Problem A

Input: Standard Input
Output: Standard Output

Expected Coin Change (II)

TECHSURGENCE

Mr. Kiptus has a bag of **N** coins. These are **N** different kinds of coins with value from **1** to **N** Taka; one coin of each type (he is rich). He wants to donate some coins. He has decided not to donate more than **M** Taka (he is also kiptus). But he is also interested to keep things dynamic. So he devises the following procedure:

1. He will first start with 0 donations.
2. He will take a coin randomly from the bag. The probability of getting any coin left in the bag is same. If there is no coin left, he will stop donating.
3. He will then check if on donating this coin, his total donation exceeds M or not. If not, then he will donate the coin, and his donation will increase by the coin's value and start again from step 2. Otherwise, he will return the coin to the bag and stop donating.

Now he is wondering what is the expected amount he will donate. Can you calculate it?

Input

The first line of the input is **T ($T \le 40$)**, then T test cases follow. In each case, there will be two integers **N ($1 \le N \le 30$)** and **M ($0 \le M \le 10^7$)**.

Output

For each test case, print a line of the format "**Case I: A/B**", where **I** is the case number and **A/B** is the expected money Mr. Kiptus will donate in reduced fraction form (A and B are coprime). It is guaranteed that **A** and **B** will be less than **10^{16}**.

Sample Input	Output for Sample Input
3 3 5 5 3 10 10	Case 1: 4/1 Case 2: 27/20 Case 3: 3127/420

Problem B

Input: Standard Input
Output: Standard Output

Deja Vu

In this problem, all you have to do is generate an array having **N** positive integers which satisfies **Q** constraints.

Every constraint is of the following kind - "The least common multiple of all the numbers from the **L**-th position to the **R**-th position of the array has to be a multiple of **X**".

You need to find the lexicographically smallest array which satisfies all the constraints.

Input

The first line of the input contains an integer **T**, denoting the number of test cases. Then the description of the T test cases follow. The first line of every test case will contain two integers **N** and **Q**, denoting the length of the array and the number of constraints respectively. Each of the following **Q** lines will contain a constraint of the following form: **L R X**.

Constraints

- $1 \le T \le 100$
- $1 \le N \le 10^5$
- $1 \le Q \le 10^5$
- $1 \le L \le R \le N$
- $1 \le X \le 10^6$

Both, sum of N and sum of Q over all test cases $\le 4 \times 10^5$

Output

For each test case, print a line containing the case number and the lexicographically smallest array which satisfies all the given constraints. As the numbers of the array can be very large, output the numbers modulo 1000000007 ($10^9 + 7$).

Sample Input	Output for Sample Input
1 7 2 2 5 2 5 7 3	Case 1: 1 1 1 1 2 1 3

An array **C** of size **N** is called lexicographically smaller than another array **D** of equal size, if and only if there exists an **i ≤ N** such that:

C[p] == D[p] for all **1 ≤ p < i** and **C[i] < D[i]**

Warning: Dataset is huge. Please use faster I/O methods.

Problem C

Input: Standard Input
Output: Standard Output

Theory of Compression

Big data is a buzzword, isn't it? According to Wikipedia *"Big data refers to data sets that are too large or complex for traditional data-processing"*. A gigantic amount of data is generated from different sources like news portals, social media, blogs, communication, photos, services and so on. On average 2.5 quintillion bytes of data are created each day. Large data servers are used to store these data and it is ever growing. So, we need a more efficient compression technique for storing data.

Mr. Maddy, a mad scientist came up with a new algorithm for text data compression named "theory of compression". The theory is as follows:

- **T** is the original text and **C** is the compressed text, where $C = P_1R_1P_2R_2P_3R_3...P_nR_n$
- The tuple $\{P_i, R_i\}$ means the pattern P_i is repeated R_i times in the original text. That is P_1 is repeated R_1 times, P_2 is repeated R_2 times ... P_n is repeated R_n times consecutively.
- $P_1, P_2, P_3, \dots, P_n$ are composed of lower case Latin letters (a-z) only and $1 \le |P_1|, |P_2|, \dots, |P_n| \le 10^5$
- $1 \le R_1, R_2, R_3, \dots, R_n \le 50000$ and they do not contain any leading zeros.

For example, let T = **ababcccddde**. Here, C = **ab2c3d3e1** is the compressed text where, P_1 = ab, R_1 = 2, P_2 = c, R_2 = 3, P_3 = d, R_3 = 3, P_4 = e, R_4 = 1. If we decompress **C** we will get the text **T**.

