Graph:

1. Implement BFS on an undirected graph. For each node, print the shortest distance path from source node. Take the input in same format of problem 2.

2. Let G = (V, E) be a **directed** graph. Write the Depth-first search (DFS) algorithm of graph G using stack to eliminate recursion. Nodes should be considered in alphabetical order. Your algorithm should print the depth first tree edges in the order they are explored by your algorithm. **[10]**

Use adjacency list to represent the graph.

Input format: First line will contain two numbers **n**, number of nodes and **e**, number of edges. Next e lines will contain edges of the form **u v** where u is source node and v is destination node.
Output format: For part 1, tree edges of the form (u, v).

Test Case:
Input:
10 14
q s
s v
v w
w s
q w
q t
t x
x z
z x
t y
y q
r y
r u
u y

Output:
For part 1:
(q, s), (s, v), (v, w), (q, t), (t, x), (x, z), (t, y), (r, u)

3. Let G = (V,E) be **a <u>directed</u>** graph. For every pair of nodes (i,j) determine if there is a path from i to j. You are to read the number of nodes n, the number of edges e and then read in the list of edges. Print the result in a matrix form. You should test your program carefully on various kinds of inputs, especially in the presence of cycles. You may use adjacency list or adjacency matrix to store this graph.

4. There are two types of professional wrestlers: **"Good Guys"** and **"Bad Guys".** Between any pair of professional wrestlers, there may or may not be rivalry. Suppose we have **n** professional

wrestlers and we have a **list of r pairs** of wrestlers for which there are rivalries. Give an **O(n + r)** time algorithm that determines whether it is possible to designate some of the wrestlers as good guys and the remainder as bad guys such that each rivalry is between a good guy and a bad guy. If it is possible to perform such a designation, your algorithm should produce it, **"Not Possible"** otherwise.

Input: First line will contain two numbers n and r. Each of the following r will contain 2 numbers denoting wrestlers who are rival.
Output: Two disjoint sets Good Guys and Bad Guys, "Not Possible" otherwise.

Test1: Input: 6 5 2 3 2 1 2 4 2 5 2 6 Output: Good Guys: 2 Bad Guys: 1 3 5 4 6	**Test2:** Input: 8 7 1 5 2 5 2 7 2 6 3 6 4 6 4 8 Output: Good Guys: 1 2 3 4 Bad Guys: 5 6 7 8
Test3: Input: 5 5 1 2 1 3 2 3 3 4 4 5 Output: Not Possible	

5. In support of the Bharat bandh, some engineers got even Internet bots to stop. One such bot was a web crawler. But the good news is that before it was stopped, it had already collected some data about the linkage of the web pages, which is provided to you as input. But this data needs to be processed more.
   You are tasked to find all groups among these web pages; A **group** is such that, from a web page in that group we can reach any other page of that group.

For simplicity, a number represents a web page. The first line of Input contains the total number of web pages, following input lines contain linkage from a page to the other. Each output line represents a group (ordering of pages in a group does not matter).

Input:

```
12
0 1
1 2
1 3
1 4
2 5
4 1
4 5
5 2
5 7
4 6
6 7
6 9
7 10
8 6
9 8
10 11
11 9
```

Output:

```
0
1 4
3
2 5
7 6 8 9 11 10
```