2013 ACM-ICPC Asia Regional Chengdu Online Contest

Problem Set

September 14, 2013

Contents

Problem A. A Game in the Hospital

Mzry1992 got sick recently. He had to go to the hospital for treatment. The time in the hospital was very boring, so Mzry1992 designed a game and wanted to persuade the nurse to play with him.

The game is played as follows: There are n piles of candies on the table. Each pile has a certain number of candies. Mzry1992 and the nurse take turns to eat candies. As the designed of the game, Mzry1992 always starts first. When it's someone's turn to eat candies, he must select a pile of candies (suppose the pile has m candies now), and eat $x(0 < x \leq \lfloor \frac{A \times m + B}{C} \rfloor)$ candies. Otherwise, he loses the game.

("$\lfloor x \rfloor$" indicates the largest integer which is not greater than x)

The nurse loved candies, so she agreed to play with Mzry1992. To start the game, they began to prepare n candy piles. After they had arranged the first $n-1$ piles of candies (where the number of candies in the i_{th} pile was x_i, $1 \leq i \leq n-1$), a new patient came in and the nurse had to go to take care of him. So Mzry1992 was left alone to decide the number of candies in the last pile. But before the nurse left, she demanded that the last pile must have no less than L candies and no more than R candies, i.e. , $L \leq x_n \leq R$.

Mzry1992 agreed with the nurse's demand, since it would be unfair for the nurse if Mzry1992 could place any number of candies in the last pile. Now Mzry1992 wondered, if both he and the nurse play with their best strategies, how many different ways of placing candies in the last pile will make him lose?

Input

The first line has a number T ($T \leq 20$) , indicating the number of test cases.

Then there comes the input of each test case.

The input of each test case has four lines.

The first line contains three integers A, B, and $C(0 \leq A, B \leq 10^{18}$, $1 \leq C \leq 10^{18})$, which are defined above in the description.

The second line contains an integer $n(1 \leq n \leq 10)$, indicating the number of candy piles.

The third line contains $n-1$ integers x_1, x_2, $\cdots$, x_{n-1}, where $x_i(1 \leq x_i \leq 10^{18})$ indicates the number of candies in the i_{th} pile.

The fourth line contains two integers L and $R(1 \leq L \leq R \leq 10^{18})$, which constrains the number of the n_{th} pile in the range $[L, R]$, i.e., $L \leq x_n \leq R$.

We can guaranteed that at any time $\lfloor \frac{A \times m + B}{C} \rfloor$ will no more than 10^5.

Output

For every case, you should output "`Case #t: `" at first, without quotes. The t is the case number starting from 1.

Then follows the answer. which indicates the number of different ways of placing candies in the n_{th} pile that will make Mzry1992 lose.

Sample input and output

Sample Input	Sample Output
1 1 2 3 3 3 4 1 8	Case #1: 1

Problem B. An Easy Problem for Elfness

Pfctgeorge is totally a tall rich and handsome guy. He plans to build a huge water transmission network that covers the whole southwest China. To save the fund, there will be exactly one path between two cities.

Since the water every city provides and costs every day is different, he needs to transfer water from one particular city to another as much as possible in the next few days. However the pipes which connect the cities have a limited capacity for transmission. (Which means the water that transfer though the pipe should not exceed a particular amount) So he has to know the maximum water that the network can transfer in the next few days.

He thought it's a maximum flow problem, so he invites an expert in this field, Elfness (Also known as Xinhang senior sister) to help him figure it out.

Unlike Pfctgeorge, Elfness quickly finds that this problem is much easier than a normal maximum flow problem, and is willing to help Pfctgeorge.

"Oh well, this problem is not a tough one. We can ..."

Abruptly, Pfctgeorge's iPhone rings, and ... the ringtone is *Mo Di Da Biao Ke*.

"You can make that? Excellent! "Pfctgeorge hangs up his iPhone, and turns to Elfness.

"Here's good news for you. A construction team told me that every pipe's capacity can be extended for one day. And the price for extending one unit capacity varies from day to day. "

"Eh well, that's a good news for you, not me. Now it's rather like a minimum cost flow problem, right? But it's still not a tough one, let me have a think. "

After a few seconds' thought, Elfness comes up with a simple solution.

"Ok, we can solve it like... "

Abruptly, here comes *Mo Di Da Biao Ke* again.

