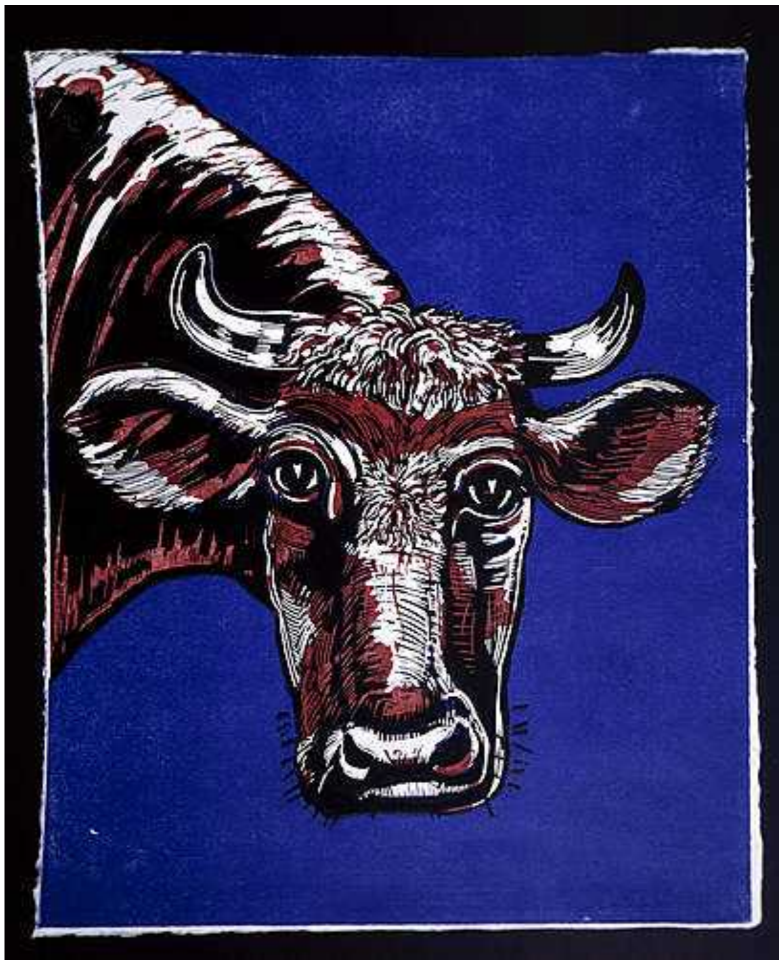
Problem E
Eat or not to Eat?

Input: Standard Input
Output: Standard Output

A young farmer has N cows, but they produced really really a very very small amount of milk. John cannot live on the milk they made, so he's planning to eat some of the 'worst' cows to get rid of hunger. Each day, John chooses the cow that produces the LEAST amount of milk on that day and eat them. If there are more than one cow with minimal milk, John will be puzzled and will not eat any of them (Yeah! That's GREAT!!).



The i−th cow has a cycle of production Ti. That means, if it produces L unit milk on one day, it will also produce L unit after Ti days −− If it will not be eaten during these day :−). Though John is not a clever man, he doubts whether the cows will be eventually eaten up, so he asks for your help. Don't forget that he will offer you some nice beef for that!

Input

The first line of the input contains a single integer T, indicating the number of test cases. ($1<=T<=50$) Each test case begins with an integer N($1<=N<=1000$), the number of cows. In the following N lines, each line contains an integer T_i($1<=T_i<=10$), indicating the cycle of the i−th cow, then T_i integers M_j($0<=M_j<=250$) follow, indicating the amount of milk it can produce on the j−th day.

Output

For each test case in the input, print a single line containing two integers C, D, indicating the number of cows that will NOT be eaten, and the number of days passed when the last cow is eaten. If no cow is eaten, the second number should be 0.

Sample Input

```
1
4
4 7 1 2 9
1 2
2 7 1
1 2
```

Sample Output

```
2 6
```

Rujia Liu

Problem G
Truckin'

Input: Standard Input
Output: Standard Output
Time Limit: 8 Seconds

A truck driver has a regular route down a long stretch of highway. There are intersections with traffic lights at sections along this highway. He would like to plan his trip so that he does not spend any time waiting at red lights. He knows the period of each traffic light, so this is simple enough. However, fuel is costly, so he also wants to conserve fuel as much as possible. Help him determine the minimum amount of fuel required for the journey.

When driving at a constant speed of v meters per second, the rate of fuel consumption is assumed to be **v + 1/v - 0.1** milliliters per second.

