Sharif Online Contest & Internal Selection Contest of Sharif

17 Aban 1386

Contents

Problem A: A tragicomic party

Last night, I'd been invited to a party. After dinner, the host invited us to do a lottery game and gave each of his N guests (including me) a ticket. Each ticket was a white square piece of paper in which a positive number (with no leading zero) were written by English digits in the center. He told us the numbers on the tickets are distinct numbers in range 1 to N, but I was not sure due to a historical background of his personality!

Moving an eye, I read K tickets of other guests and concluded the original numbers can't be unique numbers in set {1, 2, …, N}. Do you agree me?!

Input

Input consists of $1 \le T \le 100$ datasets, coming one after another. Each datasets begins with two numbers N and K. Thereafter, in the second line, K strings (of digits 0 to 9) comes which are the numbers I saw in guests hands. It's guaranteed that $1 \le K \le N \le 1000$ and no string (in second line of each dataset) has no more than 5 characters.

Output

For each dataset, write "Never trust him, again!" if you agree me that numbers can't be 1 to N or "Calm down, Dude!" if you think I might be wrong.

Sample Input	Sample Output

3
80 3
9 9 81
50 9
1 2 3 4 5 6 7 8 01
69 3
11 11 31
	Calm down, Dude!
	Calm down, Dude!
	Never trust him, again!

Description of sample output

Consider, I may have read the 180° rotated string of a guest! It's, digits 6 and 9 are vertically mirrored of each-other and 0, 1 (which were written as a short line like "|") and 8 are self-mirrored. Thus, the string 81 may be originally 18, but 11 and 31 are always 11 and 31!

Problem B: Boosted Formula

“Shorter! I want shorter Form” whispered professor, the arithmetician. “I’m sure I can represent this formula in a shorter form”, sighed professor with more disappointment than rhythm…

Professor G. has recently found a new formula which is consisted of only capital letters, plus sign, minus sign and parentheses (to change the presidency of operations). Such a formula is $A + B - (C + D)$.

In order to have it in shorter form, Professor has decided to remove all the plus signs from the formula. He believes the original formula can be simply generated from a plus-less one, and luckily he is right – It can be done by putting a plus sign between two consecutive letters or open parenthesis. This way, Above formula would be rewritten as $AB - (CD)$.

“Still lengthy!”, professor believes. He wants to rewrite the formula in the shortest form, but he doesn’t want to change the order of the letters. i.e. he want to acquire the same string when he removes all minus signs and parenthesis from the shortest-form string and the original one. Thus, $ABAF - (DEF)$ is not the shortest form of $A - DBA - (EF)$ and $A - BB$ is different from $AB - B$ (consider same B letters are different!). Also, in order to remove (his imaginary!) ambiguity, he doesn’t want to put a minus sign, right before a letter and uses parenthesis in such case. So, $AB - C$ is an incorrect formula and should be written as $AB - (C)$. As we know, length of a formula is number of its characters, including minus sign and parenthesis.

Input

On the first line of Input, there will be an integer $1 \leq N \leq 50$, number of formulas to be processed. Then, in following N lines, there will be one formula per line. A formula is a valid string of at least one and at most 5,000 characters. Each character is either a capital English letter or a minus sign or an open or a close parentheses. You can assume that all the formulas are valid (i.e. All parenthesis are matching properly).

Output

For each formula, write the length of shortest representation of it in one single line.

Sample Input	Sample Output
3 ABC -(C)-(D)EF-(CC) A-(B)-(CD-(E))-(GH)	3 12 13

Description of sample output

One shortest-representation of sample inputs can be ABC, $-(CD - (EF)CC)$ and $A - (BCD)E - (GH)$, respectively. Note that $A - (BCD - EGH)$ is not a correct formula, because second minus sign has appeared right before E.

Problem C: Cheese and Circular Mouths

N immaculate mice have just escaped from cat-land! After walking for many hours in the desert, they found a 2-dimentional piece of cheese which was in form of a convex polygon with N vertices. Each of them started biting from one vertex of the cheese.

