PROBLEM A
ABNORMAL 92'S

Once again abnormal freshmen of Amirkabir University of Technology! This time the 92's (those who entered AUT at 1392). They know nothing about ACM/ICPC and think ACM is the abbreviation of "Abnormal Contestants Motivation" that is ICPC! They also think ICPC means "I'm Crazy Professional Coder", so many of them compete in AUT local contest which tends to become a national contest.

Now we want to know how many of 92's are going to compete in the local contest and how many teams only consist of 92's. Ali Seif has given us information of AUT contestants, but we don't have enough time to do the job, so you should calculate the required numbers.

Input (Standard Input)

The first line of input contains an integer T, the number of test cases. Each test begins with an integer n $(1 \leq n \leq 100)$ which is the number of AUT teams. Each of the next N lines contains three integers a, b, c $(37 \leq a, b, c \leq 92)$, the entrance year of team members.

Output (Standard Output)

For each test print two integers in a single line separated by a single space, number of 92's contestants in the contest and number of teams that **only** consist of 92's.

Sample Input and Output

Sample Input	Sample Output
3 3 82 82 84 84 85 82 85 85 85 1 92 92 92 3 88 88 88 92 37 92 92 92 92	0 0 3 1 5 1

PROBLEM B
BUDGET 2

The Amirkabir Annual Programming Contest is held every year to choose teams participating in Tehran regional contest. This year the responsibility for supplying the budget is on Mr. Fakourfar. After spending a lot of time, university authorities agreed to pay some money for the contest's expenditure but it is not sufficient. So Mr. Fakourfar decided to compensate the shortage of the budget with the registration fees, and he wants to set the registration fees so that sum of money earned from registration fees becomes maximum.

According to the capacity of the site, Mr. Fakourfar knows maximum of n teams could participate in the contest. There are k_1 non-AUT teams, k_2 AUT-Senior teams and k_3 AUT-Junior teams willing to attend the contest. If the corresponding registration fee for these teams is c_1, c_2 and c_3, respectively, $k_1 - c_1$ non-AUT teams, $k_2 - 2*c_2$ AUT-Senior teams and $k_3 - 3*c_3$ AUT-Junior teams will finally ask to attend the contest (c_1, c_2 and c_3 should be integers). Now Mr. Fakourfar asks you to find such c_1, c_2 and c_3 so that sum of registration fees becomes maximum. (It is obvious that if $k_i - i * c_i$ becomes negative then nobody of i^{th} group is going to attend)

Input (Standard Input)

In the first line there is T, the number of tests that follow. Each test contains four integers in a single line n, k_1, k_2, k_3 $(1 \le n \le 5000, 1 \le k_1, k_2, k_3 \le 2000)$, as described in the problem statement.

Output (Standard Output)

For each test, print the maximum sum of registration fees Mr. Fakourfar can earn.

Sample Input and Output

Sample Input	Sample Output
4 1 2 3 4 5 6 7 8 20 10 10 10 70 80 100 30	1 15 45 2716

PROBLEM C
CONCATENATION

“A subsequence is a sequence that can be derived from another sequence by deleting some elements without changing the order of the remaining elements. For example, the sequence<A, B, D> is a subsequence of <A, B, C, D, E, F>.”
"The longest common subsequence (LCS) problem is to find the longest subsequence common to all sequences in a set of sequences"
"In formal language theory and computer programming, string concatenation is the operation of joining two character strings end-to-end. For example, the concatenation of "snow" and "ball" is "snowball".”

Wikipedia

Assume ***A*** is a sequence and there exists a set ***S*** of sequences $\{b_1, \ldots, b_m\}$ and an empty sequence ***B***. In each step, we can concatenate one of the sequences from ***S*** to the end of sequence ***B***. Each string of ***S*** can be used more than once. We want to know the minimum number of steps required for which the LCS of sequences ***A*** and ***B*** becomes equal or greater than ***k***.