In this problem, you will be given **C** and two integers **l** and **r** ($1 \le l, r \le 10^{12}$ and $0 \le r - l < 10^4$). You have to decompress **C** into **T** and print $T_lT_{l+1}T_{l+2}... T_r$. That is, you have to print the sequence **T[l...r]** from the indices **l** to **r** inclusive (1 based index).

For the above example C = **ab2c3d3e1**. So, T = **ababcccddde**.

- If, l = 2, r = 5, the answer will be **babc**.
- If, l = 9, r = 9, the answer will be **d**.
- If, l = 4, r = 15, the answer is **-1** as there is no such valid sequence.

Input

Input starts with an integer **TC** ($1 \le TC \le 10$) denoting the number of test cases. Each of the test cases starts with a non-empty string **C** ($1 \le |C| \le 2 \times 10^5$) denoting the compressed text. Next line of each test case will contain an integer **Q** ($1 \le Q \le 200$) denoting the number of queries. Next, there will be **Q** lines each containing two integers **l** and **r** ($1 \le l, r \le 10^{12}$ and $0 \le r - l < 10^4$). It is guaranteed that inputs are valid and $\sum (r - l + 1)$ in all test cases will be less than 10^7.

Output

For each test case, first, print the test case number in a single line and print the sequence **T[l...r]** of every query in a separate line. If there is no such valid sequence, print **-1** instead. Check the sample I/O section for more clarity.

Sample Input	Output for Sample Input
2 ab2c3d3e1 3 2 5 9 9 4 15 a10 2 1 10 1 11	Case 1: babc d -1 Case 2: aaaaaaaaaa -1

Warning: Dataset is huge. Please use faster I/O methods.

Problem D

Input: Standard Input
Output: Standard Output

Yatzy

Yatzy is a very popular game. Since the actual rules of the game and rules for this problem are not exactly the same, you are requested to go through all the rules very carefully.

Here are the rules of the game:

- You have **5** regular dice. Each of the dice has **6** faces and these faces contain **1** to **6** exactly once and each outcome is equiprobable. Outcomes of the dice are independent from each other.
- The game is played in **12** rounds.
- In each round, there are **3** stages.
- In the first stage, you will roll all the **5** dice.
- In the second stage, you can lock some of your dice(possibly none and all). If you lock some dice, you can not roll this in the current stage and the outcome of that dice remains same as the previous stage. After locking some dice, you will roll the dice that are not locked.
- In the third stage, you will do the same as the second stage.
- Now you have the final combination of the dice. The order of the dice does not matter. Here is an example of **3** stages: In the the first stage, you get **12135**. You lock the first and third (from left) dice and roll again. Then you get **14146**. Now you lock the second and fourth dice and roll again. Then you get **34246**. So, this is your final combination. You can treat this combination as **23446**, **64432**, **44632** etc too.
- There are **12** scoring zones. When you get your final combination, you have to choose a scoring zone. You will get some points according to your dice combination and chosen scoring zone.
- In each round you can choose only one scoring zone. And you can not choose a scoring zone more than once. So, basically after **12** rounds you choose each of those scoring zones exactly once.

Here is the point system for each scoring zone:

- **Scoring zone 1:** You get the sum of the outcome of those dice where the outcome is **1**. Example: If the combination is **11234**, you get **2** points. You will consider only those dice who have the outcome **1**. Then you get the sum of the outcome of those dice. So, you get **1 + 1 = 2** points.
- **Scoring zone 2:** You get the sum of the outcome of those dice where the outcome is **2**. Example: If the combination is **12224**, you get **6** points. You will consider only those dice who have the outcome **2**. Then you get the sum of the outcome of those dice. So, you get **2 + 2 + 2 = 6** points.
- **Scoring zone 3:** You get the sum of the outcome of those dice where the outcome is **3**.
- **Scoring zone 4:** You get the sum of the outcome of those dice where the outcome is **4**.
- **Scoring zone 5:** You get the sum of the outcome of those dice where the outcome is **5**.
- **Scoring zone 6:** You get the sum of the outcome of those dice where the outcome is **6**.
- **Scoring zone 7:** It is called **3 of a kind**. If you have at least **3** dice with the same outcome, then you get the sum of all the **5** outcomes. Otherwise you get **0** point. Example: If the combination is **33346**, you get **19** points.
- **Scoring zone 8:** It is called **4 of a kind**. If you have at least **4** dice with the same outcome, then you get the sum of all the **5** outcomes. Otherwise you get **0** point. Example: If the combination is **33336**, you get **18** points.