"Seriously? You can build new pipes? Thank you very much. "

"OK, my dear Elfness, we got more good news. Another construction team said they can build one or more pipes between any two cities and their pipes are exactly like the original ones except that they only work for one day. And the capacity of the new pipes is only one, but they can be extended, too. Of course, their price to build a single pipe also varies in days. "

"You mean the new pipes can be extended too? Wow, things are getting more interesting. Give me a few minutes. "

Elfness takes out his new ultrabook which is awarded in VK cup and does some basic calculation.

"I get it. The problem can be solved ..."

Mo Di Da Biao Ke again, but this time it's from Elfness's phone.

"As you see, I have to go out. But I know someone else who can also solve this; I'll recommend this guy for you. "

And of course, that poor guy is YOU. Help Pfctgeorge solve his problem, and then the favorability about you from Elfness will raise a lot.

Input

The first line has a number T ($T \leq 10$) , indicating the number of test cases.

The first line of each test case is two integers N ($1 \leq N \leq 100000$) and M ($1 \leq M \leq 100000$), indicating the number of the city that the original network connects and the number of days when Pfctgeorge needs to know about the maximum water transmissions. Then next $N-1$ lines each describe a pipe that connects two cities. The format will be like U, V, cap ($1 \leq U, V \leq N$ and $0 \leq cap < 10000$), which means the *ids* of the two cities the pipe connects and the transmission limit of the pipe. As is said in description, the network that the cities and pipes form is a tree (an undirected acyclic graph).

Then next M lines of the test case describe the information about the next few days. The format is like S, T, K, A, B. S means the source of the water while T means the sink. K means the total budget in the day. A means the cost for a construction team to build a new pipe and B means the cost for a construction team to extend the capacity of a pipe.

I am glad to list the information of building a new pipe and extending the capacity.

1. Pfctgeorge can build a new pipe between any two cities, no matter they have been directly connected or not. Pfctgeorge can build more than one new pipe between any two cities.
2. The capacity of the pipe that was newly built is ***one***.
3. Pfctgeorge can extend the capacity of any existed pipe including the newly built one and the original one.
4. Each time you extend the capacity of one pipe, the capacity of that pipe increases one.
5. The cost of building a new pipe is A and the cost of extending a pipe is B.
6. You can take any constructions in any times and the only limit is to make sure the total costs not exceed the budget.
7. ***All the work that construction team does only lasts one single day***.

Output

For every case, you should output "`Case #t:`" at first, without quotes. The t is the case number starting from 1.

Then for each day, output the maximum water Pfctgeorge can transfer from S and T with a budget of K.

Sample input and output

Sample Input	Sample Output
2 5 1 1 2 2 1 3 5 2 4 1 4 5 2 1 5 3 3 2 5 5 1 2 10 2 3 2 3 4 7 2 5 7 1 5 0 1 3 1 3 0 2 3 1 5 3 2 3 1 2 7 3 1 1 3 2 3 1	Case #1: 2 Case #2: 7 2 8 17 4

Note

In the first sample case, you can extend the capacity of the pipe which connects $city_2$ and $city_4$ by one, or just build a new pipe between $city_2$ and $city_4$.

Problem C. We Love MOE Girls

Chikami Nanako is a girl living in many different parallel worlds. In this problem we talk about one of them.

In this world, Nanako has a special habit. When talking with others, she always ends each sentence with "`nanodesu`".

There are two situations:

If a sentence ends with "`desu`", she changes "`desu`" into "`nanodesu`", e.g. for "`iloveyoudesu`", she will say "`iloveyounanodesu`". Otherwise, she just add "`nanodesu`" to the end of the original sentence.

Given an original sentence, what will it sound like aften spoken by Nanako?

Input

The first line has a number T ($T \leq 1000$) , indicating the number of test cases.

For each test case, the only line contains a string s, which is the original sentence.

The length of sentence s will not exceed 100, and the sentence contains lowercase letters from `a` to `z` only.

Output

For every case, you should output "`Case #t: `" at first, without quotes. The t is the case number starting from 1. Then output which Nanako will say.

Sample input and output

Sample Input	Sample Output
2 ohayougozaimasu daijyoubudesu	Case #1: ohayougozaimasunanodesu Case #2: daijyoubunanodesu

Problem D. Minimum palindrome

Setting password is very important, especially when you have so many "interesting" things in "`F:\TDDOWNLOAD`".