Input (Standard Input)

In the first line there is T, the number of tests that follow. First line of each test contains none empty sequence A of length at most 100 characters. Second line contains two integers k (less than or equal with length of A) and m $(1 \leq m \leq 50)$. Each of the next m lines contain none empty strings of length at most 50 that shows sequences of S. All sequences in input consist of only lower and upper English letters.

Output (Standard Output)

For each test, print the minimum number of steps required to reach a LCS of length at least k if possible, otherwise print “Impossible”.

Sample Input and Output

Sample Input	Sample Output
4 aabab 4 3 aa bb ab poopiMJpiloop 11 4 poo oop Mp	2 5 Impossible 3

Joil aa 1 1 b aaa 3 1 a	

PROBLEM D
DOE THE KILLER

John Doe the famous killer has escaped prison and is still travelling to different cities killing people because of the seven deadly sins! And the prominent police officer detective Mills is searching for him. As you know, both of them are so clever persons that each of them has inserted a GPS device in the other's cloth so at any moment they know the other person's place.

The country where they live consists of ***n*** cities which are connected with bidirectional roads so that there is at least one path between any two cities. The structure of the roads of this country is as follows: if you assume the cities as graph vertices and roads as graph edges, each edge of this graph is used in at most one cycle, in other words no two cycles of this graph has a common edge. Doe was in the prison of city ***A*** and escaped at night. Next day of his escape detective Mills who lives in city ***B***, was informed of Doe's escape and started pursuing Doe. After that each night Doe stays in his city or escapes to a neighboring city and stays all the following day in that city (until night of the next day which he can move and change his city again). Also each day detective Mills stays in his city or goes to a neighboring city and spends the night there. If Doe and Mills are in the same city at a specific moment, Mills can arrest Doe.

It is obvious that Doe does his best to postpone his arrest and Mills also does his best to arrest Doe as soon as possible. Now you have to find how many days does Mills need to arrest Doe. Assume both Doe and Mills use the best strategy. (If Mills can arrest Doe in the first day, we say one day is required for Mills to arrest Doe)

Input (Standard Input)

The input begins with a line where the number of test cases T is indicated. The first line of each test case contains four integers n, m, A, B $(2 \leq n \leq 50{,}000,\ 1 \leq m \leq 75{,}000,\ 1 \leq A,\ B \leq n,\ A \neq B)$ — number of cities, number of roads, index of the city which Doe was in prison, index of the city which Mills lives in, respectively. Then M lines follow, i^{th} of which contains two integers $u_i,\ v_i$ $(1 \leq u_i,\ v_i \leq n, u_i \neq v_i)$, meaning that there is a bidirectional road between city u_i and city v_i. It is guaranteed that input follows the description of the problem statement.

Output (Standard Output)

Print one line of output for each test, the minimum days required for Mills to arrest Doe or "Infinite" if he can't arrest him.

Sample Input and Output

Sample Input	Sample Output
2 4 4 1 3 1 2 2 3	Infinite 3

3 4 **4 1** **4 3 3 4** **1 2** **2 3** **3 4**	

PROBLEM E
EARTHQUAKE

Bad news! An earthquake is going to shake Tehran. We hope that no one gets hurt but we are sure that it will damage our department's beloved computer site and its network. In this site, there are ***n*** similar computers which are connected to each other using network cables. The network's structure is an undirected, connected acyclic graph (or a tree!). We have also calculated that on the event of a massive earthquake, each computer will survive with a probability of **p** and each cable will survive with a probability of **q**. MeHdi, administrator of the computer site, is trying to solve a problem which has boggled his mind: Assuming after an earthquake, computer department's site consists of a network of only working (survived) computers and also working cables, what is the expected number of connected components in this network? MeHdi is too busy so you should calculate the expected number of connected components!

Note: A Connected component is a sub-graph of all working computers which are connected to each other through a set of working cables and computers.

