563 Crimewave

Nieuw Knollendam is a very modern town. This becomes clear already when looking at the layout of its map, which is just a rectangular grid of streets and avenues. Being an important trade centre, Nieuw Knollendam also has a lot of banks. Almost on every crossing a bank is found (although there are never two banks at the same crossing). Unfortunately this has attracted a lot of criminals. Bank hold-ups are quite common, and often on one day several banks are robbed. This has grown into a problem, not only to the banks, but to the criminals as well. After robbing a bank the robber tries to leave the town as soon as possible, most of the times chased at high speed by the police. Sometimes two running criminals pass the same crossing, causing several risks: collisions, crowds of police at one place and a larger risk to be caught.

To prevent these unpleasant situations the robbers agreed to consult together. Every Saturday night they meet and make a schedule for the week to come: who is going to rob which bank on which day? For every day they try to plan the get-away routes, such that no two routes use the same crossing. Sometimes they do not succeed in planning the routes according to this condition, although they believe that such a planning should exist.

Given a grid of $(s \times a)$ and the crossings where the banks to be robbed are located, find out whether or not it is possible to plan a get-away route from every robbed bank to the city-bounds, without using a crossing more than once.

Input

The first line of the input contains the number of problems p to be solved.

- The first line of every problem contains the number s of streets ($1 \leq s \leq 50$), followed by the number a of avenues ($1 \leq a \leq 50$), followed by the number b ($b \geq 1$) of banks to be robbed.
- Then b lines follow, each containing the location of a bank in the form of two numbers x (the number of the street) and y (the number of the avenue). Evidently $1 \leq x \leq s$ and $1 \leq y \leq a$.

Output

The output file consists of p lines. Each line contains the text `possible` or `not possible`. If it is possible to plan non-crossing get-away routes, this line should contain the word: `possible`. If this is not possible, the line should contain the words `not possible`.

Sample Input

```
2
6 6 10
4 1
3 2
4 2
5 2
3 4
4 4
5 4
3 6
4 6
5 6
5 5 5
3 2
2 3
3 3
4 3
3 4
```

Sample Output

```
possible
not possible
```