Traffic lights stay red for a certain length of time, then turn green for a certain length of time, and repeat indefinitely. When the driver starts out, all lights are just starting their red cycle.

Input

Every case begins with a line containing integers **D**, the distance to travel in meters, and **L**, the number of traffic lights along the route. The next **L** lines will each contain three positive integers **d, r,** and **g**. **d** is the distance along the route of this light, and **r** and **g** are the length in seconds of the red and green cycles, respectively.

No route will be longer than **10000** meters, and there will never be more than **50** traffic lights along a route. Complete light cycles always take between **30** and **90** seconds. Traffic lights will be specified in order of distance along the route (and will not overlap each other or the beginning and end of the route).

Input will be terminated with a line containing two 0s. There will be at most **15** test cases.

Output

For each case, output one line consisting of the minimum fuel required (in milliliters) to make the journey as described, accurate to two decimal places.

Sample Input	Output for Sample Input
1000 3 250 20 20 500 40 40 750 19 19 0 0	998.03

Problem setter: Derek Kisman, Member of Elite Problemsetters' Panel

Problem H
Organizing the Organization

Input: Standard Input
Output: Standard Output
Time Limit: 2 Second

I am the chief of the Personnel Division of a moderate-sized company that wishes to remain anonymous, and I am currently facing a small problem for which I need a skilled programmer's help.

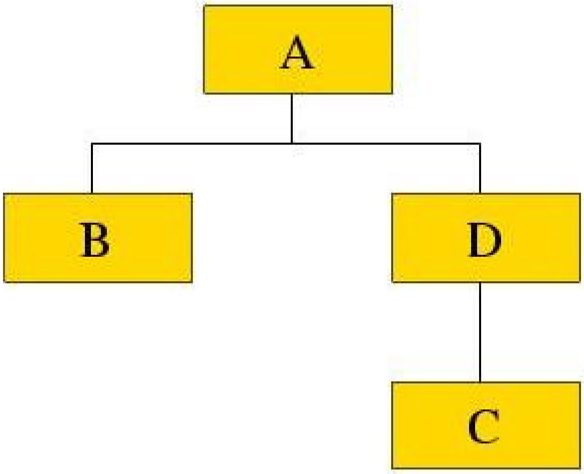


Figure 1: Example organisation

Currently, our company is divided into several more or less independent divisions. In order to make our business more efficient, these need to be organised in a hierarchy, indicating which divisions are in charge of other divisions. For instance, if there are four divisions A, B, C and D we could organise them as in Figure 1, with division A controlling divisions B and D, and division D controlling division C.

One of the divisions is Central Management (division A in the figure above), and should of course be at the top of the hierarchy, but the relative importance of the remaining divisions is not determined, so in Figure 1 above, division C and D could equally well have switched places so that C was in charge over division D. One complication, however, is that it may be impossible to get some divisions to cooperate with each other, and in such a case, neither of these divisions can be directly in charge of the other. For instance, if in the example above A and D are unable to cooperate, Figure 1 is not a valid way to organise the company.

In general, there can of course be many different ways to organise the organisation, and thus it is desirable to find the best one (for instance, it is not a good idea to let the programming people be in charge of the marketing people). This job, however, is way too complicated for you, and your job is simply to help us find out how much to pay the consultant that we hire to find the best organisation for us. In order to determine the consultant's pay, we need to find out exactly how difficult the task is, which is why you have to count exactly *how many* different ways there are to organise the organisation.

Oh, and I need the answer in five hours.

Input

The input consists of a series of test cases, at most **50**, terminated by end-of-file. Each test cases begins with three integers **n, m, k (1 n 50, 1 m n, 0 k 1500)**. **n** denotes the number of divisions in the company (for convenience, the divisions are numbered from **1** to **n**), and **m** indicates which division is the Central Management division. This is followed by **k** lines, each containing two integers **1 i, j n**, indicating that division **i** and division **j** cannot cooperate (thus, **i** cannot be directly in charge of **j** and **j** cannot be directly in charge of **i**). You may assume that **i** and **j** are always different.