Input (Standard Input)

The input begins with a line where the number of test cases T is indicated. In the first line of each test there is three numbers. An integer n $(1 \leq \mathrm{n} \leq 1000)$ which is the number of computers and real numbers p (probability of survival of a computer) and q (probability of survival of a cable) are given $(0 \leq p,\ q \leq 1)$. p and q have exactly two decimal digits after the decimal point. Then in each of the next N-1 lines, two integers a and b $(0 \leq a,\ b < n, a \neq b)$ are given which represent a cable between computers a and b in the initial computer site network.

Output (Standard Output)

For each test, print the expected number of connected components of the network after an earthquake, rounded to exactly six digits after the decimal point.

Sample Input and Output

Sample Input	Sample Output
2 2 0.50 0.50 0 1 6 0.30 0.70 0 1 0 2 1 3 1 4 1 5	0.875000 1.485000

Explanation of the first test case :

- With probability of $\frac{1}{4}$ no computer will survive. (0 components)
- With probability of $\frac{1}{2}$ exactly one computer survives. (1 component)
- With probability of $\frac{1}{8}$ both of them survive and the edge between them will be damaged. (2 components)
- With probability of $\frac{1}{8}$ both of them survive and the edge between them survives too. (1 component)

So the expected number of components is: $\frac{1}{4} * 0 + \frac{1}{2} * 1 + \frac{1}{8} * 2 + \frac{1}{8} * 1 = 0.875$

PROBLEM F
FEDERER

In this problem, you will learn more about an amusing and popular sport, tennis. You will have to simulate the scoring system of tennis.

Here is a brief explanation of how the scoring system works. In a tennis **match**, players play a certain number of consecutive **sets**. Each **set** is in turn made up of a series of **games** (and may include a **tie-break** if needed). Finally each game is made of **points**.

Points: Every point is started by one of the players serving (i.e. hitting the ball into the service box in the opposite court) and the other receiving serve. The server then attempts to return the ball into the server's court and players alternate hitting the ball across the net. When one of the players fails to make a legal return (e.g. if the ball is knocked out of the court), he or she loses the point. The specifics of how points are won are not important to us.

Games: The scoring system within a game is peculiar to say the least. As the player wins points in a game, his score goes from the initial value of 0 (read ``love'') to 15, 30, or 40 (yes, just when you think you're starting to spot a pattern in this mess it breaks down). There is no a-priori limit to the length of a game (meaning the number of points played), but a player's score is always indicated by one of these numbers according to the following rules. When a player has three points (score 40) and wins the following point as well, he wins the game unless the scoreline was 40 - 40 (read “deuce”) to start with. A player needs to win **two** consecutive points from deuce to win the game. Winning one gives him **advantage**; if followed by a second winning point the game is won by him, but if followed by a losing point the score reverts to deuce.

Example: at 40 - 30, if the first player wins the next point he wins the game. However, if the second player wins the next three points the game is his.

The scoring of a tie-break game is different, read below for more information.

Sets: A player wins a set if he wins at least six games (in the current set) and he is two games ahead of his opponent but, as you may be starting to suspect, there is yet another exception. In case the scoreline for the number of games won reaches six-all (6 - 6), a tie-break is played instead to decide the set.

Example: at 5 - 4, if the first player wins the next game he takes the set 6 - 4. But if he loses (5 - 5), the set is still undecided and can eventually go to either 7 - 5, 5 - 7 or a tie-break (6 - 6).

Tie-break: During a tie-break game, points are scored “0”, “1”, “2”, “3”, etc. The first player, who wins at least seven points with a margin of two points, wins the “Game” and “Set.

Method of choosing the serving player: In each game, only one of the players serves. (With the exception of tie-break) Also, every two subsequent games the server changes, regardless of whether the set is over or not.

In Tie-Break Game, The player whose turn it is to serve shall serve the first point of the tie-break game. The following two points shall be served by the opponent. After this, each player shall serve alternately for two consecutive points until the end of the tie-break. The player whose turn it was to serve first in the tie-break game shall be the receiver in the first game of the following set.

