Problem A. Elementary

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

The famous black pearl of the Borgias has been stolen and Sherlock is on the case. Now, the first step that needs to be taken is determining some parameters of the valuable jewel, in particular its volume. Luckily Scotland Yard is in possession of a 3D model of the missing gem, which, surprisingly enough for a pearl, is a tetrahedron. The task is obviously too boring for Sherlock, so he asked you to perform the calculation for him.

Input

The first line of input contains the number of test cases t ($t \leq 100$). The input for a single test case consists of four lines.The i-th of them contains three integers x_i, y_i, z_i ($-1000 \leq x_i, y_i, z_i \leq 1000$), which denote the coordinates of the i-th vertex of the tetrahedron.

Output

For each test case in the only line of output print a single floating point number representing the volume of the pearl. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Examples

standard input	standard output
2 0 0 0 10 0 0 0 10 0 0 0 10 1 2 3 6 5 4 12 8 9 11 12 13	166.666667 16.666667

Problem B. Liquids

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

In London there is a standard water supply system consisting of bidirectional pipes and connection points between them. A little less standard is its new functionality, recently added by Mycroft: both water and oil can be sent through the supply sytem at the same time. These liquids do not mix with each other, so it is safe to use the same pipes.

The two most important connection points are the oil refinery and the water tower, which we will call the source points of oil and water. There are also specified a collection point for oil and a collection point for water. Some of these points may overlap, but no collection point coincides with the source point of the same liquid. Each pipe has a limited flow capacity that must not be crossed, but the pumps of the supply system are strong enough to send water in one direction and oil in the other direction in the same pipe as long as the sum of the flows does not exceed the pipe's capacity. Mycroft would now like to know what is the maximum total flow capacity of oil and water that can be available at the collection points and asked you for help.

Input

The first line of input contains the number of testcases t ($t \leq 100$). The first line of each testcase contains two integers n and m ($1 \leq n \leq 200$), the number of connection points and the number of pipes. The next m lines contain descriptions of the pipes. The i-th pipe is described by three integers a_i, b_i, c_i ($a_i \neq b_i, 1 \leq c_i \leq 1000$) that mean that this pipe has a capacity c_i and connects points a_i and b_i. No two pipes connect the same pair of points. The next line contains two integers s_w, t_w, the source point for water and the collection point for water. The last line for each input contains two integers s_o, t_o, the source point for oil and the collection point for oil.

Output

For each test case output the maximum total flow capacity of oil and water that can be available at the collection points. Your answer will be accepted if for each test case the relative or absolute error is less than 10^{-6}.

Examples

standard input	standard output
3 5 4 1 3 1 2 3 2 3 4 1 3 5 2 1 4 2 5 5 4 1 3 1 2 3 2 3 4 2 3 5 1 1 4 2 5 8 8 1 5 1 2 6 1 3 7 1 4 8 1 5 6 1 6 7 1 7 8 1 8 5 1 1 3 2 4	3.000000 2.000000 2.000000

Problem C. Game

Input file:	*standard input*
Output file:	*standard output*
Time limit:	15 seconds
Memory limit:	512 mebibytes

Whenever Sherlock enters his mind palace, he likes to play a little game as a warm up. First, he imagines a weighted undirected graph. Then he performs certain number (possibly zero) of operations. In each operation, he chooses a pair of edges sharing a common endpoint. He removes these edges from the graph and adds the sum of its weights to the score. The goal is to find a sequence of moves maximizing the score.

You have just found a sheet of paper with the description of the graph that Sherlock used the last time he played. You wonder if you are able to reproduce the sequence of moves that achieves the best possible score.

Input

The first line of the input contains an integer $T \leq 50$ denoting the number of test cases. Then, description of T test cases follow.
For every test case, the first line of the input contains two positive integers n and m ($n \leq 10^5$, $m \leq 10^6$): the number of nodes and edges of the graph respectively. Each of the following m lines contains three integers x, y, w ($1 \leq x, y \leq n$, $x \neq y$, $0 \leq w \leq 10^9$) denoting the numbers of nodes connected by an edge and its weight. The graph has no multiple edges.

Output

For each test case, the first line of the output should contain an integer denoting the maximal possible score one can achieve. The next line should contain an integer k — the number of moves to perform. Then, k lines should follow, each containing two integers — the ids of edges to remove. We enumerate the edges from 1 to m, ordered as in the input.

