IBBL IUT 3rd National ICT FEST 2011 Programming Contest

Organized by:

[You Get **15** Pages (including this one), **10** Problems and **300** Minutes]

Problemset Credits

Abdullah Al Mahmud	Jane Alam Jan
Kazi Rakibul Hossain	Md. Arifuzzaman Arif
Md. Mahbudul Hasan	Md. Towhidul Islam Talukder
Shahriar Manzoor	Shahriar Rouf Nafi
Shamim Hafiz	Sohel Hafiz

Clarifications

- The time limits of the problems will be given shortly.
- The link for the rank list will also be given shortly.
- Don't use any **space** in your submitted file name. You may get compilation error.
- Don't use **freopen** in your source code. You may get wrong answer. Your sources should take input from the standard input and output in standard output.
- You can send your clarifications through PC^2. But be careful, questions should be related to the problem set. If any question is found that hurts the sentiment of the judges, the team may be disqualified or penalized.
- For other queries, ask the volunteers (foods, printing facility).
- Be careful about output formatting, if your output format is not correct, you may bet 'Output Format Error' from the judges. It doesn't necessarily mean that your code is correct except the output format.

A Story of Tomisu Ghost

It is now 2150 AD and problem-setters are having a horrified time as the ghost of a problem-setter from the past, Mr. Tomisu, is frequently disturbing them. As always is the case in most common ghost stories, Mr. Tomisu has an unfulfilled dream: he had set 999 problems throughout his whole life but never had the leisure to set the 1000th problem. Being a ghost he cannot set problems now so he randomly asks problem-setters to complete one of his unfinished problems. One problem-setter tried to convince him saying that he should not regret as 999 is nowhere near 1024 (2^{10}) and he should not worry about power of 10 being an IT ghost. But the ghost slapped him hard after hearing this. So at last one problem setter decides to complete his problem:

"**n!** (factorial **n**) has at least **t** trailing zeroes in **b** based number system. Given the value of **n** and **t**, what is the maximum possible value of **b**?"

Input

Input starts with an integer **T (≤ 4000)**, denoting the number of test cases.

Each case contains two integers **n (1 < n ≤ 10^5)** and **t (0 < t ≤ 1000)**. Both **n** and **t** will be given in decimal (base **10**).

Output

For each case, print the case number and the maximum possible value of **b**. Since **b** can be very large, so print **b** modulo **10000019**. If such a base cannot be found then print **-1** instead.

Sample Input	Output for Sample Input
4 1000 1000 1000 2 10 8 4 2	Case 1: -1 Case 2: 5227616 Case 3: 2 Case 4: 2

Problem Setter: Shahriar Manzoor, Special Thanks: Jane Alam Jan, Md. Towhidul Islam Talukder

B Fukushima Nuclear Blast

One of the most disastrous nuclear accidents in history has taken place at Japan's Fukushima Daiichi nuclear plant. Reactor buildings have been rocked by explosions, caused after damage was sustained from a massive earthquake and tsunami on 11th march and thus releasing dangerous radiation of unspecified proportions into the air.

The radiation is spreading and it's not a threat only to Japan but also to other countries. If the level of radiation reaches a high level, it will surely cause harms to human health as well as the environment.

You, one of the great programmers in the city, have planned to find the zones that are vulnerable to radiation. If you can measure the level of radiation in a certain city, further actions can be taken to prevent the people from radiation related health consequences. So, at first you want to find the time in which a certain percentage of an area is under radiation threat.

So, at first you modeled the map of a city in **2D** space as a **simple polygon** [1]. You denoted the origin of the explosion as a single point. From this origin, the radiation spreads circularly. You plotted the map such that in each unit of time the radius of the radiation grows one unit. Now, you want to find the time when **P%** of the area of a city is under threat of radiation.

Input

Input starts with an integer **T (≤ 70)**, denoting the number of test cases.

Each test case starts with a blank line. The next line contains an integer **n (3 ≤ n ≤ 5000)** denoting the number of vertices of the polygon. Each of the next **n** lines will contain two integers $\mathbf{x_i}$, $\mathbf{y_i}$ denoting the co-ordinate of a vertex of the polygon. The vertices will be given in anticlockwise order. The next line contains 3 integers $\mathbf{r_x}$, $\mathbf{r_y}$ and **P (0 < P ≤ 100)**, where $\mathbf{(r_x, r_y)}$ denotes the co-ordinate of the origin of explosion and **P** denotes the percentage. All values for the co-ordinates are between **-200** to **200** (inclusive).

