John recently has had his birthday party. As John likes traveling, all invited guests cooperated and decided to buy various tickets as a present. Each ticket can be used only once to travel from some city A to city B or vice versa. As guests were cooperating, they decided that no tickets will be between the same cities.

Now John has huge amount of tickets and needs to plan his trip. But first, he wishes to know the cities that now he can possibly visit and come back home. In other words you have to find all cities John can visit by making a round trip from the home city through that particular city.

Input

The number of tests T ($T \leq 100$) is given on the first line. Each test starts with 3 integers N M C ($N \leq 100$; $M \leq 1000$; $1 \leq C \leq N$). Where N stands for number of cities (cities are numbered from 1 to N) and M is number of tickets John has. C describes his living city number. Next M lines describe tickets with 2 integers X Y ($1 \leq T \leq 100$). X and Y are cities on the ticket.

Output

For each test case output a single line '`Case` T`:` S'. Where T is the test case number (starting from 1) and S is single space separated list of cities that John could possibly visit. This list must be sorted in increasing order. If John can't find a single city to visit print '`none`' intead (i.e. `Case 2: none`).

Sample Input

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

Sample Output

```
Case 1: 2 3 4 5 6
Case 2: none
```