Examples

standard input	standard output
4 4 3 1 2 20 2 3 10 3 4 30 3 3 1 2 1 2 3 2 3 1 3 5 4 1 2 1 1 3 10 1 4 100 1 5 1000 6 4 1 2 1000000 1 3 1000000 4 5 1000000 4 6 1000000	40 1 2 3 5 1 2 3 1111 2 1 2 3 4 4000000 2 1 2 3 4

Problem D. AntiGraffiti

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

Inspector Lestrade got a new assignment: fighting graffiti. From the recent reports he knows that vandals have n litres of paint and they always paint a figure that consists of rectangles placed one after another. The rectangles have a width of 1 meter, start at the ground level and have a positive integer height. To paint $1m^2$ exactly one liter of paint is needed, so the whole graffiti can be represented as a sequence of heights of rectangles $a_1, a_2, \ldots, a_k$, where $\sum_{i=1}^{k} a_i \le n$.

The first part of the plan is to cover the graffiti with a screen. Lestrade wants to be prepared for any possibility, which means being able to cover any possible graffiti. The screen consists of parts of $1m$ width and positive integer heights $b_1, b_2, \ldots b_\ell$. The parts are foldable, so a screen $b_1, b_2, \ldots b_\ell$ can cover a graffiti $a_1, a_2, \ldots, a_k$ if there exists a sequence of indices $1 \le i_1 < i_2 < \ldots < i_k \le \ell$, such that $b_{i_j} \ge a_j$ for all $j = 1, 2, \ldots k$. Scotland Yard does not like to spend too much, so they will order a screen with a minimum possible cost $\sum_{i=1}^{\ell} b_i$.

Let $S(n)$ be the minimum possible cost of a screen that can cover any graffiti made with n litres of paint. To keep Sherlock occupied and out of the way, Lestrade asked him to calculate $\sum_{i=A}^{B} S(i)$ modulo 10^9+7, can you him help with that?

Input

The first line of input contains the number of testcases t ($t \le 100$). Each testcase consists of a single line containing two integers A, B ($1 \le A \le B \le 10^{18}$).

Output

For each test case output the value of $\sum_{i=A}^{B} S(i)$ modulo $10^9 + 7$.

Examples

standard input	standard output
3 1 10 13 17 45 321	134 248 342894

Problem E. Incovenient blockade

Input file:	*standard input*
Output file:	*standard output*
Time limit:	15 seconds
Memory limit:	512 mebibytes

Mycroft comes to Baker Street 221B a little bit too often, so now Sherlock wants to make his travel more inconvenient. To do this, he will block either exactly one junction or exactly one road in London, with the goal to increase the travel time from Mycroft's office.

We can think of London as an undirected graph with n vertices (junctions) and positive weights on edges (roads) that represent travel time. Baker Street 221B is located at the junction number 1, and Mycroft's office at the junction number n. The happiness of Sherlock will be equal to the difference between the length of the shortest path between vertices 1 and n after and before the blocking of a junction or a street. Junctions 1 and n can not be blocked, and if after the blockade junction 1 is not reachable from juncion n, Sherlock's happiness will be equal to -1, because Mycroft will then know that something is up. Sherlock has asked you for help to calculate what his happiness will be after blocking of each junction different than 1 and n, and after blocking each edge.

Input

The first line of input contains the number of testcases t ($t \leq 100$). The first line of each testcase contains two integers n and m ($2 \leq n \leq 2 \cdot 10^5, 1 \leq m \leq 10^6$), the number of junctions and the number of roads. The next m lines contain descriptions of the roads. The i-th road is described by three integers x_i, y_i, c_i ($1 \leq x_i < y_i \leq n, 1 \leq c_i \leq 10^9$) that mean that this road has a weight c_i and connects junctions x_i and y_i. You can assume that no two roads connect the same pair of junctions and the whole graph is connected.

It is guaranteed that sum of all n in the input does not exceed $3 \cdot 10^6$, and sum of all m does not exceed $6 \cdot 10^6$.

Output

For each test case output two lines. The first line should cointain m integers seperated by spaces, the i-th integer should be equal to Sherlock's happiness after blocking the i-th road. The second line should cointain $n-2$ integers seperated by spaces, the i-th integer should be equal to Sherlock's happiness after blocking the junction number $i+1$.