Output

For each case, print the case number and the time when exactly **P%** of the total area is under threat of radiation. Round the time to nearest integer.

Sample Input	Output for Sample Input
2 3 -5 0 5 0 0 5 0 0 100 4 0 0 5 0 3 1 5 6 0 0 17	Case 1: 5 Case 2: 2

Notes

1. In geometry, a **simple polygon** is a closed polygonal chain of line segments in the plane which do not have points in common other than the common vertices of pairs of consecutive segments.
2. The judge data is prepared such that the result (before rounding) will not be between **x.3 to x.7**. For example, the result can be like 5.8, 24.3 or 81.791. But the result will not be like 24.5, 78.41, etc.

Problem Setter: Jane Alam Jan, Special Thanks: Shahriar Rouf Nafi, Md. Mahbubul Hasan

C Hamming Base

You are given **N** integers in base-**N** each of them having exactly **M** digits (may be with some leading zeros). Two integers are called **K**-similar if they have the same digits in exactly **K** positions. For example 321 and 213 are **0**-similar. 3456 and 6453 are **2**-similar, 123 and 453 are **1**-similar. You want to change these given **N**-integers in such a way that each pair of these integers are **0**-similar. To achieve this goal you can change the integers in several steps. In a single step you can change a single digit of a single integer by 1 (incrementing or decrementing). But you can't decrement if the digit is **0** or you can't increment if the digit is **N-1**.

You need to achieve your goal in minimum number of steps.

Input

Input starts with an integer **T (≤ 50)**, denoting the number of test cases.

Each case starts with a line containing two integers **N (2 ≤ N ≤ 2000)** and **M (1 ≤ M ≤ 10)**. Each of the next **N** lines contains **M** integers between **0** and **N-1** inclusive. These **M** integers form an **M** digit number in base **N**.

Output

For each case, print the case number and the minimal steps required to achieve your goal.

Sample Input	Output for Sample Input
2 3 3 0 0 0 0 0 0 0 0 0 4 2 0 0 0 0 0 2 2 0	Case 1: 9 Case 2: 8

Problem Setter: Abdullah Al Mahmud, Special Thanks: Jane Alam Jan, Md Towhidul Islam Talukder

D Mad Scientist

Sofdor Ali, the mad scientist, is a well known figure for his great inventions. He sent me a telegram saying, 'urgent need, write the code and send me the output!' And there was a pseudo code in the telegram. Finally, I managed to write it in **C++** (my apology to non C programmers). Here is the code:

```
#include <cstdio>
#include <algorithm>
using namespace std;

typedef long long i64;

const i64 MOD = (1 << 30);

void solve( i64 n, i64 m ) {
	i64 res1 = 0, res2 = 0, k, i;

	for( k = 0; k <= m; k++ ) {
		for( i = 0; ; i++ )
			if( ( n ^ i ) == k ) // n ^ i means n xor i
				break;
		res1 = max( res1, i );
	}
	for( i = 0; i <= res1; i++ ) {
		if( ( n ^ i ) > m ) {
			res2 += ( n ^ i );
			res2 %= MOD;
		}
	}
	printf("%lld %lld\n", res1, res2);
}

int main() {
	int T, t;
	i64 n, m;

	scanf("%d", &T);
	for( t = 1; t <= T; t++ ) {
		scanf("%lld %lld", &n, &m);
		printf("Case %d: ", t);
		solve( n, m );
	}
	return 0;
}
```

But soon, I found that for inputs, as specified in the telegram, the code takes huge time to compute the result. And according to the structure of the code, it's clear that for some cases it may take years to compute the output.

So, I am asking you to help me optimizing the code such that it can generate the output in reasonable amount of time.

Input

Input starts with an integer **T (≤ 10000)**, denoting the number of test cases.

Each case starts with a line containing two integers **n, m (0 ≤ n, m ≤ 2^{60})**.