Although, they all had a circular mouth but their biting speed was different. Actually, ith mouse, who starts from point P_i of the polygon, eats the cheese like a growing circle with center of P_i. More precisely, if we assume its biting speed is v_i, then in time t from start its eaten lot of cheese is the common part of the circle with center of P_i and radius of $t \times v_i$ with cheese (if it doesn't share the lot another mouse).

On the other hand, as mice close their eyes while biting, in a case of contact between two or more mice (where they to have the same piece of cheese and their mouths meet), they bite each other and they both (or more) will be died!

Having the cheese shape and biting speeds, find out order of deaths! You may remember cats' survival quote: "*no mouse lives forever*"!

Input

On the first line of Input, there will be number of cheeses to be processed, which is less than 30.

For each cheese, its size $3 \leq N \leq 1{,}000$ is on first line. Thereafter, in any of the next N lines, there's 3 integers X_i, Y_i and v_i, where$(X_i, Y_{i)}$ is the coordinates of ith point of polygon (where mouse #i starts) and $v_i$is the biting speed of mouse #i. The coordinates are at most $100{,}000$ in magnitude and speeds are positive integers less than $1{,}000$. Note that consecutive points of polygon can be possibly in a line.

Output

For each cheese, write order of death of the mice (numbered from 1 to N) in one line. The first number in each line (left most number) is the index of the mouse who died soonest. If two or more mice die at the same time, write them in increasing order of their indices.

Sample Input	Sample Output

2
4
0 0 14
0 1 14
1 1 14
1 0 14
3
0 0 1
5 1 9
9 0 9
	1 2 3 4
	2 3 1

Problem D: Deathful Avalanche

There are some happy skiers on the mountain. They are having fun together, and do not yet know of the avalanche that has just begun.

You, the mountain guard, want to warn them of the avalanche before it gets too late. Each skier is currently moving with a constant speed, either upward or downward.

You expect the avalanche to reach there in one of the times: $t_1, t_2, \cdots, t_n$. So you need to know, at each of those times, the highest height a skier is skiing at.

Input

On the first line there is the number of tests. Then for each test the description of the test comes as follows:

First come two integers m, n how that $1 \leq n, m \leq 100000$. On the next line come m pairs of integers s_i, v_i showing the initial height (at time zero) and the heightening speed of the i^{th} skier. On the next line come the times $t_1, t_2, \cdots, t_n$. These numbers (s_i, v_i and t_i) are integers having absolute value no more than 10000. Also times are always positive.

Output

For each test case print a line containing n integer numbers on a line, corresponding to the highest heights a skier is skiing at, in each of those times.

Sample Input	Sample Output

1
2 3
0 1 2 -1
0 2 4
	2 2 4

Problem E: Etymological Ladder

A word ladder is a sequence of words, in which two consecutive words differ by exactly one letter. An example of such a ladder (usually arranged vertically, hence the "ladder") would be: beer, brew, brow, word, down. Note that to get from one word to the next, the letters may be rearranged, and exactly one letter is changed.

For this problem, you will be given a dictionary of distinct words, all of the same length. Your task is to write a program that finds a word ladder of minimal length, such that the first and last word of the ladder have no letters in common.

Input

On the first line an integer t ($1 \leq t \leq 100$): the number of test cases. Then for each test case:

- A line with two space-separated integers n ($2 \leq n \leq 100$) and l ($1 \leq l \leq 20$): the number of words and their length.
- n lines with a word, each consisting of l lowercase letters (a - z).

Output

For each test case, write a single line with the words in a ladder of minimal length, separated by a single space. It is guaranteed that at least one such ladder can be constructed. If there is more than one, output the one that comes first lexicographically.

Sample Input	Sample Output
1 9 3 alt spy sea opt pea ape spa apt ale	ale alt apt opt

Note

If s and t are strings of equal length and s_i denotes the ith character of s, then s precedes t lexicographically if for some i: $s_i < t_i$ and $s_j = t_j$ for all j < i.