- **Scoring zone 9:** It is called **full house**. If you have exactly **3** dice with the same outcome and **2** dice with some other same outcome, then you get **25** points. Otherwise you get **0** point. Example: If the combination is **22255**, you get **25** points. But if the combination is **22222**, you get **0** point.
- **Scoring zone 10:** It is called **small straight**. If you get at least **4** outcome that forms an arithmetic series with difference **1**, then you will get **30** points. Otherwise you get **0** point. Example: If the combination is **34561**, you get **30** points.
- **Scoring zone 11:** It is called **large straight**. If you get all **5** outcome that forms an arithmetic series with difference **1**, then you will get **40** points. Otherwise you get **0** point. Example: If the combination is **12345** or **23456**, you get **40** points.
- **Scoring zone 12:** It is called **yatzy**. If you have exactly **5** dice with the same outcome, you get **50** points. Otherwise you get **0** point. Example: If the combination is **11111** or **33333**, you get **50** points.

You will be given some intermediate state of the game. You have to find the maximum possible expected points you can get at the end of all the **12** rounds.

Input

Input starts with an integer **T** ($1 \le T \le 1000$), denoting number of test cases.

For each test case, in the first line there will be **12** space separated integers denoting the already earned points in each scoring zone. The integers will be given serially according to the problem statement. If the point is **-1** in some scoring zone, that means it has not been chosen yet. You can be sure that the already chosen scoring zones will have valid points assigned to them and there will be at least one scoring zone that has not been chosen yet.

In the next line, there will be an integer **S** ($1 \le S \le 3$), denoting the stage you are about to play. If **S = 1**, you are about to play the first stage and no dice is rolled yet. If **S = 2** or **S = 3**, you are in the second or the third stage and you already have a combination and have to roll again.

In the next line, there will be **5** space separated integers denoting the outcome of the dice in the previous stage. If **S = 1**, all the outcome of the dice will be **-1** as there is no previous stage.

Output

For each test case, print a line in this format, **Case X: Y**. Where **X** denotes the test case number and **Y** denotes the maximum possible expected points.

Your answer will be accepted if its relative or absolute error does not exceed 10^{-6}.

Mathematically, if your answer is **A** and the jury's answer is **B**, then your answer is accepted if and only if $\frac{|A-B|}{max(1,|B|)} \le 10^{-6}$.

Sample Input	Output for Sample Input
3 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 1 -1 -1 -1 -1 -1 0 0 0 0 0 0 0 0 0 0 0 -1 1 -1 -1 -1 -1 -1 0 0 0 0 0 0 0 0 0 0 0 -1 3 1 1 1 1 2	Case 1: 200.469963431 Case 2: 2.301432126 Case 3: 8.333333333

Problem E

Paint The Doors

Input: Standard Input
Output: Standard Output

There are N students in a classroom. Each students has a unique id 1, 2, 3, ... , N. There are M doors in the school which are labeled as 1, 2, ... , M. The doors need to be painted. Each student is assigned to paint some doors. The i-th student is assigned with a range L_i to R_i and he/she will paint all the doors whose labels are between L_i to R_i inclusive. It may happen that a single door is painted by multiple students.
Now you will be given Q queries to answer. Each query will contain four integers L_M R_M L_N R_N. Now you have to answer how many doors between label L_M and R_M are there which are painted by at least one student with id between L_N and R_N inclusive.

Input

First line of the input contains a single integer T ($T \le 5$) denoting the number of test cases. Each test case starts with a line containing two integers N ($1 \le N \le 10^5$) and M ($1 \le M \le 10^5$). Each of the next N lines will contain two integers L_i and R_i ($1 \le L_i \le R_i \le M$) denoting the range associated with the i-th student. Next line will contain a single integer Q ($1 \le Q \le 10^5$) denoting the number of queries. Each of the next Q lines will contain four integers L_M R_M L_N R_N ($1 \le L_M \le R_M \le M$ and $1 \le L_N \le R_N \le N$).

Output

The first line of each test case should contain the case number in the format: **"Case X:"** (without quotes) where X denotes the test case number. After that Q lines follow each containing the answer of the queries.