Examples

standard input	standard output
2 4 4 1 2 10 1 3 20 2 4 30 3 4 40 4 4 1 2 10 2 3 10 3 4 10 1 3 100	20 0 20 0 20 0 80 80 -1 0 80 -1

Problem F. Epic

Input file:	*standard input*
Output file:	*standard output*
Time limit:	15 second
Memory limit:	512 mebibytes

Jim Moriarty is preparing his final confrontation with Sherlock. As is well known the showdown should take place on a roof of one of London's buildings (where else would one organize the meeting with his archnemesis). Jim has already chosen the street, but he can't quite make up his mind as to the specific roof and so he is gathering information about the buildings. As part of the preparation he had one of his men make a list of the *radii* of the buildings. The radius of the i-th building, denoted as r_i is the largest integer such that the heights of the buildings $i - r_i, i - r_i + 1, \ldots, i + r_i - 1, i + r_i$ are all smaller than the height of the i-th building (it is required, that all these buildings exist, meaning $i - r_i \geq 1$ and $i + r_i \leq n$). The radius is obviously an important aspect of the roof, since if it is too small, one of the nearby buildings will block the view making the confrontation far less epic.

Unluckily for Jim the list has now fallen into Sherlock's hands. Holmes wonders if it is possible to guess which street Moriarty has chosen, by inspecting the radii. He would like to know the relative order of the heights of buildings on that street (obviously there are no two buildings of the same height, so one can interpret this ordering as a permutation of numbers $1, 2, \ldots, n$). Naturally there can be many such fitting permutations and so Sherlock would first like to know the number of possibilities (if it is not too big, he can ask Scotland Yard to help him look for plausible streets). Now, Holmes could obviously do this himself, but he has a big confrontation coming up, so he asked you for help with calculating the number.

Note:It is possible that Jim is just playing games with Sherlock and there is no permutation that fits the found list.

Input

The first line of input contains the number of test cases t ($t \leq 100$). For each test case the first line consists of a single positive integer n ($n \leq 5000$). The second line contains n integers $r_1, r_2, \ldots, r_n$ ($0 \leq r_i \leq 10^9$) separated by single spaces, denoting the radii on the list found by Sherlock.

Output

For each test case in the only line of output you should print the number of permutations fitting the given list, modulo $10^9 + 7$. If there is no such permutation output a single word "NO" instead.

Examples

standard input	standard output
4 6 0 1 0 0 1 0 5 0 0 2 0 0 3 0 1 0 7 0 1 0 3 0 0 0	80 24 2 160

Problem G. Balancing expressions

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 second
Memory limit:	512 mebibytes

You have recently bought a string at a shop with balanced parenthesis expressions. Sherlock Holmes, however, the smart detective that he is, noticed that the string is not actually a balanced parenthesis expression. You are able to fix the string yourself by cutting out some of the characters with a scalpel. You will need c_i minutes to remove the ith character of the string. You wonder how much time you will need to spend to turn your string into a balanced parenthesis expression.

Balanced parenthesis expression is defined recursively as follows:

1. empty string is a balanced parenthesis expression,
2. if S is a balanced parenthesis expression, so is (S),
3. if S and T are balanced parenthesis expression, so is the concatenation ST.

Input

The first line of input contains the number of test cases t ($t \leq 10$). For every test case, the first line of the input contains an integer n ($n \leq 10^6$): the length of your string.
The second line contains a string of length n consisting of characters '(' and ')'.
The third line contains n integers $c_1, c_2, \ldots, c_n$ ($1 \leq c_i \leq 10^9$), the time required to remove each of the characters of your string.

Output

Output should contain one integer, the minimal time in minutes required to convert your string into a balanced parenthesis expression.

Examples

standard input	standard output
2 5 ((()) 1 1 1 1 1 6 ())()(1 1 1 1 1 1	1 2

Problem H. Prepare for unforeseen

Input file:	*standard input*
Output file:	*standard output*
Time limit:	15 seconds
Memory limit:	512 mebibytes

Professor Moriarty wants to prepare for his next encounter with Sherlock. He has built a maze of hideouts connected by one way passages. Now he will send his opponent a letter revealing location of his hideout with a note that he waits for the detective. Right before Sherlock arrives he will change his location, leaving him another note with a riddle. He will continue to do so until he reaches the hideout he has prepared for the last battle. Because he wants riddles to be complicated, he needs to prepare them today. He must however prepare the right number of puzzles. If he runs out of them before he arrives to his destination, he will be angry that he cannot taunt Sherlock anymore and his plan did not succeed. Also reaching the goal without using every hint will make him annoyed as he will have to waste great riddles he had prepared. Unfortunately he does not know, which hideout will be the last one. He will decide that tomorrow. He must however pick the first hideout today, to be able to send the hint to Sherlock. Now he wants to know if its possible to pick a starting position and a number of riddles that no matter which hideout he chooses tomorrow it will be possible to taunt Sherlock into following him. Being great mastermind, Moriarty does not need to prepare puzzles for a particular situation, he will always be able to modify them. He only needs different tricks to be prepared day before. Also he does not care if the path will lead through the same hideout, or passage, multiple times. They are designed in such a way that all of them look exactly the same and Holmes will never know that he has been in one of them multiple times.

