Problem A. Is it well known in Poland?

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

This problem might be well-known in some countries, but how do other countries learn about such problems if nobody poses them?

Little Cyan Fish (Xiao Qingyu) and Huge Nucleus Kernel (Da Heren) are two inseparable friends.

In the year 2020, during his training for the National Olympiad, Little Cyan Fish endeavored to solve a fascinating problem from Potyczki Algorytmiczne 2010.

> Two termites are eating an old wooden fence. This fence consists of planks of possibly different heights. Termites have already eaten some of them, and they thought that they should make their meal more interesting. They have decided to play a game and eat the planks in turns, one by one. During one turn, a termite may choose to eat only a plank which is next to a plank that has already been consumed.
>
> Assuming that each termite chooses the planks in such a way, that during the whole game the sum of heights of all planks eaten by her is as big as possible, compute the amount of wood that each of them will have eaten.
>
> *Task author: Tomasz Idziaszek.*

The problem intrigued Little Cyan Fish, leaving a profound impression on him.

Years later, when Huge Nucleus Kernel needed to prepare a task for a competition, he shared it with Little Cyan Fish. Little Cyan Fish was astounded as it reminded him of that captivating problem. To inspire more people to attempt and solve this intriguing problem, Little Cyan Fish and Huge Nucleus Kernel decided to include it in a Universal Cup Contest.

> Little A and Little B are engaged in a game. They are presented with a rooted tree forest, where each vertex u carries a positive integer value A_u.
>
> Little A and Little B alternate turns, with Little A starting the game. The current player must choose exactly one tree root to eliminate, thereby gaining its node value. The subtree of the eliminated root forms a new rooted tree, and the children of the eliminated root become the new tree roots.
>
> The game concludes when all vertices have been removed. The score of a player is the sum of the vertices values they have eliminated.
>
> Both players aim to optimize their scores, employing the best strategies. Determine the final score of Little A.
>
> The initial scenario provides a single tree with n vertices. The vertices are numbered from 1 to n, with vertices 1 being the root.

Input

The first line contains an integer n ($2 \le n \le 10^5$) indicating the size of the tree.

The second line contains n integers $a_1, a_2, \cdots, a_n$ ($1 \le a_i \le 10^9$) where a_i indicates the value of the vertex i.

For the following $(n-1)$ lines, the i-th line contains two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$) indicating an edge connecting vertices u_i and v_i.

Output

Output a single line contains a single integer, indicating the answer.

Example

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

Problem B. Path Planning

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

There is a grid with n rows and m columns. Each cell of the grid has an integer in it, where $a_{i,j}$ indicates the integer in the cell located at the i-th row and the j-th column. Each integer from 0 to $(n \times m - 1)$ (both inclusive) appears exactly once in the grid.

Let (i, j) be the cell located at the i-th row and the j-th column. You now start from $(1, 1)$ and need to reach (n, m). When you are in cell (i, j), you can either move to its right cell $(i, j + 1)$ if $j < m$ or move to its bottom cell $(i + 1, j)$ if $i < n$.

Let $\mathbb{S}$ be the set consisting of integers in each cell on your path, including $a_{1,1}$ and $a_{n,m}$. Let $\text{mex}(\mathbb{S})$ be the smallest non-negative integer which does not belong to $\mathbb{S}$. Find a path to maximize $\text{mex}(\mathbb{S})$ and calculate this maximum possible value.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \le n, m \le 10^6$, $1 \le n \times m \le 10^6$) indicating the number of rows and columns of the grid.

For the following n lines, the i-th line contains m integers $a_{i,1}, a_{i,2}, \cdots, a_{i,m}$ ($0 \le a_{i,j} < n \times m$) where $a_{i,j}$ indicates the integer in cell (i, j). Each integer from 0 to $(n \times m - 1)$ (both inclusive) appears exactly once in the grid.

It's guaranteed that the sum of $n \times m$ of all test cases will not exceed 10^6.

Output

For each test case output one line containing one integer indicating the maximum possible value of $\text{mex}(\mathbb{S})$.

