Problem A. Total Eclipse

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

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 operation any number of times:

- Remove all the cities with zero brightness from consideration.
- Select an integer k ($1 \leq k \leq n$).
- Select k distinct unremoved 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.

Sunset will always choose the maximum possible value of k for each operation. 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 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 ($1 \leq n \leq 100\,000$, $1 \leq m \leq 200\,000$), denoting the number of cities and the number of roads.

The second line 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 a 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. Visual Cube

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Little Q likes solving math problems very much. Unluckily, however, he does not have a good spatial ability. Every time 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 anymore. As a result, he turns to you for help.

Given a rectangular parallelepiped with length a, width b, and height c, please write a program to display it.

Input

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

The only line contains three integers a, b, c ($1 \le a, b, c \le 20$), denoting the dimensions of the rectangular parallelepiped.

Output

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

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td>2
1 1 1
6 2 4</td><td>..+-+
././|
+-+.+
|.|/.
+-+..
....+-+-+-+-+-+-+
.../././././././|
..+-+-+-+-+-+-+.+
./././././././|/|
+-+-+-+-+-+-+.+.+
|.|.|.|.|.|.|/|/|
+-+-+-+-+-+-+.+.+
|.|.|.|.|.|.|/|/|
+-+-+-+-+-+-+.+.+
|.|.|.|.|.|.|/|/.
+-+-+-+-+-+-+.+..
|.|.|.|.|.|.|/...
+-+-+-+-+-+-+....</td></tr>
</table>

Problem C. Count on a Tree II Striking Back

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

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 m operations. Each operation is one of the following:

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

Input

The first line contains a single integer T ($1 \le T \le 4$), the number of test cases. For each test case:

The first line contains two integers n and m ($1 \le n \le 500\,000$, $1 \le m \le 10\,000$), denoting the number of nodes and the number of operations.

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

Each of the following $n-1$ lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$) denoting a 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: their actual values 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 $f(a, b) \ge 2f(c, d)$ or $f(c, d) \ge 2f(a, b)$ always holds for each query.

Output

For each query, print a single line. If $f(a, b) > f(c, d)$ is true, print "`Yes`". Otherwise, 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:	6 seconds
Memory limit:	512 mebibytes

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 (i,j) which is at row i and column j, there are $\left(n^2\right)^{a_{i,j}}$ diamonds.

You start at $(1,1)$ and move to (n,n). At any cell (i,j), you can move to $(i+1,j)$ or $(i,j+1)$, provided that you don't move out of the grid. Clearly, you will make exactly $2n-2$ steps. When you are at a cell, you can take all the diamonds at 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 r_1, r_2, c_1, c_2, and you need to report the maximum number of diamonds that you can take without passing the cells (i,j) such that $r_1 \le i \le r_2$ and $c_1 \le j \le c_2$.

Input

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

The first line 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 r_1, r_2, c_1, c_2 $(1 \le r_1 \le r_2 \le n,\ 1 \le c_1 \le c_2 \le n)$ describing a 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: 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 mebibytes

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 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 ($1 \leq n \leq 50$, $n \leq m \leq 10^8$) 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, c_i ($1 \leq a_i \leq 10$, $-10^8 \leq b_i \leq 10^8$, $0 \leq c_i \leq 10^{16}$, $b_i^2 \leq 4a_ic_i$) describing a worker.

Output

For each test case, output a single line containing n integers, the k-th ($1 \leq k \leq n$) of which is 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 Missing Pet

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

You are given an $n \times n$ chessboard, the rows and columns of which are numbered from 1 to n respectively. Little Q punched several holes in specified locations: the i-th hole is located at (r_i, c_i).

Little Q also has a pet dog. Now, the dog is getting lost on the chessboard at the cell (r_0, c_0). It will move to a random adjacent cell every second. Each of the adjacent cells is selected with equal probability. Here, two cells are adjacent if they share a common edge. If the dog arrives at a cell with a punched hole, it will fall into the hole.

Now, Little Q is wondering: for each hole, what is the expected number of seconds that the pet walks on the chessboard, given that it finally falls into this hole? Please help him.

Input

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

The first line contains two integers n and k ($2 \le n \le 200$, $1 \le k \le 200$) indicating the size of the given chessboard and the number of holes.

Then k lines follow, the i-th of which contains two integers r_i and c_i ($1 \le r_i, c_i \le n$) indicating the location of the i-th hole.

The last line of each test case contains two integers r_0 and c_0 ($1 \le r_0, c_0 \le n$) denoting the starting location of the pet.