Problem F: Fhitschoo

After an unfortunate explosion in the ACM labs, a new creature came into existence named Fhitschoo. Fhitschoo is very friendly, but at the same time it's very dangerous. It has the ability to divide itself into two identical parts just in no time! (And of course each part is then a complete Fhitschoo)

It has a weakness though; he likes chocolate very much. You have placed n chocolates on the ground, hoping to catch Fhitschoo using them.

You have a gun and you can fire at Fhitschoo, forcing him to divide. When Fhitschoo divides, the two identical pieces start moving in a line in opposite directions. Fortunately you can control the line of movement by aiming your gun at the right position.

When a Fhitschoo comes to a chocolate piece, he starts eating the chocolate, and you can catch him there. Fhitschoos always move in a line with a constant direction, unless hit by gun.

Now you want to find out whether you can catch all the Fhitschoos, or not.

Input

On the first line there is the number of tests. Then for each test the description of the tests comes as follows:

For each test case, there is n, x, y (x, y show the position where the Fhitschoo is standing still at first) and then n pairs of integers follow, showing the positions of chocolates.

We know $1 \leq n \leq 100000$ and Each coordinate is less than 10000 in absolute value.

Output

For each test case either print the word "YES" if you can catch all Fhitschoos, or print "NO" otherwise.

Sample Input	Sample Output

2
4 0 0 1 0 0 1 -1 0 0 -1
3 1 1 0 0 1 0 0 1
	YES
	NO

Problem G: Greed Wheel

There is a game in Casino Royale named Greed Wheel. In this game there is a wheel divided into n sectors. Each sector has a number written on it. There is a pointer initially pointing to the first sector. When you turn the greed wheel, it stops with probability p_i $(1 \leq i \leq n)$ at sector i on the first turn. Note that sum of p_i's is always less than one. If the wheel completes a turn without stopping, then it stops at sector i on the next turn, with some probabilities, proportional to the original p_i's. Thus, if the wheel turns m times without stopping, with probability q, then the probability of stopping at the i^{th} sector on the next turn is $q \times p_i$.

You are going to turn the Greed Wheel two times, and read the two sectors the wheel stops at. If sum of the two numbers written on those sectors is m, then you win m dollars, otherwise you lose d dollars to the Casino. Now you want to know whether playing this game makes profit on the average, or not.

Input

Input consists of several test cases. For each test case the description comes as follows:

First there are two integer numbers n, m and a real number d. On the next line come n integers showing the numbers written on the sectors $1,2, \dots, n$. On the next line come n real numbers $p_1, p_2, \dots, p_n$. You can suppose $1 \leq n \leq 100$ and all other numbers are less than 1000000000. Also p_i's are positive real numbers whose sum is less than one.

The input ends with a line containing three zeros.

Output

For each test case print "YES" if the game makes profit on the average, and otherwise print "NO". Since you are not interested in profits of less than 10^{-4}, you should print "NO" in these cases.

Sample Input	Sample Output

4 10 10.0
5 5 5 5
0.1 0.1 0.1 0.1
4 10 9.99
5 5 5 5
0.2 0.2 0.2 0.2
0 0 0
	NO
	YES

Problem H: Hilbert

Hilbert is a game in which you control a robot on an infinite two-dimensional grid. There are three commands available to you:

- G: Go one square forward in the current direction.
- L: Turn ninety degrees counterclockwise.
- R: Turn ninety degrees clockwise.

After playing this game for a while, you wonder how many squares you can reach within a certain number of moves. Write a program to calculate the answer to this question.

Input

On the first line an integer t ($1 \leq t \leq 100$): the number of test cases. Then for each test case one line comes with an integer n ($0 \leq n \leq 1\,000\,000\,000$), the maximum number of moves.

Output

For each test case, write one line with the number of reachable squares in it.

Sample Input	Sample Output
4 0 1 2 3	1 2 5 11

Problem I: Intergalactic Calendar

Martians have a strange calendar system. They have n different periodic events, i^{th} of which happens every d_i solar days. When they want to address a date, they usually tell the number of solar days passed since the last happening of each event.

