DUET Inter University Programming Contest 2019
DUET CSE Fest 2019

Hosted by DUET
Dhaka, Bangladesh

27th April 2019

You get 17 Pages, 9 Problems & 300 Minutes

Platform Support:

CodeMarshal

Problem Set By:

Bangladesh Association of Problem Setters

Problem A

Input: Standard Input
Output: Standard Output

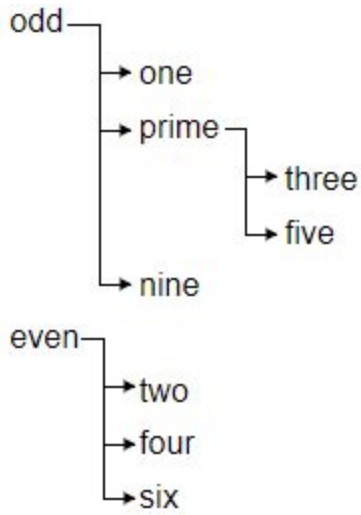
All About Respect

In this problem you will be given a block as input. You need to sort it lexicographically maintaining the block structure and print in sorted order.

A block has the following criteria:

- Block is non-empty.
- A block contains:
  - Elements separated by newlines.
  - Each Element contains:
    - A non-empty string of lower case letter only.
    - Zero or more 'sub-block's. A 'sub-block' will be indented using a **single whitespace character** from the particular element. Each 'sub-block' will have the same properties of a block.

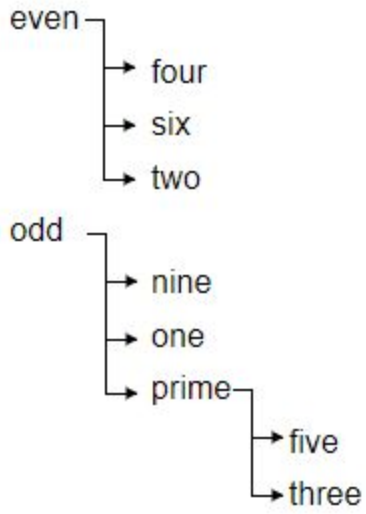
A block of size **N** will have **N** 'new line' separated strings in total. For example, consider the following block of size 10:



(For clear view, the indentations are demonstrated in the above way. IO files will have a single space character for a single indentation, check samples)

The element “odd” contains a single sub-block which contains three elements “one”, “prime” and “nine”. The element “prime” has a sub-block which contains two elements “three” and “five”.
The element “even” contains a single sub-block of three elements “two”, “four” and “six”.

If you simply sort it lexicographically, the logical block structure will be lost. You need to sort the block lexicographically, but respecting the indentations to preserve the block structure. The desired output of the above example:



(For clear view, the indentations are demonstrated in the above way. IO files will have a single space character for a single indentation, check samples)

Input

The first line will contain a single integer **T (1 ≤ T ≤ 100)**. Each test case will have a single integer **N (1 ≤ N ≤ 100)**, denoting the size of the block. The following N lines will contain strings constituting the block. For each test case, all strings are unique.

Output

Print the case number in a single line followed by the block in desired format. Please see sample for details.

Sample Input

```
1
10
odd
 one
 prime
  three
  five
 nine
even
 two
 four
 six
```

Output for Sample Input

```
Case 1:
even
 four
 six
 two
odd
 nine
 one
 prime
  five
  three
```

Problem B

Input: Standard Input
Output: Standard Output

Jumping Gollum

There is an array **A** of **N** integers indexed from **1** to **N**. Initially Gollum is sitting at the **index 1** and the precious ring is at the **index N**. Obviously, Gollum is trying to get the ring as soon as possible.
Let's say, Gollum is at the index **i (1 ≤ i < N)**. Then, in one second Gollum can take any of the following two moves:

- Move 1: Jump to the index **i + 1**
- Move 2: Jump to the index **j** where **i < j ≤ N, gcd(A[i], A[j]) > 1.**

But we know that Gollum is not superman. So there are some limitations on its jumping capability. In one jump Gollum can not move more than **K** index forward, that means if Gollum is at the index **i**, it can reach the index **j** if the following condition holds: **i < j ≤ i + K.**

You have to find the minimum time (in seconds) needed to reach the **index N,** so that Gollum can take the precious ring.