Example

standard input	standard output
2 2 3 1 2 4 3 0 5 1 5 1 3 0 4 2	3 5

Note

For the first sample test case there are 3 possible paths.

- The first path is $(1, 1) \to (1, 2) \to (1, 3) \to (2, 3)$. $\mathbb{S} = \{1, 2, 4, 5\}$ so $\text{mex}(\mathbb{S}) = 0$.
- The second path is $(1, 1) \to (1, 2) \to (2, 2) \to (2, 3)$. $\mathbb{S} = \{1, 2, 0, 5\}$ so $\text{mex}(\mathbb{S}) = 3$.
- The third path is $(1, 1) \to (2, 1) \to (2, 2) \to (2, 3)$. $\mathbb{S} = \{1, 3, 0, 5\}$ so $\text{mex}(\mathbb{S}) = 2$.

So the answer is 3.

For the second sample test case there is only 1 possible path, which is $(1, 1) \to (1, 2) \to (1, 3) \to (1, 4) \to (1, 5)$. $\mathbb{S} = \{1, 3, 0, 4, 2\}$ so $\text{mex}(\mathbb{S}) = 5$.

Problem C. New but Nostalgic Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Given n strings $w_1, w_2, \cdots, w_n$, please select k strings among them, so that the lexicographic order of string v is minimized, and output the optimal string v. String v satisfies the following constraint: v is the longest common prefix of two selected strings with different indices. Also, v is the lexicographically largest string among all strings satisfying the constraint.

More formally, let $\mathbb{S}$ be a set of size k, where all the elements in the set are integers between 1 and n (both inclusive) and there are no duplicated elements. Let $\text{lcp}(w_i, w_j)$ be the longest common prefix of string w_i and w_j, please find a set $\mathbb{S}$ to minimize the lexicographic order of the following string v and output the optimal string v.

$$v = \max_{i \in \mathbb{S}, j \in \mathbb{S}, i \neq j} \text{lcp}(w_i, w_j)$$

In the above expression, max is calculated by comparing the lexicographic order of strings.

Recall that:

- String p is a prefix of string s, if we can append some number of characters (including zero characters) at the end of p so that it changes to s. Specifically, empty string is a prefix of any string.
- The longest common prefix of string s and string t is the longest string p such that p is a prefix of both s and t. For example, the longest common prefix of "abcde" and "abcef" is "abc", while the longest common prefix of "abcde" and "bcdef" is an empty string.
- String s is lexicographically smaller than string t ($s \neq t$), if
  - s is a prefix of t, or
  - $s_{|p|+1} < t_{|p|+1}$, where p is the longest common prefix of s and t, $|p|$ is the length of p, s_i is the i-th character of string s, and t_i is the i-th character of string t.

  Specifically, empty string is the string with the smallest lexicographic order.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and k ($2 \le n \le 10^6$, $2 \le k \le n$) indicating the total number of strings and the number of strings to be selected.

For the following n lines, the i-th line contains a string w_i ($1 \le |w_i| \le 10^6$) consisting of lower-cased English letters.

It's guaranteed that the total length of all strings of all test cases will not exceed 10^6.

Output

For each test case output one line containing one string indicating the answer. Specifically, if the answer is an empty string, print `EMPTY`.

Example

standard input	standard output
2 5 3 gdcpc gdcpcpcp suasua suas sususua 3 3 a b c	gdcpc EMPTY

Problem D. Computational Geometry

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	1024 megabytes

Given a convex polygon P with n vertices, you need to choose two vertices of P, so that the line connecting the two vertices will split P into two smaller polygons Q and R, both with positive area.

Let $d(Q)$ be the diameter of polygon Q and $d(R)$ be the diameter of polygon R, calculate the minimum value of $(d(Q))^2 + (d(R))^2$.

Recall that the diameter of a polygon is the maximum distance between two points inside or on the border of the polygon.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains an integer n ($4 \le n \le 5 \times 10^3$) indicating the number of vertices of the convex polygon P.