Piloop, who is friend of the big fan of tennis - MJRS, has left the preparation of the Amirkabir contest and has gone home in a hurry to watch a tennis match between Federer and Nadal. After turning on the television, he realizes that the game had already begun and the score of the first set is ***A:B***, meaning Federer has won ***A*** games while his opponent has won ***B*** games. Unfortunately, technical problems caused the television to stop showing score, but our smart fella, Piloop, started taking notes for each point as soon as he started watching the match. For each point, He wrote down whether the server got the point or the receiver. After the tennis match is over, Piloop has to come back to university to moderate the preparation of the contest, so you have to help him figure out the match's score using his notes.

Note: Assume there is no time or set limit for this match!

Input (Standard Input)

The input begins with a line where the number of test cases *T* is indicated. In the first line of each test, two integers are given: A which is the number of games won by Federer, and B which is the number of games won by Nadal, both at the time of turning on the television.

In the second line, a none empty sequence P of length at most 1000 is given which consists of only 'S' and 'R'. If i^{th} character of P is equal to 'S', the server has got the i^{th} point and if the i^{th} character is equal to 'R' the receiver has got the i^{th} point. We also know that for the current game, Federer serves first. It is guaranteed that all inputs follow the given scoring system rules.

Output (Standard Output)

For each input, print two lines in the output. In the first line, the score of all finished game up until now with the format of A_1-B_1 A_2-B_2 ... should be printed in which A_i-B_i is the score of the match in the i^{th} set. (In the i^{th} set, Federer has won A_i games and Nadal has won B_i games.) In the second line, print the score of the players for the current game (Separated with a space). If it is a tie-break game, print the scores in the form of non-negative integers (0, 1, 2, …) but if it's a normal game, if no deuce has happened yet, print their scores in form of 0, 15, 30, 40, otherwise in case of scores being equal, print "Deuce", or if one of the players has a higher score, print "Advantage: NAME" with NAME being the name of the player who has a higher score (Federer or Nadal). For more detail, read the following samples.

Sample Input and Output

Sample Input	Sample Output
8 1 0	SETS: 1-0 GAME: 30 0

SS 1 0 SRRR 0 1 SSSRRR 4 5 SSSRRRSRS 6 4 SSRRSRR 5 4 SSSS 6 6 SSSSSSSSSRS 5 6 SSSSSRR	SETS: 1-0 GAME: 15 40 SETS: 0-1 GAME: Deuce SETS: 4-5 GAME: Advantage: Federer SETS: 6-4 0-0 GAME: Advantage: Nadal SETS: 6-4 0-0 GAME: 0 0 SETS: 6-6 GAME: 6 5 SETS: 6-6 GAME: 0 3

PROBLEM G
GOLDEN CUP

There is a big show in the national stadium of town for the opening ceremony of FIFA world cup 2014. Many people are performing various programs on the field. To provide the best visual experience, art director of the show is organizing special settings by specifying certain locations of the field that should be illuminated.

The stadium is equipped with three powerful spotlights. Each spotlight is capable of illuminating an angle of 60 degrees which has a fixed vertex at the location of spotlight, but is free to rotate around it. Illumination angles include their sides, i.e. a point strictly on one side of an angle is being illuminated by the corresponding spotlight.

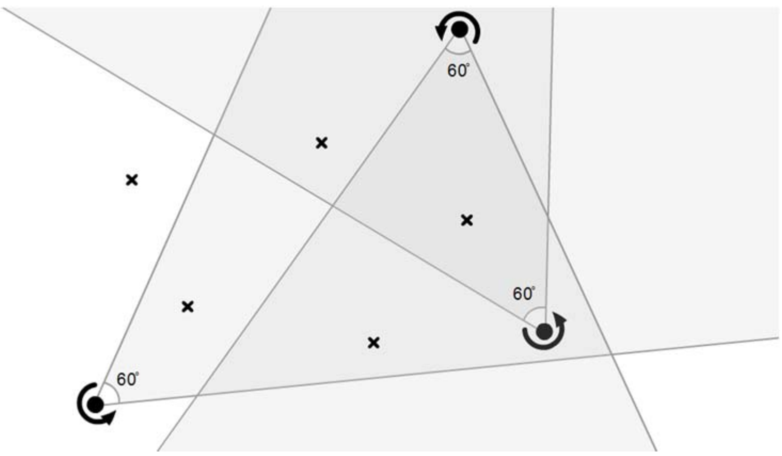


Figure 1- a sample setting of spotlights in a cropped area of the field.