Input

The first line of the input is an integer **T**, denoting the number of test cases. Next, there will be **2T** lines describing the **T** test cases. There are exactly two lines for describing a test case. The first of them contains two space-separated integers **N and K.** The second line contains **N** space separated Integers which represent the array **A**.

Output

For each test case, print a line in the format, **"Case X: Y"**, where **X** is the case number and **Y** is the minimum time needed.

Constraint

- **1 ≤ T ≤ 4000**
- **1 ≤ N ≤ 2x10^5**
- **1 ≤ K ≤ N**
- **1 ≤ A[i] ≤ 10^6**
- **Sum of N over all test cases ≤ 2 x 10^6**

Sample Input	Output for Sample Input
3 5 4 10 13 6 7 9 5 5 10 9 8 7 12 5 3 7 5 3 11 2	Case 1: 2 Case 2: 1 Cass 3: 4

N.B.: Input file is huge. Please use faster I/O.

Problem C

Input: Standard Input
Output: Standard Output

Count the Attendance

In the world of competitive programming machine learning is rather an unknown and unnecessary topic. But in the professional world all the craze is about machine learning. Although the recent surge of cryptocurrency and blockchain based "Super Models" have everyone's attention, no one really understands them and some are very skeptic about the feasibility of these models given the available hardware.

In this problem you are going to solve a problem that will also give you an insight of the problems machine learning models solve. In particular there is a special group of problems called "unsupervised learning" where a machine learns to partition given data into groups.

For this problem you are given a rather interesting case of finding groups. You are going to try and find the different friend circles in a university! I know this last sentence is making your gossip heart cry out with excitement! Just think of all the possibilities if you could find out who is friend with who, who is dating who etc.

In universities people find freedom and with that freedom lies a specific behavior that you are going to exploit. In particular, students usually come to universities not to attend classes but to meet up with friends. There are some lone wolves who take class notes and who come to save everyone like Noah during exams with their metaphorical arc - class notes. But apart from these unique creatures there are several groups in a class who will usually be present either all together or none at all. Well maybe that's too extreme. Because this is not always the case. But it's usually the case most of the time.

You will be given an attendance sheet of **N** students of a particular class for **M** days. In order to find their various group structure and intricate social connections you will first need to find which students come to the class as a group. A group does not necessarily consist of all students present in a day and each possible subset of students can be a group. So you would like find out for each student the most likely group that they belong to. Each group has a specific weight to it. It is calculated by this function:

```
int weights[1 << N], seed;    // seed will be given as input, N is the number of students
void calc() {
      weights[0] = 1;
      for(int i = 1; i < (1 << N); i++)   weights[i] = (weights[i-1] * seed) % 10000;
}
```

The likelihood of a student belonging to a group is calculated by:

*Number of days every member of that group was present * Weight of that group*

For each student, you have to find all the groups with highest likelihood that the student is part of. For example, if a group is formed of students 2 and 3, then student 1 is not part of it, and it would be meaningless to calculate its likelihood for student 1. If a group is of the highest likelihood for one of its members, it may not be so for all other members as well. For example, person 3 may belong to a group consisting of person 2 and 3 whereas person 2 may belong to another group consisting of person 1 and 2.

For each day, you will be provided a **decimal mask**. A **decimal mask** is a decimal integer, whose binary representation will denote the attendance for a day. If the i-th bit (1-indexed) from the right is on, then we can say that the i-th student is present. For example, if students 3 and 4 are present in a day, then the binary representation has to be 1100. So the given **decimal mask** will be 12. Similarly, if students 1, 2 and 4 are present, the given **decimal mask** will be 11.