Output

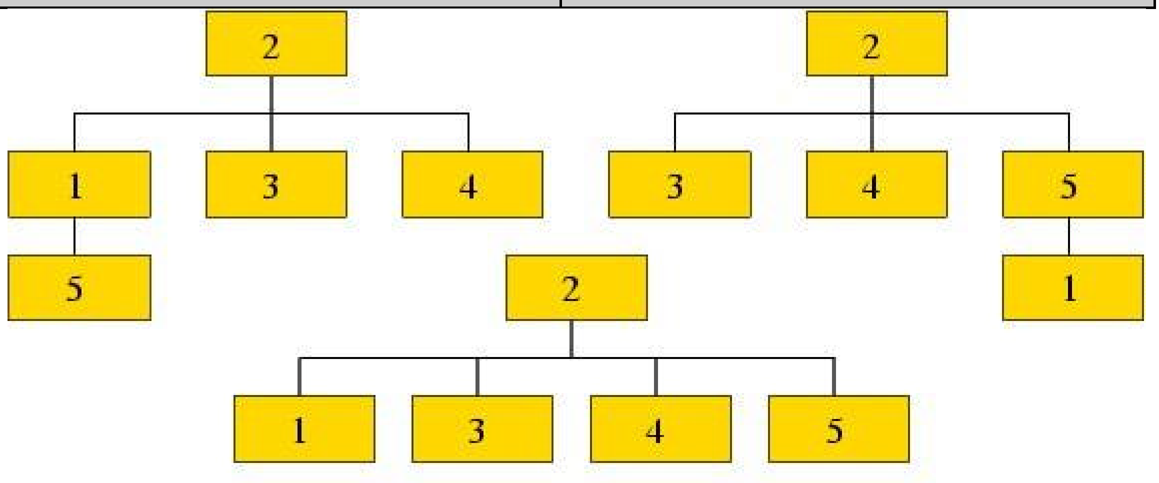
For each test case, print the number of possible ways to organise the company on a line by itself. This number will be at least **1** and at most $\mathbf{10^{15}}$.

Sample Input

```
5 5 2
3 1
3 4
4 5
1 4
5 3
4 1 1
1 4
3 0 2
```

Output for Sample Input

```
3
8
3
```



The three possible hierarchies in the first sample case

Problem setter: Per Austrin

Collector's Problem

Input: standard input
Output: standard output
Time Limit: 5 seconds

Some candy manufacturers put stickers into candy bar packages. Bob and his friends are collecting these stickers. They all want as many different stickers as possible, but when they buy a candy bar, they don't know which sticker is inside.
It happens that one person has duplicates of a certain sticker. Everyone trades duplicates for stickers he doesn't possess. Since all stickers have the same value, the exchange ratio is always 1:1.
But Bob is clever: he has realized that in some cases it is good for him to trade one of his duplicate stickers for a sticker he already possesses.
Now assume, Bob's friends will only exchange stickers with Bob, and they will give away only duplicate stickers in exchange with different stickers they don't possess.
Can you help Bob and tell him the maximum number of different stickers he can get by trading stickers with his friends?

Input

The first line of input contains the number of cases T ($T<=20$).
The first line of each case contains two integers n and m ($2<=n<=10, 5<=m<=25$). n is the number of people involved (including Bob), and m is the number of different stickers available.
The next n lines describe each person's stickers; the first of these lines describes Bob's stickers.
The i-th of these lines starts with a number $k_i<=50$ indicating how many stickers person i has. Then follows k_i numbers between 1 and m indicating which stickers person i possesses.

Output

For each case, print the test case number together with the maximum number of different stickers Bob can get.

Sample Input

2
2 5
6 1 1 1 1 1 1
3 1 2 2
3 5
4 1 2 1 1
3 2 2 2
5 1 3 4 4 3

Sample Output

Case #1: 1
Case #2: 3

Explanation of the sample output:

In the first case, no exchange is possible, therefore Bob can have only the sticker with number 1.
In the second case, Bob can exchange a sticker with number 1 against a sticker with number 2 of the second person,
and then this sticker against a sticker with number 3 or 4 of the third person, and now he has stickers 1, 2 and 3 or 1, 2 and 4.

Problem setter: Adrian Kuegel

H. Shopaholic

Lindsay is a shopaholic. Whenever there is a discount of the kind where you can buy three items and only pay for two, she goes completely mad and feels a need to buy all items in the store. You have given up on curing her for this disease, but try to limit its effect on her wallet.

You have realized that the stores coming with these offers are quite selective when it comes to which items you get for free; it is always the cheapest ones. As an example, when your friend comes to the counter with seven items, costing 400, 350, 300, 250, 200, 150, and 100 dollars, she will have to pay 1500 dollars. In this case she got a discount of 250 dollars. You realize that if she goes to the counter three times, she might get a bigger discount. E.g. if she goes with the items that costs 400, 300 and 250, she will get a discount of 250 the first round. The next round she brings the item that costs 150 giving no extra discount, but the third round she takes the last items that costs 350, 200 and 100 giving a discount of an additional 100 dollars, adding up to a total discount of 350.

