Seven segment numeric displays are ubiquitous. It uses seven segments to display numbers.

Here is a figure which depicts all the segments used in a typical seven segment display (We'll be using the acronym SSD for convenience from now on).

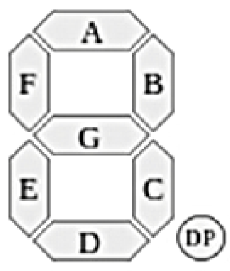
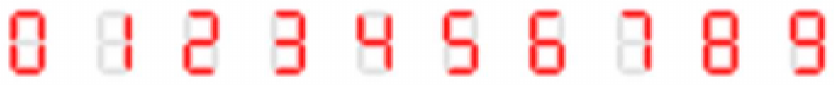


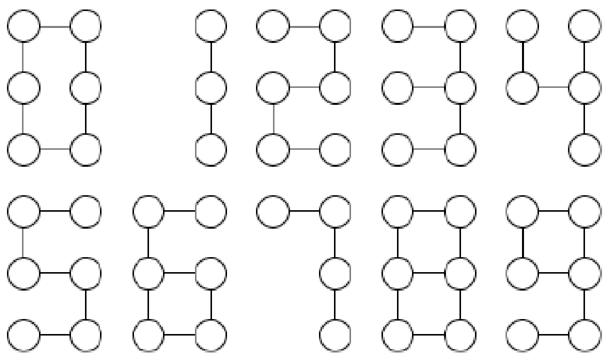
Figure 1: Segments used for SSD representation.
Here, DP represents decimal place which is not necessary in the context of this problem.

And here are the numbers from 0 to 9 represented in SSD.



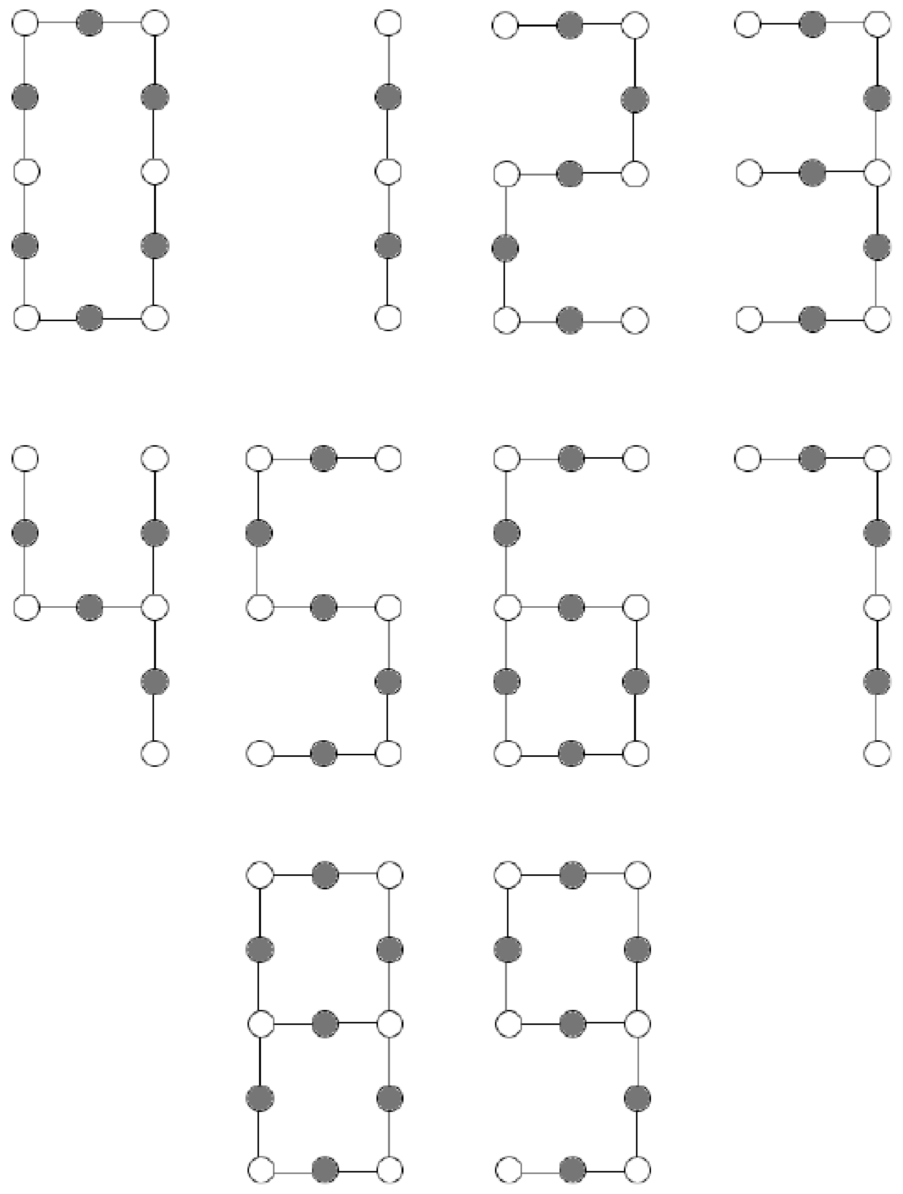
0 uses segments A, B, C, D, E, F
1: B, C
2: A, B, G, E, D
3: A, B, C, D, G
4: B, C, F, G
5: A, C, D, F, G
6: A, C, D, E, F, G
7: A, B, C
8: A, B, C, D, E, F, G
9: A, B, C, D, F, G

Now, imagine the SSD representation of a digit as a graph. The endpoints of the segments are the nodes and segments are edges. So, the digits will look like:



We call this representation a 0-degree SSD graph. A k-degree ($k > 0$) SSD graph is made by dividing each edge of a 0-degree graph into $k+1$ edges and introducing k nodes in between them.

To explain more, 1-degree graphs of all digits are shown below. The darker nodes are the newly introduced nodes.



You'll be given a graph with n nodes and m edges. You'll need to print all the (degree, digit) pairs for which the given graph is valid.

Input

The first line of the input contains an integer which denotes the number of test cases T ($1 \le T \le 20$). T sets of case will follow. Each case will start with a couple of numbers n ($1 \le n \le 500$) and m ($1 \le m \le 1000$) — the number of nodes and the number of edges respectively. Each of the next m lines will contain a pair of numbers (u, v) meaning that there is an edge from node u to node v. Nodes are numbered from 1 to n. It's guaranteed that there is no duplicate or self-edges in the input.

Output

For each set of inputs, output one set of output. First line of a set should be of the format, '`Case X: Y`' (here, X is the serial of the input and Y is the number of $(digit, degree)$ pairs) in a line. Then print each $(digit, degree)$ pair — one pair in each line. The pairs should be sorted according to digit first then degree. Each number in a pair should be separated with a space. Print a blank line between consecutive test cases.

Sample Input

```
2
16 15
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 10
10 11
11 12
12 13
13 14
14 15
15 16
4 3
1 2
1 3
1 4
```

Sample Output

```
Case 1: 3
2 2
5 2
7 4

Case 2: 0
```