Sample Input	Output for Sample Input
1 2 10 1 5 6 10 4 3 5 1 2 3 5 2 2 4 7 1 2 4 7 1 1	Case 1: 3 0 4 2

Problem F

Input: Standard Input
Output: Standard Output

Winter is Coming

Winter is coming and the seven kingdoms must prepare for what it brings. The army of white walkers is marching closer, led by the Night King. Apocalypse is near and It looks like the end of the world. The Starks, Lannisters and Greyjoys must put aside their hatred for each other and bond together if the white walkers are to be defeated.
There are **N** strongholds in the seven kingdom, each of which must be occupied by the force of one of the three houses: House Stark, House Lannister, and House Greyjoy. No two houses can occupy the same stronghold. Also, there are **M** tunnels connecting a pair of different strongholds. A tunnel connecting strongholds can be used in both ways. For strategic reasons, the leadership agrees that it is not wise to place forces from the same house on two strongholds connected by a tunnel.
But alas, it is not easy to find such an assignment. There is little time to spare, and winter is coming. The kingdom needs to find a valid assignment of the houses to the strongholds before the white walkers attack. Given a description of the strongholds and the tunnels, you need to find out whether it is possible to assign houses to all the strongholds satisfying all the constraints mentioned above.

Input

The very first line contains an integer **T**, denoting the number of test cases.
Each test case starts with two integers **N** and **M**, denoting the number of strongholds and the number of tunnels connecting strongholds respectively. This is followed by a line containing the names of **N** different strongholds separated by a single space. Names will contain only alphanumeric characters, are case sensitive, will be distinct and won't consist of more than **30** characters. The following **M** lines will contain the description of the tunnels, each of these line will have names of two different strongholds mentioned before, implying they are connected via a tunnel.
There will be a blank line before the start of every test case.

Constraints

- **1 ≤ T ≤ 30**
- **1 ≤ N ≤ 30**
- **1 ≤ M ≤ 330**
- **1 ≤ |Names| ≤ 30**

Output

For each test case, first print the case number (starting from 1), followed by **"Valar Morghulis"** if it is not possible to assign houses to all the strongholds and the kingdom is doomed. Otherwise, print **"Valar Dohaeris"** after the case number if there is hope. Refer to the sample i/o section for further details.

Sample Input

```
3

3 3
Winterfell KingsLanding TheWall
Winterfell KingsLanding
Winterfell TheWall
KingsLanding TheWall

6 10
KingsLanding Dorne Braavos CasterlyRock Dragonstone IronIslands
Dorne Braavos
Braavos CasterlyRock
CasterlyRock Dragonstone
Dragonstone IronIslands
IronIslands Dorne
KingsLanding Dorne
KingsLanding Braavos
KingsLanding CasterlyRock
KingsLanding Dragonstone
KingsLanding IronIslands

2 1
Mistwood Cornfield
Mistwood Cornfield
```

Output for Sample Input

```
Case 1: Valar Dohaeris
Case 2: Valar Morghulis
Case 3: Valar Dohaeris
```

Problem G

Input: Standard Input
Output: Standard Output

A Thousand Push Ups

Jake and Rosa are best buddies. They work in the police force of a particular precinct of a particular city of a particular country. They have investigated many cases as partners. After solving each case, they have to go to the city of the HQ(headquarter) from their own city to submit the report. For going to the HQ, Rosa always drives her motorbike and Jake rides with her. Rosa loves her bike. But the thing is, Jake hates riding on her bike more than anything! From the HQ, they travel back by train and Rosa's bike is transported as cargo on the train.

One day, Jake was saying that he dreaded riding on Rosa's bike so much that for many roads in the country, he remembered **at least** how many times they have used them for going to the HQ after solving cases. He also said that even for many cities, he remembered **at least** how many times they have visited those cities while going to the HQ after solving cases. In reply, Rosa said that she remembered **at most** how many times they have used many particular roads and also **at most** how many times they have visited many particular cities while going to the HQ after solving cases. Now based on this, Savant, a hacker who works in their precinct said that he could figure out **the minimum** and **the maximum** number of cases Jake and Rosa could have solved together. So he asked for two lists from Jake and Rosa. Since Savant is a weird guy but is also a brilliant hacker, Jake and Rosa did not waste any more time asking how he was going to do that and instead created their lists and gave him.

Formally there are **N** cities in the country and **M unidirectional** roads. The cities in the country are numbered from **1** to **N**. Jake and Rosa live in the city numbered **1**. The HQ is placed in the city numbered **N**. In this country, from any city **u**, it is **impossible to return to u** visiting other cities by road.