Your job is to find the maximum discount Lindsay can get.

Input

The first line of input gives the number of test scenarios, $1 \leq t \leq 20$. Each scenario consists of two lines of input. The first gives the number of items Lindsay is buying, $1 \leq n \leq 20000$. The next line gives the prices of these items, $1 \leq p_i \leq 20000$.

Output

For each scenario, output one line giving the maximum discount Lindsay can get by selectively choosing which items she brings to the counter at the same time.

Sample Input

```
1
6
400 100 200 350 300 250
```

Sample Output

```
400
```

Problem D: Dying Tree

Once upon the time in the forest, there were lots of trees who were all friends to one another. One of the trees T was very sick. She needed a tree doctor to save her life. As you may already know, trees can't move, but what you probably didn't know is that they can talk. Each tree t1 can talk to tree t2 if the minimum distance between any two branches from each is less than or equal to some value **k**. All trees decided to help their sick friend by trying to reach a doctor tree. They will continue to tell one another that tree T is sick until some tree S finds a tree doctor (who is at distance **d** or less from any branch of tree S). S will tell the doctor about her friend so he can go help her.

- A tree is represented by a set of points representing her branches.
- A doctor is represented by a single point.

Input

Input begins with a number **t** < 100 representing the number of test cases; t test cases follow. Each test case begins with 4 integers 0 < **n** < 100, 0 < **m** ≤ 10, 0 ≤ **k, d** ≤ 100 where n is the number of trees in the forest, **m** is the number of doctors in the forest, **k** & **d** are as described above. The next m lines represent the positions of doctors in x, y coordinates. The following lines describe the set of trees in the forest. Each set begins with an integer 0 < **b** < 10 representing the number of branches this tree has. Followed by b points representing the branches positions. The sick tree is always the first tree in the input. All points coordinates are integers with absolute values less than or equal to 1000.

Output

For each test case determine whether or not the trees can help their friend by finding a doctor for her. If yes, then print "Tree can be saved :)", if no then print "Tree can't be saved :(".

Sample Input

```
2
3 2 2 3
3 8
7 4
4
0 0
1 1
3 2
2 0
3
6 -1
7 1
8 2
3
-1 -1
2 -3
5 -2
3 2 1 2
3 8
7 4
4
```

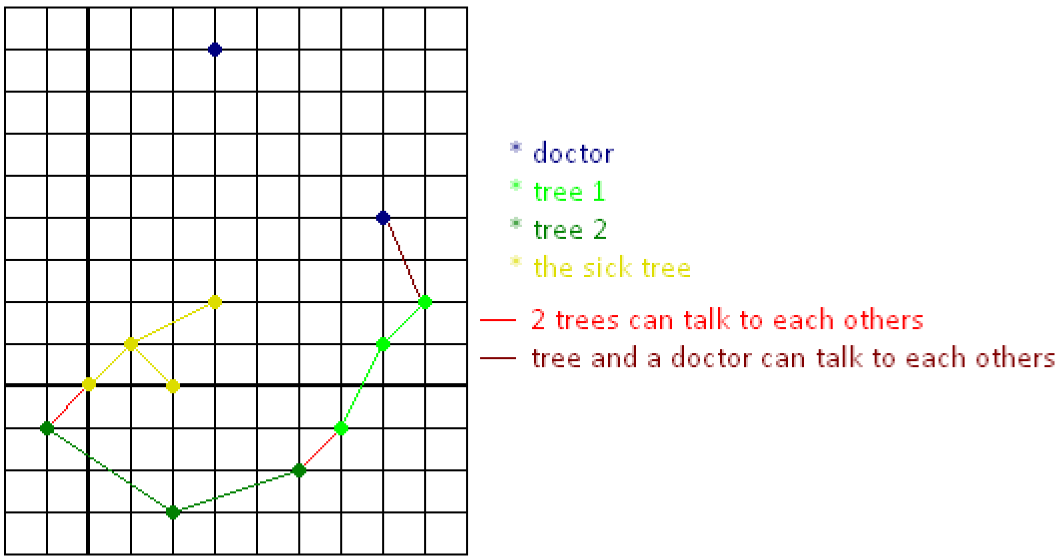
Sample Output

```
Tree can be saved :)
Tree can't be saved :(
```

```
0 0
1 1
3 2
2 0
3
6 -1
7 1
8 2
3
-1 -1
2 -3
5 -2
```