But you can't believe, Martians do remember all these events every time they want to address a date. Therefore you asked a Martian to tell you her birth date and her university entrance date. Now you want to analyze the Martian's responses to see whether there is any contradiction or not.

Input

On the first line there is the number of tests. Then for each test the description of the test comes as follows:

First there is the integer $1 \leq n \leq 1000$, followed by the integers d_i. On the next line there are the numbers which address the birth date, and on the next line there are the numbers addressing the university entrance date.

The i^{th} integer used to address a date is always between 0 and $d_i - 1$, inclusive and each d_i fits in a signed integer.

Output

For each case, print a line containing "Contradiction" if contradiction can be found. Otherwise print a line containing "Maybe Right"

Sample Input	Sample Output

2
3 2 3 4
0 0 0
1 2 0
3 2 3 4
0 0 1
1 2 0
	Contradiction
	Maybe Right

Problem J: Jauncing Cycler

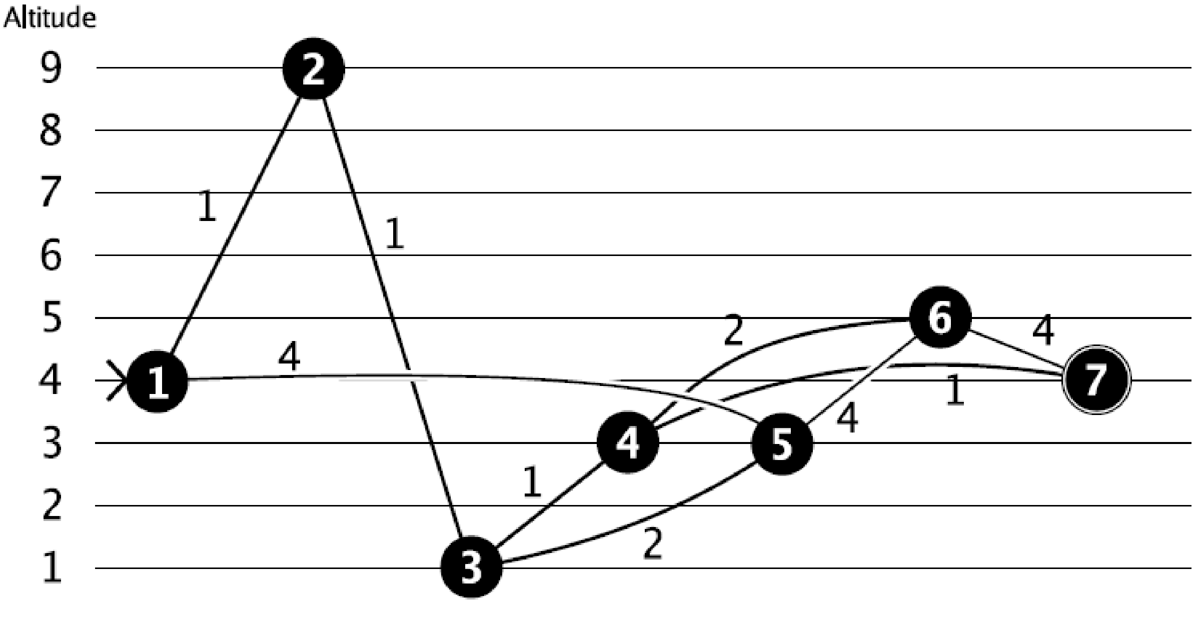
You want to cycle to a programming contest. The shortest route to the contest might be over the tops of some mountains and through some valleys. From past experience you know that you perform badly in programming contests after experiencing large differences in altitude. Therefore you decide to take the route that minimizes the altitude difference, where the altitude difference of a route is the difference between the maximum and the minimum height on the route. Your job is to write a program that finds this route.

You are given:

- the number of crossings and their altitudes, and
- the roads by which these crossings are connected.