For the following n lines, the i-th line contains two integers x_i and y_i ($0 \le x_i, y_i \le 10^9$) indicating the i-th vertex of the convex polygon P. Vertices are given in counter-clockwise order. It's guaranteed that the area of the convex polygon is positive, and there are no two vertices with the same coordinate. It's possible that three vertices lie on the same line.

It's guaranteed that the sum of n of all test cases will not exceed 5×10^3.

Output

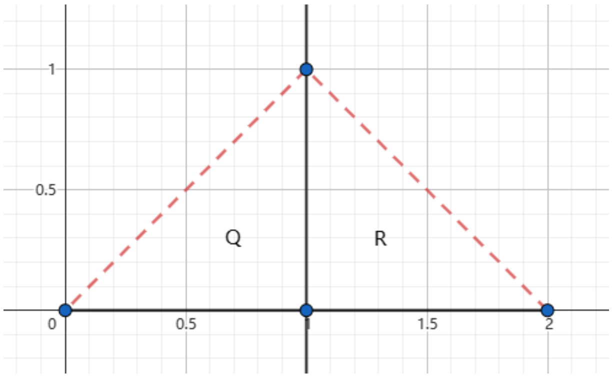
For each test case output one line containing one integer indicating the answer.

Example

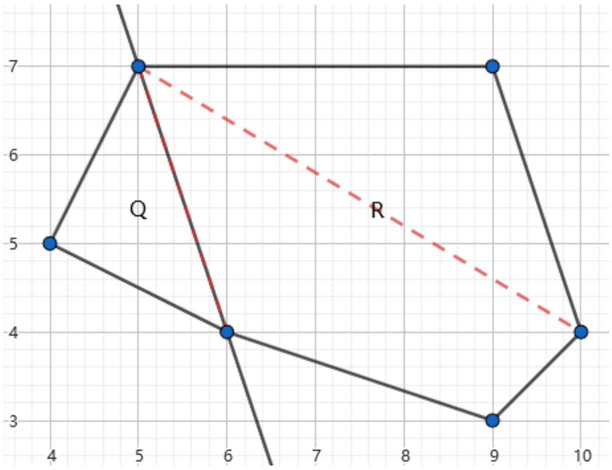
standard input	standard output
2 4 1 0 2 0 1 1 0 0 6 10 4 9 7 5 7 4 5 6 4 9 3	4 44

Note

The first sample test case is shown as follows. The diameter of smaller polygons are marked by red dashed segments. In fact, $(1,0)$ and $(1,1)$ are the only pair of vertices we can choose in this test case. You can't choose $(0,0)$ and $(2,0)$, or $(0,0)$ and $(1,1)$, because they can't split P into two smaller polygons both with positive area.



The second sample test case is shown as follows. The diameter of smaller polygons are marked by red dashed segments.



Problem E. Not Another Linear Algebra Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

What age is it that you are still solving traditional linear algebra problem?

You are given a prime number q.

Suppose A and B are two $n \times n$ square matrices such that $AB \equiv A \pmod q$, and each element of A and B is an integer from 0 to $q - 1$.

- Here, $S \equiv T \pmod q$ implies that for each $1 \le i, j \le n$, we have $S_{i,j} \equiv T_{i,j} \pmod q$.

Given a fixed matrix B ($\det B \neq 0$), it's too easy for you to just find an arbitrary suitable matrix A.

Let $f(B)$ represent the number of matrices that satisfy the equation above. Your task is to calculate:

$$\sum_{B \in M_n(\mathbb{F}_q)} [\det B \neq 0] 3^{f(B)}$$

The answer can be quite large, you only need to output it modulo another given prime number, *mod*.

Input

The first line of the input contains three integers n, q and mod. ($1 \le n \le 10^7$, $2 \le q < mod$, $10^8 \le mod \le 10^9 + 7$).

It is guaranteed that q and mod are two prime numbers.

Output

Output a single line contains a single integer, indicating the answer modulo the given number *mod*.

Examples

standard input	standard output
`2 2 1000000007`	`43046970`
`100 127 998244353`	`881381862`

Note

For $B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, matrices A that satisfy the condition are: $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$, $\begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$, $\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$, $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$, totaling 4.

