Greetings From Globussoft

- Given below are 5 Programming questions, you have to solve any 3 out of 5 questions.

- These 5 questions you can attempt in any technology like C/C++, java, .Net, PHP

- To solve these 3 questions you've max. 3 hours.

- While Solving these questions you are not allowed to use any Search Engine like Google, Yahoo, Bing ...

All the best for your test

Globussoft

QUESTION - 1

Thor, the Norse god of thunder, was shopping for groceries when he noticed a sale on Kleenex brand tissues. This got him thinking about Kleene's recursion theorem and its application to quines in functional programming languages. As this gave him a headache, he instead turned his attention to how one might recognise regular expressions with Kleene stars on a Turing machine. Unfortunately, this just made his headache worse. So he took out a slip of paper, jotted down a brainf**k program to handle regular expressions containing Kleene plusses, paid for his groceries, and congratulated himself on a job well done.

Note: You can use any programming language you want, as long as it is brainf**k.

Input

The first line contains an integer **T** ($1 \leq \mathbf{T} \leq 1000$). Then follow **T** test cases.

For each test case: The first line contains a regular expression **P** ($1 \leq |\mathbf{P}| \leq 30$). The next line contains an integer **Q** ($1 \leq \mathbf{Q} \leq 10$). Then follow **Q** lines, each containing a string **S** ($1 \leq |\mathbf{S}| \leq 100$). Finally, there is an empty line at the end of each test case.

Each line, including the last, is terminated by a single newline (linefeed) character, which has ASCII value 10.

All regular expressions are guaranteed to be valid; in particular, **P** may not start with a plus, and it may not contain two consecutive plusses. **P** is a string over the alphabet {a,b,c,d,+}, and **S** is a string over the alphabet {a,b,c,d}.

Output

T lines each containing a string of length **Q**. The **i**th character of the string indicates whether **S** is in the regular language defined by **P**: 'Y' for a match, and '.' otherwise. Note that we are concerned whether **P** matches **S**, as opposed to a substring of **S**. In other words, we could insert '^' at the beginning of **P** and '$' at the end, and then test for a match using e.g. m// in Perl. See the example for further clarification.

Example

Input:

```
3
a
2
a
aa

a+
2
```

```
a
aa

a+bc
6
abbacadabba
aaaabc
abc
bc
abcd
babc
```

Output:

```
Y.
YY
.YY...
```

QUESTION – 2

You've come across N ($1 \leq N \leq 200$) adorable little Foxlings, and they're hungry! Luckily, you happen to have M ($1 \leq M \leq 200$) crackers on hand, and everyone knows that Foxen love crackers! You'd like to distribute all of your crackers, without splitting any of them, among the Foxlings - but you have to be careful. Foxling i must be fed at least A_i crackers, or it will remain hungry, but no more than B_i of them, or it will become hyper ($1 \leq A_i \leq B_i \leq 200$). You certainly don't want any hungry or hyper Foxlings on your hands, and you're curious as to how many ways this can be accomplished.

There are T ($1 \leq T \leq 100$) scenarios as described above. For each one, you'd like to determine the number of different distributions of your crackers that would satisfy all of the Foxlings, modulo 10^9+7 (as this value can be quite large).

Input

First line: 1 integer, T

For each scenario:

First line: 2 integers, N and M

Next N lines: 2 integers, A_i and B_i, for $i=1..N$

Output

For each scenario:

Line 1: 1 integer, the number of valid cracker distributions modulo 10^9+7

Example

```
Input:
2
2 5
1 4
2 6
3 5
2 2
2 9
2 3

Output:
3
0
```

QUESTION – 3

Lucy has made too many friends but she does not know how many friends are in her circle. Assume that every relation is mutual. If Lucy is Patty's friend, then Patty is also Lucy's friend. Your task is to help Lucy in keeping track of each person's circle size.

