The social networking sensation, SoCBook, has decided to implement a new policy to protect community standards: they will pay users to spy on their friends and report inappropriate behaviour. The spies do not need to be watched – SoCBook assumes that since they are being paid they will always behave properly – but it is ok if two people who are friends are both hired.

Of course SoCBook wants to hire the smallest number of spies possible and still keep an eye on everyone, and that is where you come in. Your assignment is to create an algorithm that will find the smallest set of spies so that everyone who is not a spy has at least one friend who is a spy.

More formally, given a graph G, you are to find the smallest set S of vertices so that every vertex is either in S, or is directly joined to at least one vertex in S (this is called the "dominating set" problem).

You are free to choose any algorithmic paradigm you like. I should mention, however, that this is an NP-Complete problem.

Input consists of a text file giving one or more instances of the problem. The first line contains an integer identifying the number of instances. The remainder of the file consists of sets of lines. The first line in a set gives the number of vertices in the graph. The rest of the lines in the set give the adjacency matrix for the graph.

For example, the input might look like

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

Your output should list, for each instance, the vertices chosen for the set of spies.

For the sample input shown above, the output should look something like this (assuming the vertices are numbered from 1 to n). The answer for Instance 1 is not unique – you do not need to find all optimal solutions.

Instance 1:
        vertex 1
        vertex 4

Instance 2:
        vertex 2