For $B = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$, the matrices A that satisfy the condition are: $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$, totaling 1.

For $B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, all matrices A satisfy the condition, totaling 16.

For $B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, matrices A that satisfy the condition are: $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$, $\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$, $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, $\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$, totaling 4.

For $B = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, matrices A that satisfy the condition are: $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$, $\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$, $\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$, $\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$, totaling 4.

For $B = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$, the matrices A that satisfy the condition are: $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$, totaling 1.

Therefore, the answer is $3^4 + 3^1 + 3^{16} + 3^4 + 3^4 + 3^1 \equiv 43046970 \pmod{10^9 + 7}$.

Problem F. X Equals Y

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

For positive integers X and $b \geq 2$, define $f(X, b)$ as a sequence which describes the base-b representation of X, where the i-th element in the sequence is the i-th least significant digit in the base-b representation of X. For example, $f(6, 2) = \{0, 1, 1\}$, while $f(233, 17) = \{12, 13\}$.

Given four positive integers x, y, A and B, please find two positive integers a and b satisfying:

- $2 \leq a \leq A$
- $2 \leq b \leq B$
- $f(x, a) = f(y, b)$

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \leq T \leq 10^3$) indicating the number of test cases. For each test case:

The first line contains four integers x, y, A and B ($1 \leq x, y \leq 10^9$, $2 \leq A, B \leq 10^9$).

It's guaranteed that there are at most 50 test cases satisfying $\max(x, y) > 10^6$.

Output

For each test case, if valid positive integers a and b do not exist, output `NO` in one line.

Otherwise, first output `YES` in one line. Then in the next line, output two integers a and b separated by a space. If there are multiple valid answers, you can output any of them.

Example

standard input	standard output
6 1 1 1000 1000 1 2 1000 1000 3 11 1000 1000 157 291 5 6 157 291 3 6 10126 114514 789 12345	YES 2 2 NO YES 2 10 YES 4 5 NO YES 779 9478

Problem G. Classic Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	8 seconds
Memory limit:	1024 megabytes

Given an undirected complete graph with n vertices and m triples $P_1, P_2, \cdots, P_m$ where $P_i = (u_i, v_i, w_i)$, it's guaranteed that $1 \le u_i < v_i \le n$, and for any two triples P_i and P_j with different indices we have $(u_i, v_i) \ne (u_j, v_j)$.

For any two vertices x and y in the graph ($1 \le x < y \le n$), define the weight of the edge connecting them as follows:

- If there exists a triple P_i satisfying $u_i = x$ and $v_i = y$, the weight of edge will be w_i.
- Otherwise, the weight of edge will be $|x - y|$.

Calculate the total weight of edges in the minimum spanning tree of the graph.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^5$) indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \le n \le 10^9$, $0 \le m \le 10^5$) indicating the number of vertices in the graph and the number of triples.

For the following m lines, the i-th line contains three integers u_i, v_i and w_i ($1 \le u_i < v_i \le n$, $0 \le w_i \le 10^9$) indicating the i-th triple. It's guaranteed that for all $1 \le i < j \le m$ we have $(u_i, v_i) \ne (u_j, v_j)$.

It's guaranteed that the sum of m of all test cases will not exceed 5×10^5.

Output

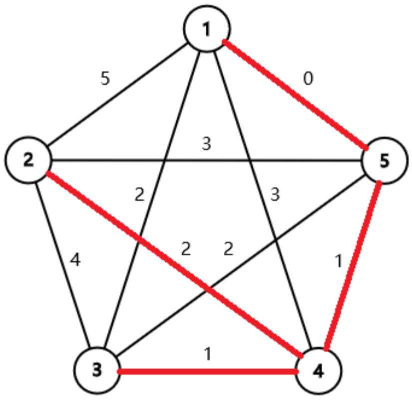
For each test case output one line containing one integer indicating the total weight of edges in the minimum spanning tree of the graph.