The art director has specified ***n*** distinct points on the field that should be illuminated. But he cannot figure out how to setup rotations of spotlights to illuminate these points. You should write a program to find the maximum number of points that can be illuminated.

Input (Standard Input)

The first line of input contains a single integer T– the number of tests that follow. Description of each test starts with three lines, each containing two space-separated integers $s_x,\ s_y$ – locations of spotlights. Next line contains a single integer n $(1 \le n \le 100)$ – the number of points. Each of the next n lines contains two space-separated integers $x_i,\ y_i$ – the coordinates of i^{th} point.

It is guaranteed that all locations (including spotlights and requirement points) in one setting are distinct and the absolute value of the coordinates does not exceed 10^6.

Output (Standard Output)

For each test, output a single line containing the maximum number of points that art director can illuminate.

Sample Input and Output

Sample Input	Sample Output
1 0 1 1 0 -1 0 4 0 10 0 -10 10 0 -10 0	3

PROBLEM H
HELPING MJRS

These days, MJRS is very busy working on his thesis. His thesis's subject is the problem of conflict-free coloring. The study of this problem is motivated by problem of frequency-assignment to communication antennas so that for each point in the union of areas covered by these antennas, there exists an antenna with a unique frequency that covers that point. The goal is to minimize the number of assigned frequencies (colors).

But the problem MJRS is facing for his thesis is the problem of conflict-free coloring of points on a line. Specifically, Assume there are ***n*** points $x_1 < x_2 < \cdots < x_n$ on the x-axis, in which the ***i^th^*** point is colored with color c_i. This coloring is conflict-free if for each continuous subsequence $C_{ij} = c_i, c_{i+1}, \dots, c_j$ of colors, there exists a point k $(i \leq k \leq j)$, which its color is unique in the C_{ij} sequence. In other word for each m where $i \leq m \leq j$ and $m \neq k$ we have $c_m \neq c_k$. For example, {1, 2, 1} is a conflict-free colored sequence, because every continuous subsequence of it has a unique color.

To complete his thesis, MJRS has to write a program that for each sequence of colors $\{c_1, \dots, c_n\}$, it determines whether it's conflict-free or not. MJRS is too busy working on AUT local preparation, so it's up to you to write this piece of program for him.

Input (Standard Input)

In the first line there is T, the number of tests that follow. The first line of each test case contains an integer n $(1 \leq n \leq 10^5)$, the number of colored points. The second line of the test contains n integers $c_i (1 \leq c_i \leq n)$, which is the color of i^{th} point.

Output (Standard Output)

For each test, if the given colored sequence is conflict-free, print "YES", otherwise print "NO".

Sample Input and Output

Sample Input	Sample Output
2 4 1 2 3 2 7 1 2 3 1 3 2 1	YES NO

PROBLEM I
IRON COLLECTORS

After the presidential election and improvement of the economic situation! poopi wants to retire from ACM and build settlements. He has named this plan Aban Housing. He needs ***K*** kilograms of iron for his first project. In order to do that he has to go to the iron bazaar. There are ***n*** stores in bazaar which sell iron, numbered from ***1*** to ***n*** and store with index ***1*** is the main store of the bazaar. The i^{th} store has w_i kilograms of iron to sell.

There are ***n-1*** alleys in the structure of the bazaar. All alleys of the bazaar are unidirectional and movement is possible in a direction which makes you farther to the main store. All stores are reachable from the main store.