Problem Setter: Asmaa Magdi
Special Thanks: Muhammad Magdy
Alternate Solution: Sohel Hafiz

Illustration: The following diagram depicts Sample #1



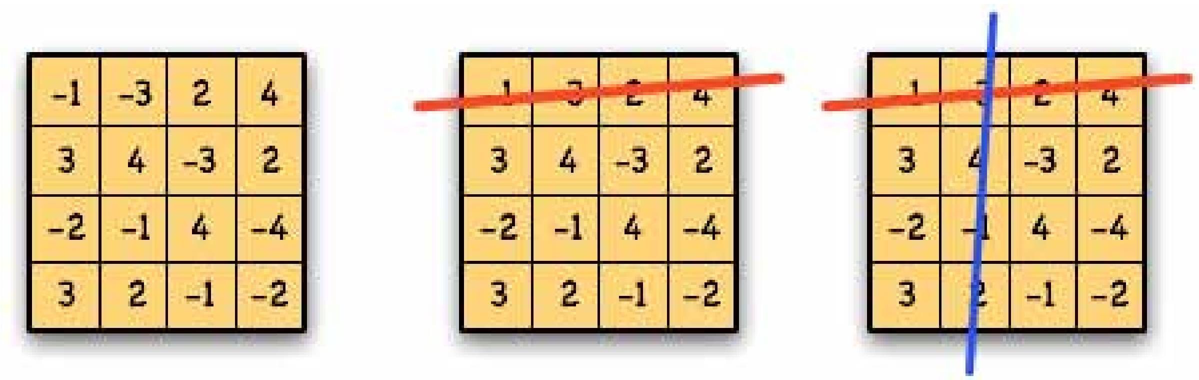
Problem G

GRID GAME

Alice and Bob both have lots of candies but want more. They decide to play the following turn-based game.

They fill an n x n grid M with random integers. Alice begins the game by crossing off an uncrossed row i of the grid. Now it's Bob turn and he crosses off an uncrossed column j of the grid. At the end of Bob's turn, Alice takes the number candies in the ith row and jth column of M, call this value $M(i, j)$, from Bob. (If $M(i, j)$ is negative, then Alice gives $|M(i, j)|$ candies to Bob.) The game continues alternating turns from Alice to Bob until the entire board is crossed off.

What is the largest amount of candies that Alice can win from Bob (or least amount to lose if she cannot win) if both Alice and Bob play optimally?



The beginning of a game between Alice (red) and Bob (blue).

Program Input

The first line of the input contains an integer t ($1 \le t \le 20$), the number of test cases. Each test case starts with n ($1 \le n \le 8$), the size of the grid. Then follow n lines containing n numbers separated by spaces describing M. We call the jth number on ith line $M(i, j)$ ($-1000 \le M(i, j) \le 1000$).

Program Output

For each test case, print the largest amount of candies that Alice can win from Bob. If she cannot win, print the negative number indicating the minimum number of candies she loses.

Sample Input & Output

INPUT

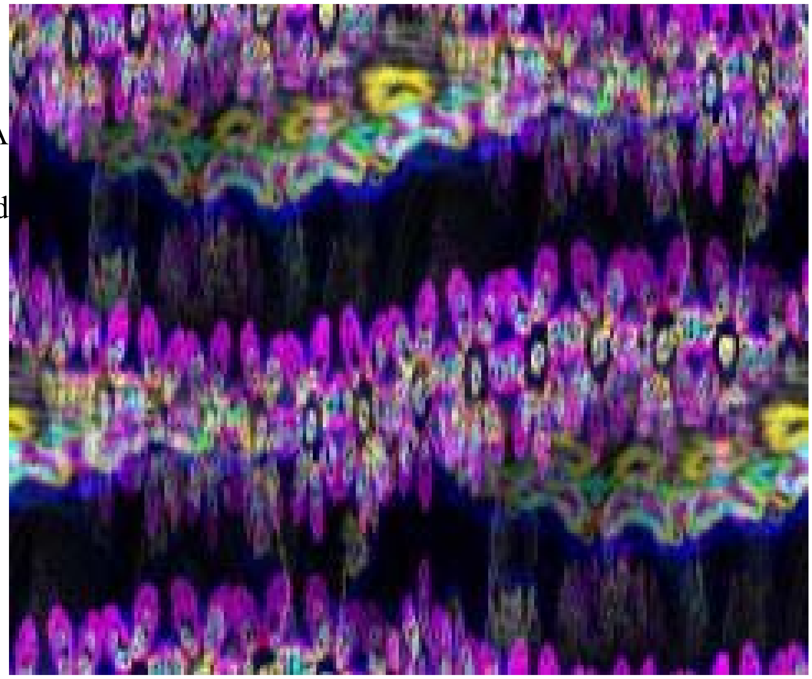
```
3
2
10 10
-5 -5
2
10 -5
10 -5
2
10 -5
-5 10
```