Output

For each case, print the output generated by the code.

Sample Input	Output for Sample Input
3 0 0 5 3 10 13	Case 1: 0 0 Case 2: 7 22 Case 3: 15 29

Problem Setter: Kazi Rakibul Hossain, Special Thanks: Jane Alam Jan, Md. Arifuzzaman Arif

E Corrupted Friendship

Many of us know a person named Mr. Haba Kom Khan. He is very corrupted and lazy. He maintains links with a lot of (corrupted) persons. In order to corrupt more lazy persons, he invites his friends in a recursive fashion in a meeting. The idea is:

If X has some cards, he keeps one card and he invites one of his friends Y who has no invitation cards yet and X gives all the remaining cards to Y. Then we say that Y is invited by X. Y then has the responsibility to distribute as many cards as he can, and Y keeps one card and continues the same procedure as X. When Y cannot find any more friends to give the invitation cards, he gives the remaining cards back to X. X then checks for his friends who have no cards yet. If any such friend is found, X continues the same procedure. Otherwise, if X was invited by Z then X gives the remaining cards back to Z. X keeps the cards if there is no such person.

Initially Mr. Haba has **N** invitations cards, and he starts the invitation process as stated. There are **N** persons, and they are numbered from **1** to **N**. Mr. Haba is the person numbered **1**. And the strange fact is that, after the invitation process, each person gets exactly **1** card.

Given all the information of persons being invited by others, Mr. Haba wants you to find the total number of invitations that was made. He also asks you to find the number of different pairs of persons who are certainly not friends. Help Mr. Haba Kom Khan to succeed in his corrupted life!

Input

The first line of input will contain **T (≤ 30)** denoting the number of cases.

Each case starts with an integer **N ($1 \le N \le 10^5$)**. Each of the next **N-1** lines will contain two integers, **X** and **Y ($1 \le X, Y \le N, X \ne Y$)** denoting that person **Y** received his invitation card from person **X**. Input is huge. So, faster I/O methods (e.g. scanf, printf, BufferedReader, BufferedWriter) are recommended.

Output

For each case, print the case number, total number of invitations made, and the number of different pairs of persons who are surely not friends. See samples for detailed formatting.

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

Problem Setter: Shahriar Rouf Nafi, Special Thanks: Jane Alam Jan

F Fantasy Cricket

ICC World Cup 2011 has just finished. During the world cup, lots of cricket fans were playing an online game named "Fantasy Cricket".

The score board of fantasy cricket looks like the following image. After each match of the world cup the score board of fantasy cricket updates.

		Manager	Team Name	Total
1	▶	Mind the Gap	Mind the Gap XI	12384
2	▶	Shoeb Saleheen	Aslan's XI	12344
3	▶	Ashik Khan	Shera XI	11476
4	▲	tanim jubaer	crimson_tide	11302
5	▲	S.m.nahid Mahmud	salamance	11168
6	▼	raihan reza	Invincible	10924
7	▼	Munir Hassan	M's XI	10849
8	▶	Saifullah Ali	The Abnormaldehyde's XI	10308
9	▲	sajid sharlemin	THE RAGGED	10008
10	▲	raihan mallick	Bagh_Ailo	9992

Figure 1

Each player plays a role of a manager here. In the rank list there is a symbol besides each manger. There are three kinds of symbols. These are explained in the following table.

Symbol	Explanation	ASCII Representation
	The rank of the player has upgraded after last match, i.e. if the current rank of the player is **K**, the rank of the player before the last match was greater than **K**.	**U**
	The rank of the player has downgraded after the last match, i.e. if the current rank of the player is **K**, the rank of the player before the last match was less than **K**.	**D**
	The rank of the player has not changed after the last match, i.e. if the current rank of the player is **K**, the rank of the player before the last match was also **K**.	**E**

You will be given such a score board after a particular match. Can you determine any possible valid ordering of the players exactly before that round? For this problem you have to print the number of possible ordering before the last match.

Here is an example –

Rank		Manager
1	▲	A
2	▼	B
3	▲	C
4	▼	D

Figure 2

Possible Previous Order 1	Possible Previous Order 2
B	B
A	D
D	A
C	C

Figure 3

For this rank (figure 2), only two different ordering are possible (figure 3) before the last match which comply the current ordering with the symbols.

