```
import java.util.Map;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.LinkedList;
import java.util.Queue;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;

class VertexOutOfRangeException extends Exception {
        public String toString() {
                return "Valid input for the vertex in specified range is expected!";
        }
}



public class Solution {

    static BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        public static void connectedComponents(Map<Integer, ArrayList<Integer>> adjacencyList, Map<Integer, Boolean> visited,
ArrayList<Integer> smallOutput, int vertex)  {
                smallOutput.add(vertex);
                visited.put(vertex, true);

                ArrayList<Integer> adjVerticies = adjacencyList.get(vertex);

                for(int i = 0; i < adjVerticies.size(); i++) {
                        if(!visited.containsKey(adjVerticies.get(i))) {
                                connectedComponents(adjacencyList, visited, smallOutput, adjVerticies.get(i));
                        }
                }
        }

        public static ArrayList<ArrayList<Integer>> allConnectedComponents(Map<Integer, ArrayList<Integer>> adjacencyList) {
                Map<Integer, Boolean> visited = new HashMap<>();
                ArrayList<ArrayList<Integer>> output = new ArrayList<>();

                int i = 0;
                while(visited.size() != adjacencyList.size()) {
                        while(i < adjacencyList.size()) {
                                if(!visited.containsKey(i)) {
                                        ArrayList<Integer> smallOutput = new ArrayList<>();
                                        connectedComponents(adjacencyList, visited, smallOutput, i);
                                        output.add(smallOutput);
                                }
```

```
                        i += 1;
                }
        }

        return output;
}


public static void main(String[] args) throws VertexOutOfRangeException, IOException{
String[] strNums;
strNums = br.readLine().split("\\s");
int noOfVertices = Integer.parseInt(strNums[0]);
int noOfEdges = Integer.parseInt(strNums[1]);


        Map<Integer, ArrayList<Integer>> adjacencyList = new HashMap<>();

        for(int i = 0; i < noOfVertices; i++) {
                adjacencyList.put(i, new ArrayList<>());
        }

        int currentEntry = 1;

        while(currentEntry <= noOfEdges) {
    String[] strNums1;
    strNums1 = br.readLine().split("\\s");
                int source = Integer.parseInt(strNums1[0]);
                int destination = Integer.parseInt(strNums1[1]);

                ArrayList<Integer> edgeListForDestination = adjacencyList.get(source);
                ArrayList<Integer> edgeListForSource = adjacencyList.get(destination);


                if(edgeListForDestination != null && edgeListForSource != null) {
                        edgeListForDestination.add(destination);
                        edgeListForSource.add(source);
                } else {
                        throw new VertexOutOfRangeException();
                }

                currentEntry += 1;
        }

        ArrayList<ArrayList<Integer>> allConnectedComponents = allConnectedComponents(adjacencyList);

        for(int i = 0; i < allConnectedComponents.size(); i++) {
                ArrayList<Integer> components = allConnectedComponents.get(i);
                Collections.sort(components);
```

```
                        for(int k = 0; k < components.size(); k++) {
                                System.out.print(components.get(k) + " ");
                        }
                        System.out.println();
                }

        }

}
```