Input Specification

The first line of input contains one integer T (*T<=10*) specifying the number of test cases to follow. Each test case begins with a line containing an integer N *(N<=100000)*, the number of new relations. Each of the following *N* lines contains couple of strings denoting the names of two people who have just formed relation, separated by a space. Names will have no more than *20* characters.

Output Specification

Print a line containing one integer, the number of people in the combined circle of two people who have just become friends.

Input

```
1
4
Lucy Patty
Patty Alice
Alice Mira
Tiffany Jayden
```

Output

```
2
2
3
4
2
```

QUESTION – 4

Kingdom of Rohan is under attack! There are N vital army stations. King will decide what army should be guarding what station, to get the best strategic advantage against Sauron attacks. All armies are already in some stations, but not necessarily the stations required by the king. As a result armies will have to be moved. Distances between any pair of stations are known. They are not necessarily symmetrical, because road from station A to B could be different than road from B to A. When a army moves, it doesn't have to take a direct road and instead can choose to cross other stations, if that results in a shorter path.

Given the distances between stations and king's relocation orders, find the minimum total travel distance for all the armies.

Input

First line contains an integer **T**, number of test cases. Then the description of **T** test cases follow. Each test case starts with an integer **N**, which is the total number of army stations. Next **N** lines have **N** integers each, description of distances. **b**'th integer on line **a** is the distance from station **a** to station **b**. Description of kings orders follows. In a first line, a single integer **R**, number of orders. Next **R** lines will contain two integers **s** and **d** each, describing an order to move an army from station **s** to **d**.

Constrains:

1 <= T <= 50

1 <= N, R <= 50

1 <= distance <= 10^6

1 <= s, d <= N

Output

Print a single line for each test case. A string "Case #t: " without quotes, where **t** is the number of test case, starting from 1. Following the string, you must print the total distance armies must travel during relocation.

0 <= total distance <= 10^7

Example

Input:

```
1
3
0 1 1
1 0 1
1 9 0
2
2 1
3 2
```

Output:

```
Case #1: 3
```

QUESTION – 5

The Bloons (not to be confused with balloons) are attacking! They are attempting to navigate your course of L ($1 \le L \le 1000$) cells, laid out in a row and numbered from 1 to L. You don't know what they'll do to you if they manage reach the end, and you don't want to find out! To that end, you've constructed some defensive towers along the course. You might say that this is a Bloons Tower Defense.

There are N ($1 \le N \le 1000$) towers ready to take out any Bloons that get close. The ith tower is located next to cell C_i ($1 \le C_i \le L$), and can launch darts at any Bloons that are no more than R_i ($0 \le R_i \le 1000$) cells away - that is, Bloons in cells $C_i - R_i$ to $C_i + R_i$, inclusive. Every second, it will do D_i ($1 \le D_i \le 10^9$) HP worth of damage to any Bloons in this range.

M ($1 \le M \le 1000$) Bloons will attempt to float through your course, one after another. The ith Bloon begins with H_i ($1 \le H_i \le 10^9$) HP, and will pop as soon as it has taken at least that much damage in total. Each Bloon starts in cell 1, and moves along the course at a speed of 1 cell per second. If a Bloon moves past cell L, it safely exits the course and can no longer be popped.

There are T ($1 \le T \le 20$) scenarios as described above. For each, you'd like to determine how far along the course each of the M Bloons will make it.

Input

First line: 1 integer, T

For each scenario:

First line: 2 integers, L and N

Next N lines: 3 integers, C_i, R_i, and D_i, for $i=1..N$

Next line: 1 integer, M

Next M lines: 1 integer, H_i, for $i=1..M$

Output

For each scenario:

M lines: If the ith Bloon will survive the course, the string "Bloon leakage" (without quotes) - otherwise, 1 integer, the furthest cell which the ith Bloon will reach, for $i=1..M$

Example

Input:

```
1
10 3
3 3 1
4 0 4
10 2 2
4
1
20
9
11
```

Output:

```
1
Bloon leakage
5
8
```