It is guaranteed that all given holes are distinct, and the pet is not located at a hole initially. It is also guaranteed that $\max(n, k) > 5$ holds in at most one test case.

Output

For each test case, output a single line with k integers: for each hole, in the order they are given in the input, print the expected number of seconds the pet walks on the chessboard, given that it finally falls into this hole.

More precisely, if a hole is reachable and the reduced fraction of the expected number of seconds is $\frac{p}{q}$, you should output the minimum non-negative integer r such that $q \cdot r \equiv p \pmod{10^9 + 7}$. You may safely assume that such r always exists in all test cases. If a hole is unreachable, output "`-1`" instead.

Example

standard input	standard output
2 3 3 1 1 1 2 2 1 2 2 5 3 5 3 4 1 3 2 4 5	-1 4 4 669185882 381533358 341349117

Problem G. In Search of Gold

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

Sunset got a map of 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 contains a single integer T ($1 \le T \le 10\,000$), the number of test cases. For each test case:

The first line contains two integers n and k ($2 \le n \le 20\,000$, $0 \le k \le n-1$, $k \le 20$) 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, b_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$, $1 \le a_i, b_i \le 10^9$) denoting a 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 mebibytes

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 $f_i(x)$ over all i 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 m operations. Each operation has one of the following forms:

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

Input

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

The first line contains two integers n and m ($1 \le n, m \le 100\,000$) denoting the number of functions and the number of operations.

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

Each of the next m lines describes an operation in the format shown above.

Output

For each query, print a single line containing an integer denoting the minimum value of $f_i(x)$. When there are 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 mebibytes

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 sides 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 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 $S_1, S_2, \ldots, S_{|S|}$ 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 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, q ($1 \le n, m \le 400$, $1 \le q \le 200\,000$) denoting the dimensions 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 $w_{i,1}, w_{i,2}, \ldots, w_{i,m}$ ($1 \le w_{i,j} \le n \times m$) denoting the number written in each cell. (Note that you will have to rotate this representation if you want "U" to actually mean "up", etc.)

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

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

Output

For each query, output a line with a single integer: how many distinct numbers are 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. Walking Plan

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

There are n intersections in Bytetown, connected with m one-way streets. These intersections are labeled by $1, 2, \ldots, n$. 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, move along a street at least k_i times, and finally arrive to the t_i-th intersection. Note that k_i is the required number of *moves*, not *streets*: it is allowed to use any street more than once.

Little Q's smartphone will record his walking route. Little Q cares more about statistics than about staying healthy. So he wants to minimize the total walking length on each day. Please write a program to help him find the best route.

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 m ($2 \le n \le 50$, $1 \le m \le 10\,000$) denoting the number of intersections and one-way streets.

Each of the next m lines contains three integers u_i, v_i, w_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$, $1 \le w_i \le 10\,000$) denoting a one-way street from intersection u_i to intersection v_i with length w_i.

In the next line, there is an integer q ($1 \le q \le 100\,000$) denoting the number of days.

Each of the next q lines contains three integers s_i, t_i, k_i ($1 \le s_i, t_i \le n$, $1 \le k_i \le 10\,000$) describing the walking plan.

Output

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

Example

standard input	standard output
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

Problem K. King of Hot Pot

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

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 ready at moment a_i, and it will take Little Q b_i units of time to fully eat it. Little Q can start eating dish i at any moment $t \geq a_i$, and then he has to eat it until moment $t + b_i$. Little Q can't be eating more than one dish of meat at the same time.

Little Q is called "King of Hot Pot", and he wants to show off before Tangjz by fully eating k dishes of meat as soon as possible. The timer starts at moment 0. Please write a program to help Little Q, for each k independently ($1 \leq k \leq n$), find k dishes of meat and the order to eat them such that the total time before he fully eats k dishes is minimized. Note that any waiting time is also included in the answer.

Input

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

The first line 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 ($1 \leq a_i, b_i \leq 10^9$) describing a 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 is the minimum total time before Little Q can fully eat k dishes of meat.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td>1
5
1 2
4 6
3 5
4 2
3 2</td><td>3 5 7 12 18</td></tr>
</table>

Problem L. String Distance

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

For two strings S and T, you can do the following operation an 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 $A[1..n]$ and $B[1..m]$, and also 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 continuous substring $A[l_i..r_i]$ and the whole string B.

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 string A which consists of n ($1 \le n \le 100\,000$) lower-case English letters.

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

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

Each of the following q lines contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$) describing 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