Input

The first line contains $T \leq 100$ - number of test cases.
Single test case input:
The first line of input contains two integers: n - number of hideouts and m - number of one way passages. The following m lines describe the one way passages. Each road description consists of 2 integers a_i, b_i ($1 \leq a_i, b_i \leq n$), which mean that there is a one way passage from a_i to b_i. There are no loops ($a_i \neq b_i$), passages do not repeat, but two hideouts might be connected by two passages in different directions.

In every test $n \geq 1$ and there are up to 10^5 vertices and $3 \cdot 10^5$ edges in all the tests.

Output

Single test case output:
In the first line of output write `YES` or `NO`, the answer to Moriartys question: Is it possible to choose starting possition and number of riddles such that no matter which end position he choses tomorrow, he will be able to pick such way from start to end that he uses all the riddles.

Examples

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

Problem I. Scuttling Across Fields

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

After decoding Moriartys map, Sherlock knows where the stolen queens jewelries are. Now he needs to get to it as quickly as possible. The map shows coordinates of the treasure, but Sherlock needs to cross swamps, quick sands and mine fields. Being extremely careful Sherlock wants to avoid different traps, that Moriarty might have prepared.

Land presented by map can be seen as a half-plane containing all non-negative y coordinates. Sherlock currently stays in the point $(0, 0)$, and wants to get to the point (A, B). Whole terrain consists of $n + 1$ parts, divided by lines parallel to X axis. I-th part is between lines $y = y_{i-1}$ and $y = y_i$ (assuming, $y_0 = 0$ and $y_{n+1} = \infty$).

Sherlock knows, judging on the terrain type, that if he wants to be sure to avoid all the traps that might be hidden at the i-th part he cannot move faster than v_i meters per second. Obviously it could not be too easy, so he must go through all the different types of terrain.

Help Sherlock to determine minimal time required to get to the hidden treasure.

Input

The first line contains $T \le 100$ — number of test cases.
Single test case input:
The first line of input contains three integer numbers n, A, B — number of different field parts, and target coordinates. The second line contains n positive integers y_i, such that $y_i < y_{i+1}$ for $i = 1, 2, \ldots, n - 1$. The third lain contains $n + 1$ positive integers v_i — maximal speeds of Sherlock on i-th field part (between$y = y_{i-1}$ and $y = y_i$, where $y_0 = 0$ and $y_{n+1} = \infty$). You can assume that $B > y_n$.

Limits: $A \in [0, 10^6]$, $y_i, v_i, B \in [1, 10^6]$, $n \in [0, 10^5]$, sum of all n in one test does not exceed $7 \cdot 10^5$.

Output

Single test case output:

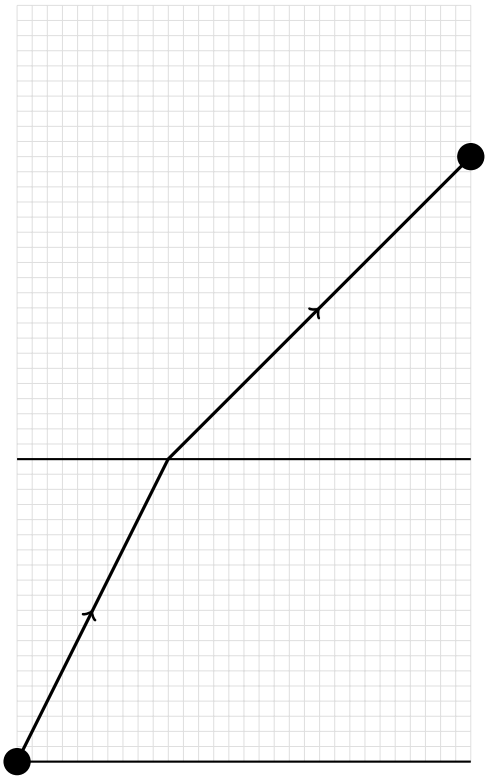
In the only line of output write one number — minimal time Sherlock needs to reach the target. Answer will be accepted if absolute or relative error will be smaller than 10^{-6}.