In Jake's list, there are **A** items in one of the following formats:

0 e x *(meaning the e-th* road is used **at least x** times)

1 u x *(meaning the u-th* city is visited **at least x** times)

In Rosa's list, there are **B** items in one of the following formats:

0 e x *(meaning the e-th* road is used **at most x** times)

1 u x *(meaning the u-th* city is visited **at most x** times)

Now your task is to do what Savant said he could do. Find out **the minimum** and **the maximum** number of cases they could have solved! Oh, there is one more thing that Savant knows, which is, Jake and Rosa have not solved more than **10^9** cases. So even if it is possible in a situation that they can solve more than 10^9 cases, Savant would report the maximum number of cases to be **10^9**.

Input

There are T test cases. For each case, in the first line, there are two integers, **N** and **M**. **M** lines will follow, each containing two integers **a** and **b** denoting there is a road from the *a-th* city to the *b-th* city. The roads are numbered as given in the input with **1-based indexing**. After that, the next line will contain one integer, **A**. Then there will be **A** lines where each line will contain three integers representing an item of Jake's list. Then the next line will contain the integer, **B**. Then there will be **B** lines where each line will contain three integers representing an item of Rosa's list.

Constraints

- **$1 \le T \le 100$**
- For 80% of the test cases, **$2 \le N \le 10$**
- For 20% of the test cases, **$2 \le N \le 100$**
- $1 \le M \le \frac{N \times (N-1)}{2}$
- **$0 \le A, B < N+M$**
- **$0 \le x$** in any item of any list **$\le 10^9$**
- **$1 \le e$** in any item of any list **$\le M$**
- **$2 \le u$** in any item of any list **$\le N$**
- All roads are distinct.
- There is no self-loop.
- No two items in the same list would be about the same city or the same road.
- There is no road that originates from the city of the HQ.

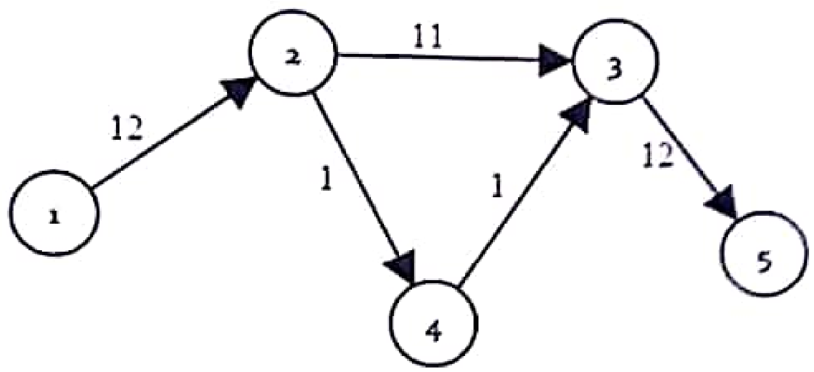
Output

For each case, state the case number and then print the answer for that case. If there is no way that all the items in the two lists are correct, you should just print "**-1**" (without the quotes). Otherwise, if there is at least one way to make sure that all the items in the two lists can be correct, you should print **the minimum and the maximum** number of cases Jake and Rosa could have solved together according to their lists.

Sample Input	Output for Sample Input
1 5 5 1 2 2 3 2 4 4 3 3 5 3 0 1 10 0 4 1 1 2 12 4 0 1 15 0 4 1 0 2 13 1 3 16	Case 1: 12 14

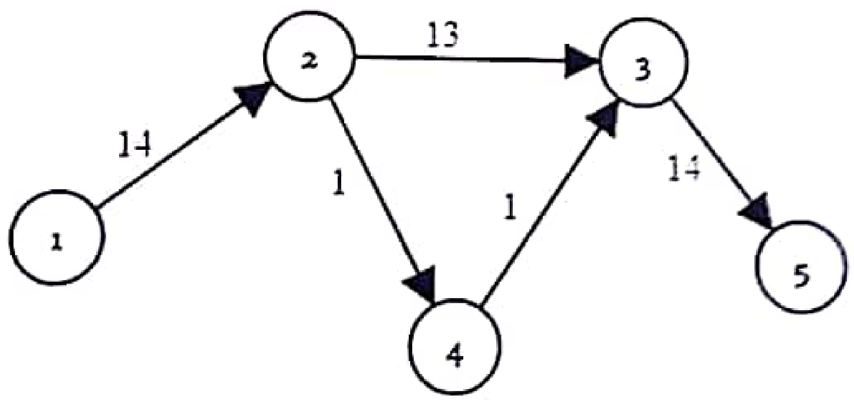
Explanation

A possible way where Jake and Rosa have solved 12 cases is shown below



Here, each circle represents a city and the number of the city is written inside the circle. The number 11 written above the road from 2 to 3 signifies that after solving 11 cases, they used this road on their route to the HQ. There is no other way to solve less than 12 cases in the given test case.