We define the safety of a password by a value. First, we find all the substrings of the password. Then we calculate the maximum length of those substrings which, at the meantime, is a palindrome.

A palindrome is a string that will be the same when writing backwards. For example, `aba`, `abba`, `abcba` are all palindromes, but `abcab`, `abab` are not.

A substring of S is a continous string cut from S. `bcd`, `cd` are the substrings of `abcde`, but `acd`, `ce` are not. Note that `abcde` is also the substring of `abcde`.

The smaller the value is, the safer the password will be.

You want to set your password using the first M letters from the alphabet, and its length should be N. Output a password with the smallest value. If there are multiple solutions, output the lexicographically smallest one.

All the letters are lowercase.

Input

The first line has a number T ($T \leq 15$) , indicating the number of test cases.

For each test case, there is a single line with two integers M and N, as described above. ($1 \leq M \leq 26$, $1 \leq N \leq 10^5$)

Output

For test case X, output "`Case #X: `" first, then output the best password.

Sample input and output

Sample Input	Sample Output
2 2 2 2 3	Case #1: ab Case #2: aab

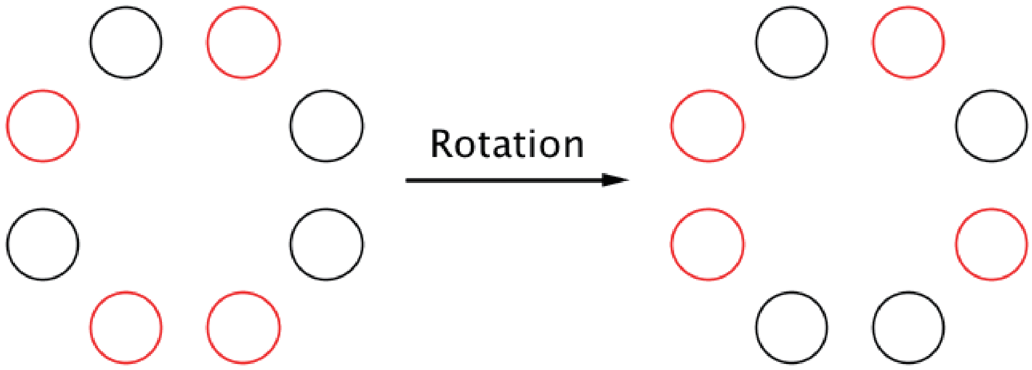
Problem E. Round Table

I have m little buddies, and tonight we will have a fancy dinner in my room.

Fortunately, I have a round table which is large enough for all my little buddies. (As for me, I will not sit in the round table for some reasons) And the round table is so large that I will not let my little buddies sit shoulder to shoulder.

That means I will select m seats from n seats, and there maximal length of consecutive seats in the original round table won't be larger than or equal to k. I want know how many different ways I can choose.

Here is one more thing, two ways are considered the same if and only if one can be obtained by rotation.



The answer may be very large so the answer should modulo $10^9 + 7$.

Input

The first line has a number T ($T \leq 200$) , indicating the number of test cases.

Next each line contain three integer n, m, k ($1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$, $2 \leq k \leq 10^5$, $m \leq n$).

Output

For every case, you should output "`Case #t: `" at first, without quotes. The t is the case number starting from 1.

Then follows the answer, See the sample for more details.

Sample input and output

Sample Input	Sample Output
3 3 1 2 5 2 2 8 3 2	Case #1: 1 Case #2: 1 Case #3: 2

Problem F. G(x)

For a binary number x with n digits $(A_n A_{n-1} A_{n-2} \cdots A_2 A_1)$, we encode it as $G(x) = x \otimes \lfloor \frac{x}{2} \rfloor$. Where "$\otimes$" is bitwise XOR operation and "$\lfloor x \rfloor$" indicates the largest integer which is not greater than x.

Due to some reasons, Mzry1992 encode his password P into $G(P)$, and additionally, he encode $P+1$ into $G(P+1)$ too, and write $G(P)$ and $G(P+1)$ into his diary.

This story happened many years ago and now you hold the diary with these numbers in your hands. Unfortunately, some digits are unreadable now. Could you determine the values of these digits using the readable digits?

Input

The first line has a number T $(T \leq 100)$, indicating the number of test cases.