Input

The first line contains a number **T (1 ≤ T ≤ 10)** which represents the number of test cases. The first line of each test case consists of **3** integers, **N (3 ≤ N ≤ 20), M (0 < M ≤ 100000)** and **seed (0 < seed < 10000)** denoting the number of students in the class, the number of days of attendance you are given and the seed needed to calculate the group weights. Each of the next **M** lines contains an integer denoting the **decimal mask** (whose binary representation denotes the attendance of students of that day, mentioned earlier).

Output

For each test case print the case number in first line. After that, print one line for each student containing his/her maximum likelihood belonging to any group, followed by the **decimal mask** of that group. If multiple such group exists, you have to print **decimal masks** of all those groups sorted in ascending order separated by single space. Check sample for details.

Sample Input

```
2
5 6 2
4
11
10
17
19
23
3 3 2
5
6
5
```

Output for Sample Input

```
Case 1:
8608 23
8608 23
8608 23
2048 10 11
16608 16
Case 2:
64 5
64 6
64 5 6
```

Explanation of Sample I/O

In case 1, for students 1, 2, and 3, maximum likelihood is 8608, which can be obtained from group {1, 2, 3, 5}. For student 4, maximum likelihood is 2048, which can be obtained from groups {2, 4} and {1, 2, 4}. For student 5, maximum likelihood is 16608, which can be obtained from group {5}.

Problem D

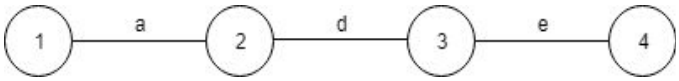
Input: Standard Input
Output: Standard Output

Save the World

Mr. MarZam lives in the techland. He works as a CID. He has a great reputation for his work. There is not a single investigation that he wasn't able to solve. There are **N** beautiful cities in the techland and they all are connected with **N-1** bidirectional roads.

A few days ago, a new gang of criminals has raised who are very jealous of seeing Techland's people in peace. So, they are planning to attack in some of the cities.

As Mr. MarZam has spies all over the country, he got the news that something wrong is going to happen. In order to protect people from these attacks, he wants one of his best spies, Mr. ShaVel, to infiltrate the gang and learn about their upcoming attack plans. But it comes out that it's not easy for Mr. ShaVel to communicate with Mr. MarZam when he is with the gang. So they made an alternate plan to communicate with each other. They named the cities with integers from **1** to **N** and labeled each of the roads with **lowercase** letters (**a** to **z**). Multiple roads can be labeled with the same letters. Mr. ShaVel will send a city name **U** and a sequence of letters **S** which means - there will be an attack on the city named **V** (**U** $\neq$ **V**) if the shortest path from **U** to **V** has the same letter sequence as **S**.



The sequence of letters from city 1 to city 3: *ad*
The sequence of letters from city 1 to city 4: *ade*

According to the plan, Mr. ShaVel joined the gang. After Mr. MarZam received the first message from his spy, he fell into a very critical situation. He figured out that there can be multiple cities which can have the same sequence of letters from city **U**. So, he needs to take protection in all those cities. But figuring out those cities may take too much time and delay the process to take preventive measures. As you guys are a great team and you are always ready to help people, Mr. MarZam has come to you for helping him with a piece of code to find out the cities. But to make your life a bit simple, Mr. MarZam only asked you to give him the total number of cities which may have an attack.

Input

The first line of the input will contain an integer **T (1 ≤ T ≤ 5)** which denotes the number of test cases.