Example

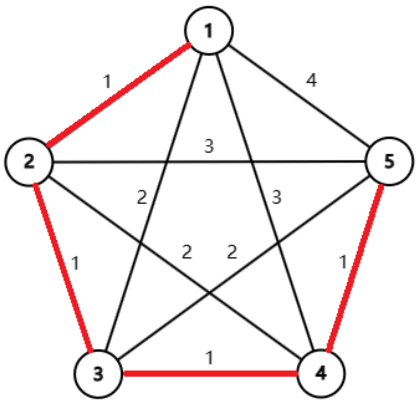
standard input	standard output
3 5 3 1 2 5 2 3 4 1 5 0 5 0 5 4 1 2 1000000000 1 3 1000000000 1 4 1000000000 1 5 1000000000	4 4 1000000003

Note

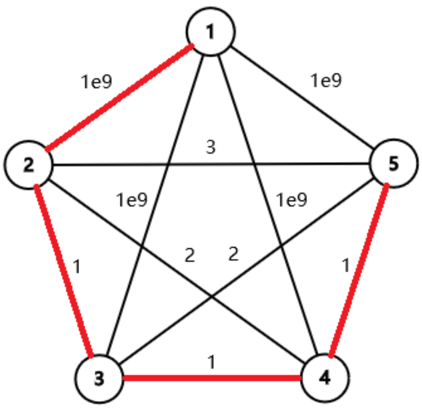
The first sample test case is illustrated as follows. The minimum spanning tree is marked by red segments.



The second sample test case is illustrated as follows. The minimum spanning tree is marked by red segments.



The third sample test case is illustrated as follows. The minimum spanning tree is marked by red segments.



Problem H. Swapping Operation

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 1024 megabytes

Given a non-negative integer sequence $A = a_1, a_2, \ldots, a_n$ of length n, define

$$F(A) = \max_{1 \le k < n}((a_1 \,\&\, a_2 \,\&\, \cdots \,\&\, a_k) + (a_{k+1} \,\&\, a_{k+2} \,\&\, \cdots \,\&\, a_n))$$

where & is the bitwise-and operator.

You can perform the swapping operation at most once: choose two indices i and j such that $1 \le i < j \le n$ and then swap the values of a_i and a_j.

Calculate the maximum possible value of $F(A)$ after performing at most one swapping operation.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains an integer n $(2 \le n \le 10^5)$ indicating the length of sequence A.

The second line contains n integers $a_1, a_2, \cdots, a_n$ $(0 \le a_i \le 10^9)$ indicating the given sequence A.

It's guaranteed that the sum of n of all test cases will not exceed 10^5.

Output

For each test case output one line containing one integer indicating the maximum possible value of $F(A)$ after performing at most one swapping operation.

Example

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

Note

For the first sample test case, we can swap a_4 and a_6 so the sequence becomes $\{6, 5, 4, 6, 5, 3\}$. We can then choose $k = 5$ so that $F(A) = (6 \,\&\, 5 \,\&\, 4 \,\&\, 6 \,\&\, 5) + (3) = 7$.

For the second sample test case, we can swap a_2 and a_4 so the sequence becomes $\{1, 1, 1, 2, 2, 2\}$. We can then choose $k = 3$ so that $F(A) = (1 \,\&\, 1 \,\&\, 1) + (2 \,\&\, 2 \,\&\, 2) = 3$.

For the third sample test case we do not perform the swapping operation. We can then choose $k = 2$ so that $F(A) = (1 \,\&\, 1) + (2 \,\&\, 2 \,\&\, 2) = 3$.

Problem I. Digit Mode

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Let $m(x)$ be the *mode* of the digits in decimal representation of positive integer x. The mode is the largest value that occurs most frequently in the sequence. For example, $m(15532) = 5$, $m(25252) = 2$, $m(103000) = 0$, $m(364364) = 6$, $m(114514) = 1$, $m(889464) = 8$.

Given a positive integer n, DreamGrid would like to know the value of $(\sum_{x=1}^{n} m(x)) \bmod (10^9 + 7)$.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains a positive integer n ($1 \le n < 10^{50}$) without leading zeros.

It's guaranteed that the sum of $|n|$ of all test cases will not exceed 50, where $|n|$ indicates the number of digits of n in decimal representation.

