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 ($t \leq 50$). The first line of each testcase contains two integers n and m ($1 \leq n \leq 35$), 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 $a_1, a_2, \dots, a_k$, 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 $b_1, b_2, \dots b_\ell$. The parts are foldable, so a screen $b_1, b_2, \dots b_\ell$ can cover a graffiti $a_1, a_2, \dots, a_k$ if there exists a sequence of indices $1 \le i_1 < i_2 < \dots < i_k \le \ell$, such that $b_{i_j} \ge a_j$ for all $j = 1, 2, \dots k$. 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 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	80
6	24
0 1 0 0 1 0	2
5	160
0 0 2 0 0	
3	
0 1 0	
7	
0 1 0 3 0 0 0	

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 $T \leq 100$ — 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}.

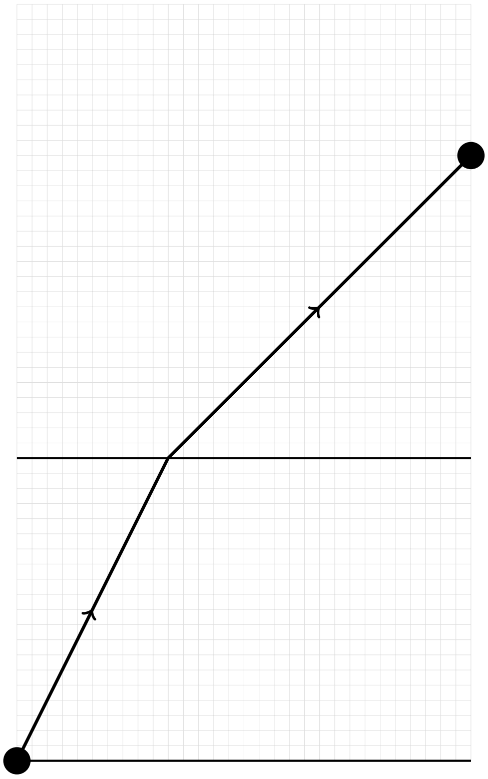
Examples

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 L. Bus

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

Lestrade is on the case of the cheating at the bus payments.

He chose the bus and counted the passengers inside.

The bus fare equals one shilling in London. However, not everything is that easy — **no more than one** child can ride for free with each grown-up passenger. That means that a grown-up passenger who rides with his k ($k > 0$) children, pays overall k rubles: a ticket for himself and $(k - 1)$ tickets for his children. Also, a grown-up can ride without children, in this case he only pays one shilling. Lestrade knows that in England children can't ride in a bus unaccompanied by grown-ups.

Help Lestrade count the minimum and the maximum sum in shillings, that all passengers of this bus could have paid in total.

Input

The input file consists of a single line containing two space-separated numbers n and m ($0 \leq n, m \leq 10^5$) — the number of the grown-ups and the number of the children in the bus, correspondingly.

Output

If n grown-ups and m children could have ridden in the bus, then print on a single line two space-separated integers — the minimum and the maximum possible total bus fare, correspondingly.

Otherwise, print "`Impossible`" (without the quotes).

Examples

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

Problem M. Moriarty's Cypher

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

To ensure that his mathematical papers will not be stolen, professor Moriarty used special cypher for them. More, the same operation scrambles the message as unscrambles it.

This operation is performed by replacing vowels in the sequence

(a i y e o u)

with the vowel three advanced, cyclicly, while preserving case (i.e., lower or upper). Similarly, consonants are replaced from the sequence

(b k x z n h d c w g p v j q t s r l m f)

by advancing ten letters.

The fascinating thing about this transformation is that the resulting language yields pronounceable words, so it looks as the paper is written on some rare language.

For this problem, you will write code to translate Moriarty's papers into plain text.

Input

First line of the input contains integer T ($1 \le T \le 20$) — number of test cases.

Each test case consists of a single line containing up to 100 characters, representing some text written by Moriarty. All characters will be plain ASCII, in the range space (32) to tilde (126), plus a newline terminating each line.

Output

For each input test case, print its translation into plaintext. The output should contain exactly the same number of lines and characters as the input.

Example

standard input	standard output

`1`
`Fokat otnafad T.`
	`Given integer N.`

Problem N. The Magic Number

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

Sherlock is trying to decrypt encrypted letter. He knows open part of the key — integer n. Key k is smallest positive integer such that the decimal representations of the integers in the sequence $n, 2n, 3n, 4n, 5n, \dots, kn$ contain all ten digits (0 through 9) at least once.

Input

First line of the input contains integer T ($1 \le T \le 30$) — number of the testcases. Each of next n lines contains one integer n ($1 \le n \le 2 \cdot 10^8$).

Output

For each test case, print the key k at the separate line.

Example

standard input	textslstandard output
3 1 10 3141592	10 9 5