Each test case starts with a line having two integers **N (1 ≤ N ≤ 10^5)** and **Q (1 ≤ Q ≤ 10^5)**. Each of the next **N - 1** lines will contain two integer $\mathbf{U_i}$**(1 ≤** $\mathbf{U_i}$ **≤ N),** $\mathbf{V_i}$ **(1 ≤** $\mathbf{V_i}$ **≤ N)** and a lowercase letter $\mathbf{C_i}$ which means there is a

bidirectional road between city $\mathbf{U_i}$ and $\mathbf{V_i}$ and the road is labeled with $\mathbf{C_i}$. Next, **Q** lines follow. Each of the lines will contain an integer $\mathbf{U_i}$ **(1 ≤ $\mathbf{U_i}$ ≤ N)** which is a city name and a non-empty sequence of lowercase letters $\mathbf{S_i}$ provided by Mr. ShaVel.

In a single input file, the sum of lengths for all the letter sequences will be less than or equal to $\mathbf{2 \times 10^6}$.

Output

For each test case, you should print the case number on the first line in the format: "**Case X:**" where **X** is the case number. Each of the next **Q** lines should contain the answer to Mr. ShaVel's messages. Please see the sample I/O for more clarification.

Sample Input

```
1
6 3
1 2 a
1 3 a
2 4 b
3 5 b
2 6 c
1 ab
2 b
1 ac
```

Output for Sample Input

```
Case 1:
2
1
1
```

N.B.: Input file is huge. Please use faster I/O.

Problem E

Input: Standard Input
Output: Standard Output

Alice & Her Cousins

Alice has a positive number of cousins. One day she wanted to distribute some chocolates among her cousins. All of her cousins had to get an equal number of chocolates because she didn't want them to know that she didn't love them equally. In order to fulfill her purpose, she went to a local grocery store and bought a box having **N** chocolates and **equally** distributed all the chocolates inside that box among all her cousins.

You are sure that if she had **Y** or **Z** number of cousins, then it wouldn't have been possible to distribute **N** chocolates equally among her cousins. Knowing this information, you have to tell if it is impossible for her to have **X** number of cousins.

Input

The first line of the input will contain an integer **T**, denoting the number of test cases. Then **T** lines will follow. Each line will have three **distinct** integers **X, Y, Z** separated by a space.

Constraints

- **$1 \leq T \leq 100$**
- **$2 \leq X, Y, Z \leq 10^9$**

Output

For each test case, print a line containing the case number and if it is impossible for her to have **X** number of cousins, then print "**Impossible**", otherwise print "**Possible**". Look at the sample input/output for more details.

Sample Input	Output for Sample Input
2 5 3 7 16 4 8	Case 1: Possible Case 2: Impossible

Problem F

Input: Standard Input
Output: Standard Output

Building Blocks

Jacob Builders is the largest builder in the city of Rozenthal. The city has become a hub of manufacturing, and this has created a huge problem for the residents. To control the chaos the city has limited the different types of blocks that can be manufactured at any time. Blocks are used to build pillars by stacking them one above another. Now Jacob Builders want to build a new skyscraper and need to know if they can construct some pillars with the new restriction. In order to find that out, they've come to you for help.

You are given an array of **N** integers, where each integer is the length of a pillar. Next, you are given the lengths of **K** blocks and a list of **Q** queries of the form **[L, R]**. For each query, you have to answer how many pillars of lengths between the **L** and **R** index of the array can be made using the **K** blocks. You can use any block any number of times.

For this problem you will given two integers A and B, and the lengths of the pillars can be generated using the formula:

$$Pillar[i] = (A \times i) + B \text{, where } 1 \leq i \leq N$$

Input

Input starts with a line containing the number of test cases, **T**. For each test case the first line will contain an integer, **N**, followed by a line containing two integers, **A** and **B**. After that you will be given a line with the integer **K** and **K** space separated integers on the next line. The next line will contain an integer **Q**. Each of the next **Q** lines will contain two integers **L** and **R**.

Output

For each test case, output the case number followed by results of the queries for that test case, each on a seperate line. See the sample output for more details.

