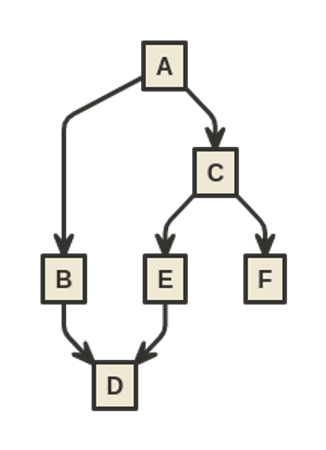
Microservice architecture has gained more attention recently. In microservice architecture, one service needs to call other services (its dependencies) in order to get the data it requires to process a request. If one of its dependencies fails, it might not be able to process the request.

**Example dependency diagram:**



In the above example, A depends on B and C, and C depends on E and F, and so on.

In order to ensure that each of the microservices functions correctly, a monitoring system is used. Alerts are triggered whenever errors are detected in a service. The problem is, if a service fails, the service depending on it might fail as well, triggering several alerts. Having multiple alerts at the same time makes it difficult to know which service failed first. For example, if the alert for F is triggered, the alert for C and A might (or might not) be triggered as well.

Your task is to write software that can identify which service (or services) is the root cause of the failure. A failure is a root cause only when there are no other direct or transitive dependencies which also fail.

**Input format**

1st line: number of different test cases (NC).

2nd line: number of services (NS), followed by the name of each service separated by a space. The service names contain uppercase English letters only, underscore characters and numbers. Service names will be provided in lexicographical order.

3rd line: number of dependency relationships (ND), followed by the dependencies in the format of "A,B" meaning "A depends on B." The letters should be separated by a space.

4th line: number of different alert cases (NA).

5th line: number of services with alerts, followed by the name of each service separated by a space.

\* The 5th line repeats the number of times indicated on the 4th line.

\* The 2nd through 5th lines repeat the number of times indicated on the 1st line.

**Constraints**

1 ≤ NC ≤ 5

1 ≤ NS ≤ 1000

1 ≤ ND ≤ 2000

1 ≤ NA ≤ 2000

Services shall not have direct or indirect circular dependency. In other words, if A depends on B, then B will not depend on A.

There will not be any duplicate dependencies in the input.

**Output format**

For each case, print the number of services that are identified as the root cause followed by the name of the services arranged lexicographically separated by spaces.

**Example Input**

1   
6 A B C D E F   
6 A,B A,C C,E C,F B,D E,D   
4   
1 E   
3 A C E   
2 A D   
3 D E F

**Example Output**

1 E   
1 E   
1 D   
2 D F

**Explanation of Sample Output**

The input has 1 case, 6 services, 6 connections and 4 alerts. It is the case shown in the diagram in the description.

In the first case, only service E has an alert. Thus, E is the root cause.

In the second alert case, services A, C and E have alerts. Since A depends on C and C depends on E, we don’t consider them the root cause. E has no dependency and has an alert. Thus, E is the root cause.

In the third case, services A and D have alerts. Since A depends on D through C, B and E, we still consider that the failure on service D might have caused a failure on service A even if the alerts on services C, B and E did not fire. Thus, D is the root cause.

In the fourth case, services D, E and F have alerts. E depends on D, so E is not the root cause. D and F do not depend on any services which failed, so they both are the root causes. This example shows that there can be multiple root causes. D must be listed in the answer before F because the output is expected in lexicographical order.