Output

For each test case output one line containing one integer, indicating the value of $(\sum_{x=1}^{n} m(x)) \bmod (10^9+7)$.

Example

standard input	standard output
5 9 99 999 99999 999999	45 615 6570 597600 5689830

Problem J. Escape Plan

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

BaoBao, one of the most famous monster hunters, wakes up in the middle of Heltion City dominated by monsters. Having troubles remembering what has happened, BaoBao decides to escape from this horrible city as soon as possible. Despite arming no weapon, he luckily puts his hand on a map in his right pocket, which contains valuable information that can possibly help him find a way out.

According to the map, Heltion City is composed of n spots connected by m undirected paths. Starting from spot 1, BaoBao must head towards any of the k exits of the city to escape, where the i-th of them is located at spot e_i.

However, it's not an easy task for BaoBao to escape since monsters are everywhere in the city! For all $1 \le i \le n$, d_i monsters are wandering near the i-th spot, so right after BaoBao arrives at that spot, at most d_i paths connecting the spot will be blocked by monsters and are unable for BaoBao to pass. When BaoBao leaves the i-th spot, the monsters will go back to their nests and the blocked paths are clear. Of course, if BaoBao comes back to the spot, at most d_i paths will be again blocked by the monsters. The paths blocked each time may differ.

As BaoBao doesn't know which paths will be blocked, please help him calculate the shortest time he can escape from the city in the worst case.

Input

There are multiple test cases. The first line of the input contains an integer T (about 100), indicating the number of test cases. For each test case:

The first line contains three integers n, m and k ($1 \le n \le 10^5$, $1 \le m \le 10^6$, $1 \le k \le n$), indicating the number of spots, the number of undirected paths and the number of exits of the city.

The second line contains k distinct integers $e_1, e_2, \dots, e_k$ ($1 \le e_i \le n$), indicating k exits of Heltion City.

The third line contains n integers $d_1, d_2, \dots, d_n$ ($0 \le d_i \le m$), where d_i indicates the number of monsters at the i-th spot.

For the following m lines, the i-th line contains three integers x_i, y_i and w_i ($1 \le x_i, y_i \le n$, $x_i \ne y_i$, $1 \le w_i \le 10^4$), indicating an undirected edge of length w_i connecting spot x_i and y_i.

It's guaranteed that the total sum of n will not exceed 10^6 and the total sum of m will not exceed 3×10^6.

Output

For each case output one line containing one integer. If BaoBao can get to some exit in the worst case, output the shortest possible time cost; Otherwise if BaoBao cannot get to any exit in the worst case, output "-1" (without quotes) instead.

Example

standard input	standard output
2 3 4 1 3 1 1 1 1 2 1 1 2 2 2 3 1 2 3 2 3 2 2 2 3 2 0 0 1 2 1 1 3 1	4 -1

Problem K. Final Defense Line

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	64 megabytes

There is a circle in the plane. Both the coordinates of the center and the radius are unknown.

Chiaki found three distinct points A, B and C in the plane. And she also knows the shortest distance from each point to the circumference.

Chiaki would like to find the smallest circle according to above information.

Note that in general, a circle with infinite radius is a line. But in this problem, line is not considered as a circle.

Input

There are multiple test cases. The first line of input contains an integer T ($1 \le T \le 2 \times 10^5$), indicating the number of test cases. For each test case:

The first line contains three integers x_a, y_a and d_a ($-100 \le x_a \le 100, y_a = 0, 1 \le d_a \le 100$) denoting the coordinates of A and the shortest distance to the circumference.

The second line contains three integers x_b, y_b and d_b ($-100 \le x_b \le 100, y_b = 0, 1 \le d_b \le 100$) denoting the coordinates of B and the shortest distance to the circumference.

The third line contains three integers x_c, y_c and d_c ($-100 \le x_c, y_c, d_c \le 100, d_c \ne 0$) denoting the coordinates of C and the shortest distance to the circumference.

If the distance is equal to 0, the point is on the circumference. If distance is greater than 0, the point is inside the circle. If distance is less than 0, the point is outside the circle and the shortest distance is the absolute value.

