Your city mayor have finally decided to end the misery of all the residents and introduced a free bus service. The service will consist of two buses, going through various roads in the city in a circular path.

As the chief architect of this design, you are assigned to check if a given design has any fault. A design will be faulty, if there is such a state that both buses visit the same cell in the city at the same time.

The city was originally designed like a rectangular grid (30 × 30). Both of the buses can move through the city, from one cell to another, in 4 directions (up, down, left, right). Current cell and the next destination cell will always be situated either parallel to X-axis or Y-axis. Also, the distance between the cells will at least be one. Two buses start their journey at the same time and can move one cell at a time in the direction of its next destination.

Suppose, the route of a bus contains p_0, p_1, p_2, ..., p_{n-1}. Then this bus will start its journey from p_0 and continue through the grid to reach p_1, then p_2, ..., p_{n-1}, p_0, p_1, p_2, ... (continuing for infinite times).

Given the route for both buses, you are to determine if the route design is faulty or not. Remember that the buses will not collide when they cross each other at the cell border, hence the design will not be faulty.

Input

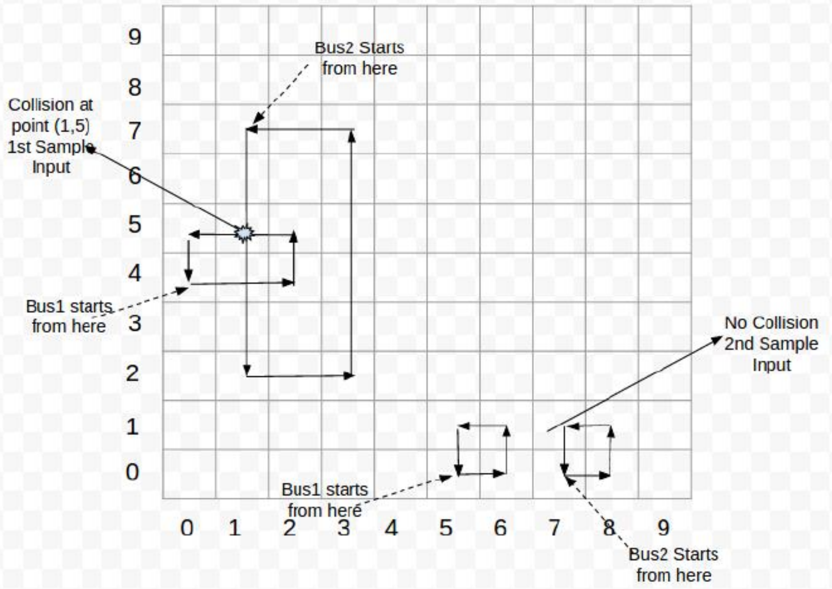
Input start with an integer T ($T \leq 20$), denoting the number of test cases.

Each test case starts with a blank line. The next line contains an integer N ($2 \leq N \leq 30$) denoting the number of destinations the first bus will visit. Each of the next N lines will contain two integers x_i, y_i ($0 \leq x_i, y_i < 30$) denoting the coordinate of a cell. The next line contains an integer M ($2 \leq M \leq 30$) denoting the number of destinations the second bus will visit. Each of the next M lines will contain two integers x_i, y_i ($0 \leq x_i, y_i < 30$) denoting the coordinate of a destination cell.

Output

For each case, print the case number and (x, y) which is the cell where first collision between the buses. If there is no collision then print '`No Collision`'.

Note: The picture on the right illustrate the sample cases below.



Sample Input

```
2

4
0 4
2 4
2 5
0 5
4
1 7
1 2
3 2
3 7
4
5 0
6 0
6 1
5 1
4
7 0
8 0
8 1
7 1
```

Sample Output

```
Case 1: 1 5
Case 2: No Collision
```