Examples

standard input	standard output
2 1 30 40 20 12 19 3 45 92 19 34 45 4 5 6 7	3.3520355 18.0406011

Note

Optimal route for the first example is presented on the following illustration:



Problem J. Explosions

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

Evil villains like explosions ans James Moriarty is no exception to this rule. Being a mastermind he has now obtained n nuclear bombs, which he calls $f_1, f_2, \ldots, f_n$. Each bomb has three parameters p_i, q_i and b_i. If at the moment of setting the i-th bomb off there already is an explosion of order x, then after the detonation there will be an explosion of order $f_i(x) = \max(x, \frac{p_i}{q_i}x + b_i)$. He also has one small bomb that will result in an explosion of order 0(the bomb is like, really small). Now he wants to set up some order of the bombs $f_{\pi(1)}, f_{\pi(2)}, \ldots, f_{\pi(n)}$, where π is a permutation of n elements, such that after detonating all the bombs, the last explosion is as big as possible. Since he really likes his small bomb, he already decided that he will detonate it first. Formally he wants to choose a permutation π that maximizes the final result of the following procedure:

1. set $x := 0$
2. for $i = 1, 2, 3, \ldots, n$ set $x := \max(x, f_{\pi(i)}(x))$.

Unfortunately every mastermind has his weak spot, and Jim's is dealing with affine functions and permutations, so he asks you for help with his problem.

Input

The first line of input contains the number of test cases t ($t \leq 100$). For each test case the first line contains one positive integer n ($1 \leq n \leq 2000$). Each of the next n lines contains three integers p_i, q_i, b_i ($-10^9 \leq b_i \leq 10^9, 1 \leq p_i, q_i \leq 10^9$) separated by single spaces — the description of the i-th bomb.

Output

For each test case the only line of output should contain n integers $\pi(1), \pi(2), \ldots, \pi(n)$ separated by single spaces. These numbers should describe a permutation of the bombs that leads to the largest possible last explosion. Note that the small bomb (which gives the explosion of order 0) is always detonated first and you should not include it in your answer. If there are many permutations which lead to the optimal answer, output any of them.

Examples

standard input	standard output
2 3 3 2 -65 4 3 17 10 9 3 3 4 3 7 5 5 -30 12 9 7	2 3 1 3 2 1

Problem K. Bridges

Input file:	*standard input*
Output file:	*standard output*
Time limit:	40 seconds
Memory limit:	512 mebibytes

All of the London's bridges have collapsed in a recent fire and it has been decided that at most k new bridges will be built. Mycroft knows that to ensure the safety of the citizens it is essential that every policeman in the city is able to reach the closest bridge as soon as possible to get to the other side of London. There are n police stations in London, ith of them is located at integer coordinates (x_i, y_i) and having w_i policemen stationed in it. Mycroft needs to know what the minimal sum of distances from each policeman to his closest bridge can be, where the distance between points (x, y) and (x', y') is measured as $|x - x'| + |y - y'|$.
The river is a line that can be described by $y = ax + b$. Every bridge must be a point anywhere on this line, not necessarily at integer coordinates.

Input

The first line of input contains the number of testcases t ($t \le 40$). For every test case, the first line of the input contains integers a and b that define the river ($-100 \le a \le 100$, $-10^9 \le b \le 10^9$).
The second line contains integers n and k denoting the number of police stations and the maximal number of bridges that can be built ($n \le 1000$, $k \le 10^9$).
The next n lines contain three integers x_i, y_i, w_i, respectively the coordinates of the ith station and number of policemen stationed in it ($-10^9 \le x_i, y_i \le 10^9$, $1 \le w_i \le 100$.

Output

The output for each test case should be one number: the minimal possible sum of distances of each policeman to his closest bridge. The result will be considered correct if it differs from the actual one by no more than 0.01.

Examples

standard input	standard output
4 0 0 3 1 -10 10 1 0 10 1 10 10 1 1 0 3 2 6 5 4 0 2 1 2 -1 1 0 4 6 3 -2 4 6 2 6 1 3 2 1 4 6 1 5 2 1 6 0 1 97 0 1 1 23 32 99	50.00 9.00 15.00 2244.34