It is guaranteed that the minimum possible radius of the circle is at most 10^4.

Output

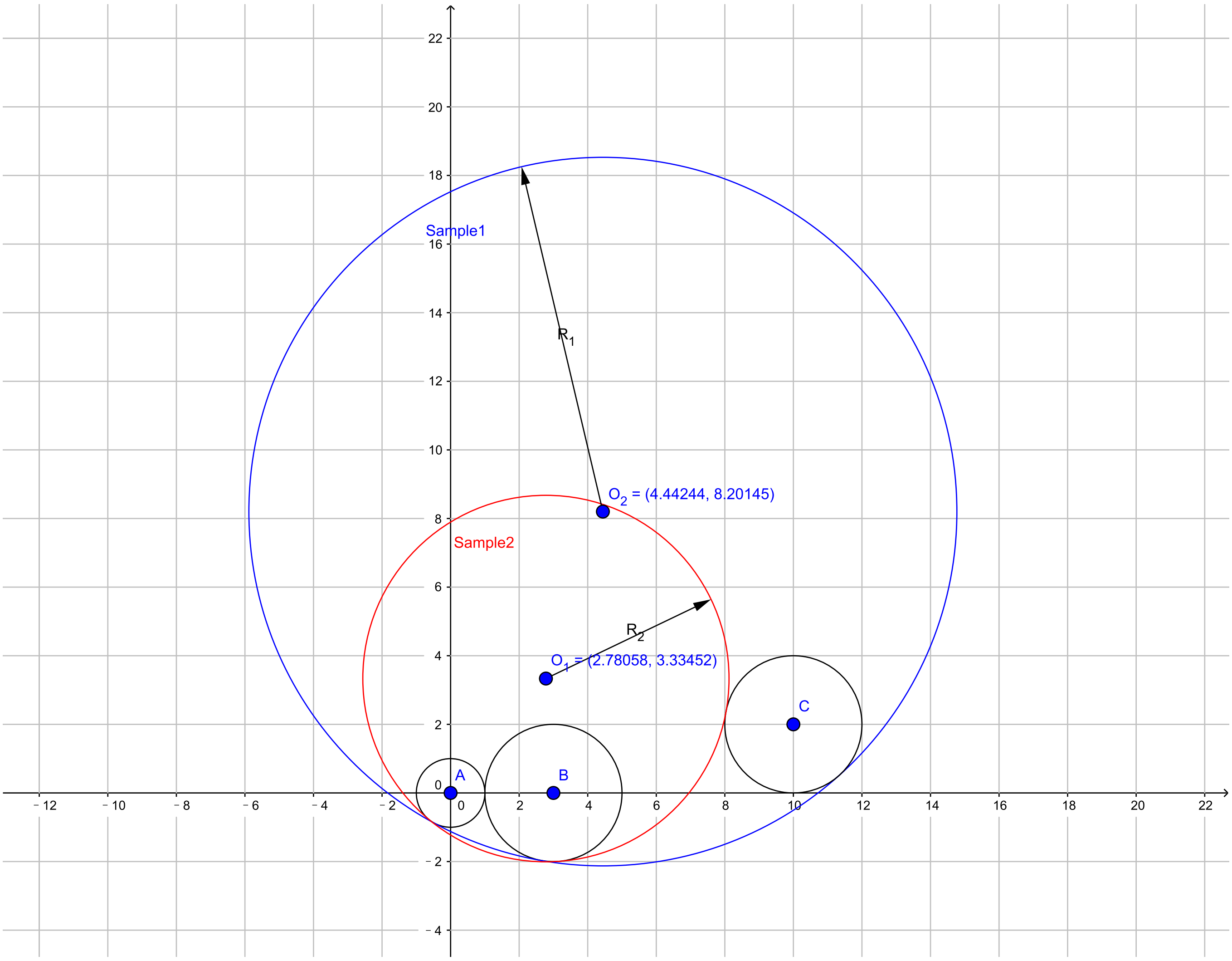
For each test case, if there are infinite possible circles, output -1 in a single line. If there is no such circle, output 0 in a single line. Otherwise, output an integer m and a real number r in a single line separated by one space denoting the number of possible circles and the radius of the smallest circle. You answer will be accepted if the relative error of your answer is no more than 10^{-6}.