For every test case, it has 2 lines of same number of digits describe $G(P)$ and $G(P+1)$, In every line, it only contains `1`, `0` and `?`. Unreadable digits are denoted with symbol `?`, The length of every line in the input is up to 10^5.

Output

For every case, you should output "`Case #t: `" at first, without quotes. The t is the case number starting from 1.

Then, if there is impossible to restore $G(P)$ and $G(P+1)$, you should output "`Impossible`" in the second line.

Otherwise, if $G(P)$ is unique, you should output restored $G(P)$ and $G(P+1)$ in the same format.

Otherwise, you should output "`Ambiguous`" and the number of possible $G(P)$ in the second line.

Sample input and output

Sample Input	Sample Output
3 10?? 10?? 0010 0110 1?01 0?01	Case #1: Ambiguous 3 Case #2: 0010 0110 Case #3: Impossible

Note

In the first sample case, the three possible situations are:

- 1010
  1011
- 1011
  1001
- 1001
  1000

Problem G. F(x)

For a decimal number x with n digits $(A_n A_{n-1} A_{n-2} \cdots A_2 A_1)$, we define its weight as $F(x) = A_n \times 2^{n-1} + A_{n-1} \times 2^{n-2} + \cdots + A_2 \times 2 + A_1 \times 1$. Now you are given two numbers A and B, please calculate how many numbers are there between 0 and B, inclusive, whose weight is no more than $F(A)$.

Input

The first line has a number T $(T \leq 10000)$, indicating the number of test cases.

For each test case, there are two numbers A and B $(0 \leq A, B < 10^9)$

Output

For every case,you should output "`Case #t: `" at first, without quotes. The t is the case number starting from 1. Then output the answer.

Sample input and output

Sample Input	Sample Output
3 0 100 1 10 5 100	Case #1: 1 Case #2: 2 Case #3: 13

Problem H. Little Wish ∼lyrical step∼

N children are living in a tree with exactly N nodes, on each node there lies either a boy or a girl.

A girl is said to be protected, if the distance between the girl and her nearest boy is no more than D.

You want to do something good, so that each girl on the tree will be protected. On each step, you can choose two nodes, and swap the children living on them.

What is the minimum number of steps you have to take to fulfill your wish?

Input

The first line has a number T ($T \leq 150$) , indicating the number of test cases.

In a case, the first line contain two number n ($1 \leq n \leq 50$), D ($1 \leq D \leq 10000000$), Which means the number of the node and the distance between the girls and boys.

The next lines contains n number. The i_{th} number means the i_{th} node contains a girl or a boy. (0 means girl 1 means boy), The follow $n-1$ lines contains a, b, w, means a edge connect a_{th} node and b_{th} node, and the length of the edge is w ($1 \leq w \leq 10000000$).

Output

For every case, you should output "`Case #t: `" at first, without quotes. The t is the case number starting from 1.

Then follows the answer, -1 meas you can't comlete it, and others means the minimum number of the times.

Sample input and output

Sample Input	Sample Output
1 3 1 0 0 1 1 2 1 1 3 1	Case #1: 1

Problem I. This Is The Job The Bear Finds

This is the job the bear finds
This is the job the bear finds
This is the job the bear finds
Not with a code but a brick.