In front of each store there is a worker with a Blue Nissan with a capacity greater than K who can move in alleys, buys some iron from each store on his path, puts it on his Blue Nissan and finally delivers it to poopi's helicopter in front of maybe another store. Salary of a worker is ***P1*** Oshloobs. Now poopi wants to sign contract with some of these workers and use them to collect the required iron. Each time poopi's helicopter lands in front of a store to load the iron from workers' Blue Nissans into his helicopter costs him ***P2*** Oshloobs.

He only could buy a Helicopter from ACM income and wants to save much more money to buy a Boeing 787, so help him to minimize the overall cost of the job.

Input (Standard Input)

In the first line there is T, the number of tests that follow. The first line of each test case contains four integers $n, K, P1, P2$ $(1 \leq n \leq 50{,}000,\ 1 \leq K \leq 10^6,\ 1 \leq P1, P2 \leq 10^6)$—number of stores, required amount of iron in Kilograms, salary for each worker, cost of landing of helicopter(each time), respectively. The Second line contains *n* numbers; the i^{th} number is w_i $(1 \leq w_i \leq 10^6)$. Then $n-1$ lines follow, i^{th} of which contains two integers $u_i,\ v_i$ $(1 \leq u_i,\ v_i \leq n,\ u_i \neq v_i)$, meaning that there is an alley between store u_i and store v_i. It is guaranteed that input follows the description of the problem statement and there is enough iron in the bazaar (in other word sum of iron's in bazaar is more than or equal to K).

Output (Standard Output)

For each test, output a single line containing the minimum amount of money poopi has to pay to buy K kilograms of iron. Assume that poopi buys iron using MeHdi's credit cart and workers don't have to pay the price of the iron they collect. (just consider cost of workers and poopi's helicopter).

Sample Input and Output

Sample Input	Sample Output
2 3 6 1 1	4 8

1 2 3 **1 2** **1 3** **5 10 1 3** **1 5 2 1 2** **1 2** **1 3** **2 4** **2 5**	

Explanation of the second sample input:

- Worker of the store with index 1 collects 3 kilograms of iron
  I. 1 kilogram from store with index 1
  II. 2 kilograms from store with index 3
- Worker of the store with index 2 collects 7 kilograms of iron
  I. 5 kilogram from store with index 2
  II. 2 kilograms from store with index 5

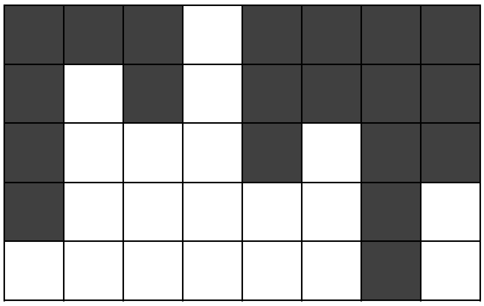
poopi has to land with his helicopter two times in stores with indices 3, 5. So it costs him 8 Oshloobs.

PROBLEM J

JUST NO MORE COUNTERS 2

Once upon a time there was a teacher. He was always telling his students: "You can solve any problem using just counters." The teacher meant that it is better to find a solution using counters even if there are other solutions!

By the way, you are now asked to count the number of completely white rectangles in a grid of **R** rows and **C** columns. Rows of this grid are numbered from ***1*** to ***R*** (up to bottom) and Columns of this grid are numbered from ***1*** to ***C*** (Left to right). In Column j the topmost b_j cells are painted in black, and the remaining cells of that column are white (as shown in the figure).



As you might already know, the 'execution time' is very important in a programming contest so it seems that you cannot use a counter here! Feel free to calculate the answer in any way you want but make it fast enough.

Input (Standard Input)

The input begins with a line where the number of test cases T is indicated. The first line of each test contains two integers R and C $(1 \le R,\ C \le 1000)$, dimensions of the grid. The next line will consist of C integers, the j^{th} number is b_j $(0 \le b_j \le R)$.

Output (Standard Output)

For each test print the number of completely white rectangles in a single line.

Sample Input and Output

Sample Input	Sample Output
3 2 3 0 0 0 2 3 1 0 2 4 4 1 2 3 4	18 5 15