Example

standard input	standard output
2 0 0 1 3 0 2 10 2 2 0 0 1 3 0 2 10 2 -2	2 10.327329213474 2 5.341730785446

Note

The image below shows the sample.


Sample1
Sample2
R_1
R_2
O_2 = (4.44244, 8.20145)
O_1 = (2.78058, 3.33452)
A
B
C

Problem L. New Houses

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

With the construction and development of Guangdong, more and more people choose to come to Guangdong to start a new life. In a recently built community, there will be n people moving into m houses which are arranged in a row. The houses are numbered from 1 to m (both inclusive). House u and v are neighboring houses, if and only if $|u - v| = 1$. We need to assign each person to a house so that no two people will move into the same house. If two people move into a pair of neighboring houses, they will become neighbors of each other.

Some people like to have neighbors while some don't. For the i-th person, if he has at least one neighbor, his happiness will be a_i; Otherwise if he does not have any neighbor, his happiness will be b_i.

As the planner of this community, you need to maximize the total happiness.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \le n \le 5 \times 10^5$, $1 \le m \le 10^9$, $n \le m$) indicating the number of people and the number of houses.

For the following n lines, the i-th line contains two integers a_i and b_i ($1 \le a_i, b_i \le 10^9$) indicating the happiness of the i-th person with and without neighbors.

It's guaranteed that the sum of n of all test cases will not exceed 10^6.

Output

For each test case output one line containing one integer indicating the maximum total happiness.

Example

standard input	standard output
3 4 5 1 100 100 1 100 1 100 1 2 2 1 10 1 10 2 3 100 50 1 1000	400 2 1050

Note

For the first sample test case, the optimal strategy is to let person 1 move into house 1 and let person 2 to 4 move into house 3 to 5. Thus, person 1 have no neighbors while person 2 to 4 have neighbors. The answer is $100 + 100 + 100 + 100 = 400$. Of course, we can also let person 2 to 4 move into house 1 to 3 and let person 1 move into house 5. This will also give us 400 total happiness.

For the second sample test case, as there are only 2 houses, person 1 and 2 have to be neighbors. The answer is $1 + 1 = 2$.

For the third sample test case, the optimal strategy is to let person 1 move into house 1 and let person 2 move into house 3. Thus, both of them have no neighbors. The answer is $50 + 1000 = 1050$.

Problem M. Canvas

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

There is a sequence of length n. At the beginning, all elements in the sequence equal to 0. There are also m operations, where the i-th operation will change the value of the l_i-th element in the sequence to x_i, and also change the value of the r_i-th element in the sequence to y_i. Each operation must be performed exactly once.

Find the optimal order to perform the operations, so that after all operations, the sum of all elements in the sequence is maximized.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and m ($2 \le n, m \le 5 \times 10^5$) indicating the length of the sequence and the number of operations.

For the following m lines, the i-th line contains four integers l_i, x_i, r_i and y_i ($1 \le l_i < r_i \le n, 1 \le x_i, y_i \le 2$) indicating the i-th operation.

It's guaranteed that neither the sum of n nor the sum of m of all test cases will exceed 5×10^5.

Output

For each test case, first output one line containing one integer, indicating the maximum sum of all elements in the sequence after all operations. Then output another line containing m integers $a_1, a_2, \cdots, a_m$ separated by a space, indicating the optimal order to perform the operations, where a_i is the index of the i-th operation to be performed. Each integer from 1 to m (both inclusive) must appear exactly once. If there are multiple valid answers, you can output any of them.

Example

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

Note

For the first sample test case, after performing operations $4, 1, 3, 2$ in order, the sequence becomes $\{2, 2, 2, 1\}$. The sum of all elements is 7.

For the second sample test case, after performing operations $2, 1$ in order, the sequence becomes $\{2, 0, 2, 1\}$. The sum of all elements is 5.