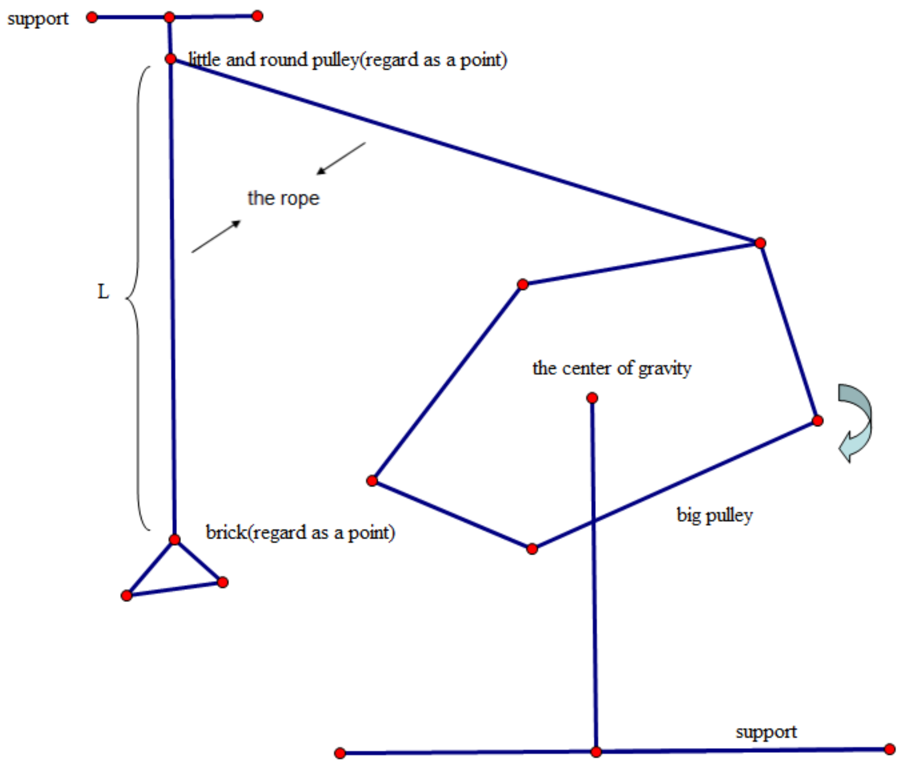
– Kenneth Snow, *The Hollow Bear*

Bearchild is working in a construction team. Life is hard, and his job is just carrying bricks from here to there, what a waste of time! So, our lovely Bearchild has invented a strange machine to help him.

The machine consists three parts: A rope, a little, round pulley (regarded as points) and a big pulley which is a convex hull (Bearchild is short of money and can't make his big pulley round!). One end of the rope is fixed with bricks, the rope goes around the small pulley, and finally connects the big pulley on its contour. It's guaranteed that the small pulley is located to the left of the big one.

You can see that when the big pulley spins clockwise around its center of gravity, the bricks will be pulled up, and finally reach the small pulley as the destination. Note that the rope will circle around the contour of the big pulley as it spins.

See the picture for more information, there will be no collision when spinning.



In this problem, first of all we let the pulley spin for one or more circles so the rope becomes tight. After that, the positions of the two pulleys are given. Then you have to deal a lot of queries, for each query, the distance from the bricks and the small pulley (also as the destination) is given. Your task is to calculate the degree the pulley has yet to spin to pull the bricks to the destination.

Input

The first line has a number T ($T \leq 1000$) , indicating the number of test cases.

For each test case, first line is a number n, which is the number of points of the big pulley (As a convex hull). Then n lines follow, each with two numbers as the x and y coordinate of each point. Those points are given in counter-clockwise order. Then a line with two numbers, which are the x and y coordinate of the small pulley. Those coordinates are in the range $[-1000000, 1000000]$.

Then a number m indicating the number of queries.

For next q lines, each contains a number l ($0 < l < 10^9$), which is the distance between the bricks and the small pulley.

For 90% of the data, we have $n, m \leq 100$; All the data have $n \leq 10^4$, $m \leq 10^5$, and all the numbers in the input are integers.

Output

For every case, you should output "`Case #t:`" at first, without quotes. The t is the case number starting from 1.

Then q lines each answers a query in the Input, which should be printed accurately rounded to three decimals.

Sample input and output

Sample Input	Sample Output
2 4 0 0 1 0 1 1 0 1 -1 2 2 1 4 3 0 0 1 1 -1 1 -2 1 3 1 2 3	Case #1: 90.000 360.000 Case #2: 64.413 140.446 233.660

Problem J. A Bit Fun

There are n numbers in a array, as $a_0, a_1 \cdots, a_{n-1}$, and another number m. We define a function $f(i, j) = a_i|a_{i+1}|a_{i+2}| \cdots |a_j$. Where "|" is the bit-OR operation. $(i \leq j)$

The problem is really simple: please count the number of different pairs of (i, j) where $f(i, j) < m$.

Input

The first line has a number T $(T \leq 50)$, indicating the number of test cases.

For each test case, first line contains two numbers n and m.$(1 \leq n \leq 100000, 1 \leq m \leq 2^{30})$ Then n numbers come in the second line which is the array a, where $1 \leq a_i \leq 2^{30}$.

Output

For every case, you should output "`Case #t: `" at first, without quotes. The t is the case number starting from 1.

Then follows the answer.

Sample input and output

Sample Input	Sample Output
2 3 6 1 3 5 2 4 5 4	Case #1: 4 Case #2: 0