Constraints:

- **$1 \leq T \leq 10$**
- **$1 \leq N \leq 10^{15}$**
- **$0 \leq A, B \leq 10^{15}$**
- **$1 \leq K, Q \leq 50$**
- **$1 \leq L, R \leq N$**
- **1 ≤ The length of any pillar ≤ 10^{15}**
- **1 ≤ The length of any block ≤ 10^{5}**

Sample Input

```
2
100
1 0
2
2 3
3
1 10
10 50
50 100
100
3 0
2
5 7
3
2 8
7 50
35 80
```

Output for Sample Input

```
Case 1:
9
41
51
Case 2:
4
44
46
```

Problem G

Input: Standard Input
Output: Standard Output

Interesting Device

The scientists of Gigaland have developed some interesting devices. Those devices can solve very complex calculations by communicating with each other. The devices are placed in various points of 3D space. The communication between any pair of devices is always performed in a straight line connecting the coordinates of both devices. Communications don't face any difficulty if other devices lie on the straight line connecting a pair of devices.

The government of Gigaland has decided to use those devices for weather forecasting. To do that with best performance, scientists have calculated optimal coordinates for each device. As the devices are very sensitive, all of those are covered by a special veil which separates the devices from the remaining universe(!), so that the devices can communicate with each other without any external barrier. They covered the devices in such a way that, every line of communication remains within the volume enclosed by the veil.The material of the veil is very rare and costly, that's why the covering was done with veil of minimum surface area. It is confirmed that the volume of interior space enclosed by the veil is always greater than zero.

After completing the setup they have faced an interesting issue: an invisible infinite plane which has passed through the veil hampering the communication between some devices. They have decided to destroy the portion of invisible plane which is inside the veil. To do that they need to calculate the area of that particular part. Help them to calculate the area. It is confirmed that this area is nonzero

Input

Input starts with an integer **T** (**T ≤ 25**) denoting the number of test cases followed by a blank line.

First three lines contains three integers giving the **X, Y** and **Z** coordinates of three noncollinear distinct points defining the infinite plane. Next line contains an integer **N** (**4 ≤ N ≤ 200**), the number of devices. Each of the following **N** lines contains three integers giving the **X, Y** and **Z** coordinates of a device. Absolute value of all the coordinates of input set is not greater than **100.**

Consecutive test cases are separated by a blank line.

Output

For each test case, print a line in the format, **"Case X: Y"**, where **X** is the case number and **Y** is the area of the portion of infinite plane which is inside the veil. Absolute error less than 10^{-3} will be ignored.

Sample Input

```
2

0 0 5
5 5 5
0 5 5
8
0 0 0
10 0 0
0 10 0
10 10 0
0 0 10
10 0 10
0 10 10
10 10 10

0 0 0
0 5 0
10 8 10
8
0 0 0
10 0 0
0 10 0
10 10 0
0 0 10
10 0 10
0 10 10
10 10 10
```

Output for Sample Input

```
Case 1: 100.00000000
Case 2: 141.421356237
```

Problem H

Input: Standard Input
Output: Standard Output

Skill Matching

BAPSIT is one of the most popular companies in the tech industry right now. They have been producing the most anticipated softwares in the market for years. But due to some budget allocation related issues, they are going to fire some of their employees as soon as possible.

There are a total of **N** employees currently working at BAPSIT. Each of those individuals has got a particular skill where they are very specialized at, as they have been working here for a long time. The skill set can be represented as integers in the range from **1** to **M**. Some employees have already made friends with each other and this friendship network can be represented as a tree. **It is to be noted that, for any pair of employees u and v, if u is a friend of v, then v is also a friend of u in this friendship network.**

Now the company wants to fire the maximum number of employees but at the same time, they also want to maintain all of their projects very smoothly. So they want at least one employee from each particular skill to stay at the company. Besides, they also want to make sure that, those who will eventually stay in the company, must be able to communicate with each other. For any pair of employees u and v, they can communicate with each other if they are directly friends with each other or they are connected with each other through some other employees in the remaining friendship network.