Your program must find the route that minimizes the altitude difference between the highest and the lowest point on the route. If there are multiple possibilities, choose the shortest one.

For example consider this circumstance:



In this case the shortest path from 1 to 7 would be through 2, 3 and 4, but the altitude difference of that path is 8. So, you prefer to go through 5, 6 and 4 for an altitude difference of 2. (Note that going from 6 directly to 7 directly would have the same difference in altitude, but the path would be longer!)

Input

- On the first line an integer t $(1 \leq t \leq 100)$: the number of test cases. Then for each test case:
- One line with two integers n $(1 \leq n \leq 100)$ and m $(0 \leq m \leq 5000)$: the number of crossings and the number of roads. The crossings are numbered 1..n.
- n lines with one integer h_i $(0 \leq h_i \leq 1\,000\,000\,000)$: the altitude of the i-th crossing.
- m lines with three integers a_j , b_j $(1 \leq a_j, b_j \leq n)$ and c_j $(1 \leq c_j \leq 1\,000\,000)$: this indicates that there is a two-way road between crossings a_j and b_j of length c_j. You may assume that the altitude on a road between two crossings changes linearly.

You start at crossing 1 and the contest is at crossing n. It is guaranteed that it is possible to reach the programming contest from your home.

Output

For each test case, output one line with two integers separated by a single space:

- the minimum altitude difference, and
- the length of shortest path with this altitude difference.

Sample Input	Sample Output

1
7 9
4
9
1
3
3
5
4
1 2 1
2 3 1
3 4 1
4 7 1
1 5 4
5 6 4
6 7 4
5 3 2
6 4 2
	2 11

Problem K: Knowledge Bonus

Lingo is a once popular game show where the contestants have to guess words. In the original version the contestants would have to guess a five-letter word each round.

In between the rounds of regular word guessing, the contestants can win a bonus prize if they can guess a ten-letter word. The ten-letter word is displayed with the letters permuted. Some letters are colored indicating that they are displayed in the right position. Since there are not that many ten-letter words, it happens frequently that the word is actually a compound: a word constructed by concatenating two shorter words. In this problem we assume that the ten-letter word is always of this form.

Given a dictionary and a sequence of ten letters, you must calculate the possible solutions to the ten-letter word game. Two solutions are considered different if they are constructed from different parts, even if their concatenation is the same. This is illustrated by the the second sample case.

Input

On the first line an integer t ($1 \leq t \leq 100$): the number of test cases. Then for each test case:

- One line with an integer n ($1 \leq n \leq 200$): the number of words in the dictionary.
- n lines with a single word in the dictionary. Each word is between 1 and 9 (inclusive) characters long and consists of only lowercase letters.
- One line with an integer q ($1 \leq q \leq 100$): the number of queries.
- q lines with a single query string. Each query is exactly 10 characters long and will consist of uppercase and lowercase letters. Lowercase letters are in the right position and uppercase letters may be in the wrong position.

All words in the dictionary for a single test case are distinct.

Output

For each test case, output for each query:

- One line with an integer s: the number of solutions.
- min(1000, s) lines, each a solution formatted as two dictionary words separated by a hyphen (-).

The solutions to a single query must be ordered lexicographically. If the number of solutions exceeds 1000, then only output the first 1000 solutions.

Notes

If s and t are strings of equal length and: s_i denotes the ith character of s, then s precedes t lexicographically if for some i: $s_i < t_i$ and $s_j = t_j$ for all $j < i$. In this problem statement, the hyphen precedes all letters lexicographically. For each test

Sample Input	Sample Output
2 5 gunner integral relating tail un 4 TAILGUNNER unINTEGRAL UNrelating IMPOSSIBLE 3 aaaa aaaaa aaaaaa 1 AAAAAAAAAA	6 gunner-tail integral-un relating-un tail-gunner un-integral un-relating 2 un-integral un-relating 1 un-relating 0 3 aaaa-aaaaaa aaaaa-aaaaa aaaaaa-aaaa