A possible way where Jake and Rosa have solved 14 cases is shown below.



There is no other way to solve more than 14 cases in the given test case.

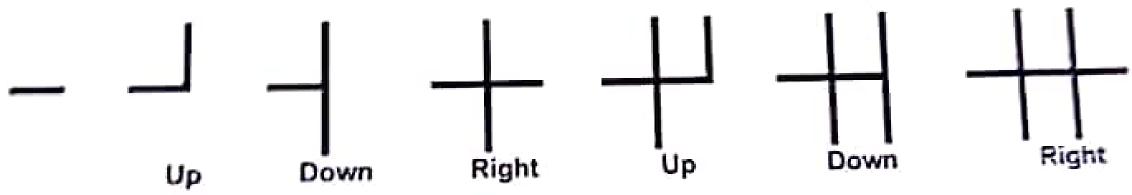
Problem H

Toothpicks

Input: Standard Input
Output: Standard Output

There lives a certain boy somewhere in the world who loves problem solving, spends his time alone thinking about different problems and is obsessed about playing with small objects. Whenever he goes to a restaurant, after ordering food, while people chatter indistinctly, he usually keeps quiet, listening. In the meantime, he subconsciously moves the utensils (plates, forks, spoons, tissue holders etc) on the table so that everything is in a formed manner. That's why some of his close friends refer that he has a little OCD!

One day, in a restaurant, he found a box of toothpicks in his table. He brought some of those and started playing. He placed the toothpicks in a specific order so that each two create a right angle (90 degrees) at their common end. He formed a long connected structure with those, like contiguous plus signs as shown in the image. Suddenly he came up with this problem: 'How many right angles can be formed in this way using **N** toothpicks?' The answer certainly was not **N-1**. The following image (left to right) shows the order in which he would place the first 7 toothpicks. Here, the number of 90 degree angles are 0, 1, 2, 4, 5, 6 and 8 respectively. Notice that, after the first toothpick the order of placement is Up, Down, Right, Up, Down, Right, Up and so on.



While he is busy playing, we want to do the calculations. And by we, we mean you! You will be given the number of toothpicks available. You have to find the number of 90 degree angles formed if those toothpicks are organized according to the boy's idea.

Input

Input will start with a positive integer, **T (≤ 100)** denoting number of test cases. Each case will contain a single integer **N (1 ≤ N ≤ 10000)** in a line denoting the number of toothpicks.

Output

For each case, print the case number first in the format **"Case T: "** without the quotes. Then print an integer denoting the number of 90 degree angles formed. Then print a new line. See sample I/O for clarification.

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

Problem I

Input: Standard Input
Output: Standard Output

Mysterious Lock Maker

Mr. Bokkor is a lock maker. Each key of his locks is a series of alphabet [a-z]. He is called mysterious because he tries to keep the key of the lock from the english dictionary so that people who buys his lock can easily remember the key. Recently, the association of lock maker has introduced some rules for every lock maker in his country. Every lock maker is given a signature. The signature contains only alphabet [a-z]. While making a lock, the maker has to make sure that the key can be built from the signature after deleting zero or more alphabets from the signature. You can not add or rearrange the alphabets in the signature while making keys.

Now Mr. Bokkor has a serious problem while making key for a lock. Because he chooses the key from english dictionary and needs to check if the key is compatible with the regulation of the association. So he is asking your help to check the validity of keys. He will give you the signature and a dictionary. You need to check if the words of the dictionary is a valid key or not.

Input

First line of the input contains an integer **T** **($1 \le T \le 100$)**. The number of test case. For every test case first line contains a string S. **($1 \le |S| \le 10^5$)**. Second line contains an integer **N**, number of words in dictionary. Next N lines contain the words of the dictionary. One word in one line.

Output

For every test case there will be **N+1** line of output.

First line of the output contains a string "**Case X:**", where X is the test case number.

For each of the next **N** lines, print '**valid**' if the i-th word of the dictionary can be used as a key, otherwise print '**invalid**'. See sample I/O for clarification.

Constraints

- **$1 \le T \le 100$**
- **$1 \le |S| \le 10^5$**
- **Summation of the length of words in all test cases $\le 2 \cdot 10^5$**

Sample Input	Output for Sample Input
1 aabcxyza 5 abc abca aaxya aaaa aaxzy	Case 1: valid valid valid invalid invalid