For example, let's say that employee 1 and 2 are friends, employee 2 and 3 are friends and at the same time employee 3 and 4 are also friends. In this example, all the employees will be able to communicate with each other. But if we fire the employee 3 from the company then the employee 1 and 4 won't be able to communicate with each other anymore.

Now as a senior project manager of BAPSIT, you have to find the maximum number of employees BAPSIT can fire.

Input

The first line of the input contains a single integer **T**, which denotes the number of test cases. The first line of each test case contains two integers **N** and **M**. The next line of a test case contains an array **S** of **N** space separated integers where $\mathbf{S_i}$ represents the skill of the i-th employee. Each of the next **(N - 1)** lines contains two integers **u** and **v** which denotes that the employees **u** and **v** are friends with each other.

Note: It is guaranteed that, in the initial friendship network provided in the input, all the employees are able to communicate with each other and there exists at least one employee from each particular skill.

Constraints

- **$1 \leq T \leq 10$**
- **$1 \leq N \leq 1000$**
- **$1 \leq M \leq 10$**
- **$1 \leq S_i \leq M$**
- **$1 \leq u, v \leq N$**

Output

For each test case, output a single line in the format "**Case X: D**" without the quotes. Here, **X** is the case number and **D** is the maximum number of employees the company can fire.

Sample Input

```
1
10 4
1 2 4 2 1 4 1 2 4 3
1 2
1 3
1 4
3 5
3 6
3 7
6 8
6 9
9 10
```

Output for Sample Input

```
Case 1: 4
```

Explanation

The company can fire the employees **2, 4, 5** and **7**. The remaining employees cover all the required skills and they can also communicate with each other either directly or through some of their friends.

Problem I

Input: Standard Input
Output: Standard Output

Almost Pattern Matching

Two strings **S1** and **S2** are said to be *almost equal* if they are of the same length and either of the following is true:

1. The strings **S1** and **S2** are equal, i.e, they are exactly the same.
2. **S1** and **S2** has exactly one mismatch between them, i.e, there is exactly one index *i* such that **S1[i] ≠ S2[i]**

Some examples of *almost equal* strings:

1. *hello* and *hello*
2. *good* and *food*
3. *aaabbb* and *aaaabb*

However, the following below are **not** *almost equal* strings:

1. *hello* and *helloo*
2. *feel and good*
3. *abcde and abdce*

Given a text **S**, and **Q** queries where each query contains a pattern **P**, count for each query how many substrings are there in the text which are *almost equal* to the pattern **P**.

Input

The first line of the input contains an integer **T**, denoting the number of test cases. Then the description of **T** test cases follow. The first line of every test case will contain the text **S**. The next line contains an integer **Q**. Each of the next **Q** lines will contain a pattern **P**.

Constraints

- **1 ≤ T ≤ 8**
- **1 ≤ |S|, |P| ≤ 262144**
- **1 ≤ Q ≤ 262144**

Sum of characters in the pattern across all queries in a single test case will not exceed **262144**.

|S| and **|P|** will be a **power of two** always. This means that **$|S| = 2^Y$**, where **Y** is any non-negative integer satisfying the constraints above. Similarly, **$|P| = 2^Z$**, where **Z** is any non-negative integer satisfying the constraints.

S and **P** will only contain characters from the **first 16** characters of the lowercase English alphabet **(a-p)**.

Output

For each test case, first print the case number in one line, like *"**Case X:**"*, where ***X*** is the case number starting from **1**. Then for each query, output the number of substrings in the text which are *almost equal* to the pattern **P**.

Sample Input

```
2
abcdcabcdacbfefe
8
abcd
dabc
dc
p
ff
acbd
gg
efef
aaaababa
4
bbbb
aaaa
abcdcabcdacbfefe
aa
```

Output for Sample Input

```
Case 1:
2
1
5
16
4
1
0
1
Case 2:
0
3
0
7
```

N.B.: Input is huge, please use faster I/O methods.