OUTPUT

```
5
5
-10
```

Calgary Collegiate Programming Contest 2008

Problem G: Alien DNA



Alien DNA is more complicated than human DNA. For example, rather than viewing an alien DNA strand as a sequence of base pairs, it can be viewed as a sequence of base sets. A base set is, simply put, a string of lowercase letters where each letter represents a different base. Note that human DNA is also a sequence of base sets, except a base set is simply a pairing of `a` with `t` or `c` with `g`.

Since alien DNA is more complicated than human DNA, the algorithms used in alien bioinformatics are also much more complicated. However, you have only been working at AlienBioTechNextGenCorp for three days and are not yet trusted enough to handle the complicated tasks.

Your first job is simple. You must develop software to cut a strand of alien DNA into as few segments as possible. The only constraint is that base sets of a given segment must share at least one common base.

Input begins with an integer $t \leq 100$, the number of test cases to be processed. Each test case begins with a single value n ($1 \leq n \leq 10{,}000$) representing the length of the alien DNA strand. Each of the n following lines consists of a single string of lowercase letters representing a base set. Each string will contain at least one character, and no repeated characters. The base sets are given in the input in the same order they appear on the DNA strand.

For each test case, output the minimum number of cuts required to partition the strand into segments that share a common base.

Sample input

```
2
5
as
sd
df
fg
```

```
gh
3
plum
orange
plum
```

Sample output

```
2
2
```

Babak Behsaz and Zachary Friggstad

A. Game of Blocks

John decided to buy his son Johnny some mathematical toys. One of his most favorite toy is blocks of different colors. John has decided to buy blocks of C different colors. For each color he will buy googol (10^{100}) blocks. All blocks of same color are of same length. But blocks of different color may vary in length.

Jhonny has decided to use these blocks to make a large 1 x n block. He wonders how many ways he can do this. Two ways are considered different if there is a position where the color differs. The example shows a red block of size 5, blue block of size 3 and green block of size 3. It shows there are 12 ways of making a large block of length 11.

Input

Input starts with a positive integer T ≤ 25. T test cases follow.

Each test case starts with an integer 1 ≤ C ≤ 100. Next line consists c integers. ith integer 1 ≤ len_i ≤ 750 denotes length of i^{th} color. Next line is positive integer n ≤ 10^{15}.

Output

For each case output case number followed by the number of ways Johnny can make the desired block modulo 100000007 (a prime number). See sample output for exact format.

Sample Input	Sample Output
4 3 3 3 5 11 3 3 5 3 1111111111111 4 1 1 100 100 1000000 3 1 1 1 5	Case 1: 12 Case 2: 20634244 Case 3: 94126777 Case 4: 243

Problemsetter: Tanaeem M Moosa, Special Thanks: Md. Arifuzzaman Arif

4502 - How Many bases?

Asia - Dhaka - 2009/2010

A classical problem of number theory is Find the number of trailing zeroes in N^M, in base B . This problem is quite conventional and easy. But a number can have same number of trailing zeroes in more than one base. For example, if decimal number 24 is written in 3, 4, 6, 8, 12 and 24 based number system, it will look like 80, 60, 40, 30, 20 and 10 respectively. So in all number systems it has only one trailing zero. Given a number N^M, your job is to find out the number of integer bases in which it has exactly T trailing zeroes.

Input

The input file contains at most 10000 line of input. Each line contains three integers N ($1 \le N \le 10^8$), M ($0 < M \le 10^7$) and T ($0 < T \le 10^4$). The meaning of N, M and T is given in the problem statement. Input is terminated by a line containing three zeroes, which obviously should not be processed for calculation.

Output

For each line of input produce one line of output. This line contains the serial of output followed by an integer NB, which is modulo 100000007 value of number of bases in which N^M has exactly T trailing zeroes.

Sample Input

24 1 1

100 200 10

23 18 2

0 0 0

Output for Sample Input

Case 1: 6

Case 2: 312

Case 3: 3

Dhaka 2009-2010

Problemsetter: Shahriar Manzoor
Special Thanks: Derek Kisman