L. String Distance

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

For two strings S and T, you can do the following operation for arbitrary number of times: Select a string S or T, insert or delete a character at any position. The distance between two strings S and T is defined as the minimum number of operations to make S and T equal.

You will be given two strings $A[1..n], B[1..m]$ and q queries.

In each query, you will be given two integers l_i and r_i ($1 \le l_i \le r_i \le n$), you need to find the distance between the continous substring $A[l_i..r_i]$ and the whole string B.

Input

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

For each case, the first line of the input contains a string A consists of n ($1 \le n \le 100\,000$) lower-case English letters.

The second line of the input contains a string B consists of m ($1 \le m \le 20$) lower-case English letters.

The third line of the input contains a single integer q ($1 \le q \le 100\,000$), denoting the number of queries.

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

Output

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

Example

standard input	standard output
1 qaqaqwqaqaq qaqwqaq 3 1 7 2 8 3 9	4 2 0