Name of the managers are not important for this problem. So, for a scoreboard, you will be given a sequence of ASCII representation of the symbols (stated above), i.e. you will be given a string which only contains **'U', 'D'** and **'E'**.

Input

Input starts with an integer **T (≤ 300)**, denoting the number of test cases.

Each case starts with a line containing a string. The length of the string will be between **1** and **1000**. The string will contain characters from **{'U', 'D', 'E'}**.

Output

For each case, print the case number and the number of possible orderings modulo **1000000007**.

Sample Input	Output for Sample Input
3 UDUD EEE DU	Case 1: 2 Case 2: 1 Case 3: 0

Problem Setter: Md. Arifuzzaman Arif, Special Thanks: Jane Alam Jan, Kazi Rakibul Hossain

G Weird Advertisement

Renat Mullakhanov (rem), one of the most talented programmers in the world, passed away on March 11, 2011. This is very sad news for all of us. His team went to ACM ICPC World Finals - 2004, placed 4th and won gold medals. He really was a great programmer. May he rest in peace. This problem is dedicated to him.

2DPlaneLand is a land just like a huge **2D** plane. The range of **X** axis is **0** to 10^9 and the range of **Y** axis is also **0** to 10^9. People built houses only in integer co-ordinates and there is exactly one house in each integer co-ordinate.

Now **UseAndSmile** Soap Company is launching a new soap. That's why they want to advertise this product as much as possible. So, they selected **n** persons for this task. Each person will be given a rectangular region. He will advertise the product to all the houses that lie in his region. Each rectangular region is identified by **4** integers x_1, y_1, x_2 and y_2. That means this person will advertise in all the houses whose **x** co-ordinate is between x_1 and x_2 (inclusive) and **y** co-ordinate is between y_1 and y_2 (inclusive).

Now after a while they realized that some houses are being advertised by more than one person. So, they want to find the number of houses that are advertised by at least **k** persons. Since you are one of the best programmers in the city; they asked you to solve this problem.

Input

Input starts with an integer **T (≤ 13)**, denoting the number of test cases.

Each case starts with a line containing two integers **n ($1 \le n \le 30000$), k ($1 \le k \le 10$)**. Each of the next **n** lines will contain **4** integers x_1, y_1, x_2, y_2 **($0 \le x_1, y_1, x_2, y_2 \le 10^9$, $x_1 < x_2$, $y_1 < y_2$)** denoting a rectangular region for a person.

Output

For each case, print the case number and the total number of houses that are advertised by at least **k** people.

Sample Input	Output for Sample Input
2 2 1 0 0 4 4 1 1 2 5 2 2 0 0 4 4 1 1 2 5	Case 1: 27 Case 2: 8

Problem Setter: Jane Alam Jan, Special Thanks: Md. Mahbubul Hasan

H A Change in Thermal Unit

Measuring temperature and temperature differences are common task in many research and applications. Unfortunately, there exists more than one unit of measuring temperatures. This introduces a lot of confusion at times. Two popular units of measurements are Celsius(**C**) and Fahrenheit (**F**). The conversion of **F** from **C** is given by the formula:

$$\mathbf{F} = \frac{\mathbf{9}}{\mathbf{5}}\,\mathbf{C} + \mathbf{32}$$

In this problem, you will be given an initial temperature in **C** and an increase in temperature in **F**. You would have to calculate the new temperature in **C**.

Input

Input starts with an integer **T (≤ 100)**, denoting the number of test cases.

Each case contains a line with two integers **C** and **d (0 ≤ C, d ≤ 100)**, where **C** represents the initial temperature in Celsius and **d** represents the increase in temperature in Fahrenheit.

Output

For each case, print the case number and the new temperature in Celsius after rounding it to two digits after the decimal point.

Sample Input	Output for Sample Input
2 100 0 0 100	Case 1: 100.00 Case 2: 55.56

Problem Setter: Shamim Hafiz, Special Thanks: Sohel Hafiz, Jane Alam Jan

I Prime Independence

A set of integers is called prime independent if none of its member is a prime multiple of another member. An integer **a** is said to be a **prime multiple** of **b** if,

a = b × k (where **k** is a prime [1])

So, **6** is a prime multiple of **2**, but **8** is not. And for example, **{2, 8, 17}** is prime independent but **{2, 8, 16}** or **{3, 6}** are not.

Now, given a set of distinct positive integers, calculate the largest prime independent subset.

Input

Input starts with an integer **T (≤ 25)**, denoting the number of test cases.

Each case starts with an integer **N (1 ≤ N ≤ 40000)** denoting the size of the set. Next line contains **N** integers separated by a single space. Each of these **N** integers are distinct and between **1** and **500000** inclusive.

Output

For each case, print the case number and the size of the largest prime independent subset.

Sample Input	Output for Sample Input
3 5 2 4 8 16 32 5 2 3 4 6 9 3 1 2 3	Case 1: 3 Case 2: 3 Case 3: 2

Notes

1. An integer is said to be a prime if it's divisible by exactly two distinct integers. First few prime numbers are **2, 3, 5, 7, 11, 13, ...**

Problem Setter: Abdullah Al Mahmud, Special Thanks: Jane Alam Jan

J Save from Radiation

Most of you are aware of Nuclear Power Plant Explosion at Fukushima after devastating earth quake and tsunami. Many people in Bangladesh were seen to be concerned with radiation. The message says:

BBC Flash news: Japan Government confirms radiation leak at Fukushima nuclear plants. Asian countries should take necessary precautions. If rain comes, remain indoors first 24 hours. Close doors and windows. Swab neck skin with betadine where thyroid area is, radiation hits thyroid first. Take extra precautions. Radiation may hit Philippine at around 4 pm today. If it rains today or in the next few days in Hong Kong, do not go under the rain. If you get caught out, use an umbrella or raincoat, even if it is only a drizzle. Radioactive particles, which may cause burns, alopecia or even cancer, may be in the rain.

Many people suggested many things. We, the programmer society, were not inactive. We uploaded a picture in a very popular social network website, facenote and said, "There are two ways to be safe from radiation. One is the Disky way and another is the Buseen way." You may get clear idea of those two ways from the picture.

Anyway, after the explosion of Fukushima Nuclear Power Plant a bottle of water was sent to my laboratory for experiment. But my stupid robot assistant kept it with **N** identical water bottles of which I can't distinguish the special water bottle which came from Japan. I know that if a rat drinks the water from the special bottle, it will die in the 5^{th} minute. Now, I want to identify the special bottle. But I do not like rat either. So, I want to buy minimum number of rats. Can you help me to find out the minimum number of rats I need for identifying the special bottle in 5 minutes?

I forgot to tell you that, only one drop of water from a bottle is enough to find out if the water is safe or not. And you may also give waters from several bottles to a single rat. If one of these bottles is the special bottle then the rat will die, otherwise it will not. And assume that the time to give waters to the rats is negligible, because once you decide the strategy; you may ask the robot assistant to do it. And the robot can do it in no time. And you can also assume that the bottles contain sufficient amount of waters.

Input

Input starts with an integer **T (≤ 3000)**, denoting the number of test cases.

Each test case starts with a line containing an integer **N ($0 \le N \le 10^{16}$)**.

Output

For each case, print the case number and the minimum number of rats required for identifying the special bottle.

Sample Input	Output for Sample Input
2 1 2	Case 1: 1 Case 2: 2

Notes

1. For case 2, the robot assistant kept the special bottle with 2 identical bottles. So, there are 3 bottles, one of them is the special bottle. The minimum number of rats is 2. Because just give a drop from bottle 1 to the 1st rat and a drop from bottle 2 to the 2nd rat. If the 1st rat dies, so it's clear that bottle 1 is the special one. If the 2nd rat dies, then the second bottle is the special one. And if none of them dies, then the 3rd bottle is the special bottle. So, 2 rats are enough. But there are other options, too. Like give the first rat two drops from bottle 1 and 2. And give the second rat a drop from bottle 2. Then the special bottle can also be determined. So, there can be many options, but you need at least 2 rats.

Problem Setter: Md. Mahbubul Hasan, Special Thanks: Jane Alam Jan, Md Towhidul Islam Talukder