```java
import java.util.*;

class Tarjan {
    /** number of vertices **/
    private int V;
    /** preorder number counter **/
    private int preCount;
    /** low number of v **/
    private int[] low;
    /** to check if v is visited **/
    private boolean[] visited;
    /** to store given graph **/
    private List<Integer>[] graph;
    /** to store all scc **/
    private List<List<Integer>> sccComp;
    private Stack<Integer> stack;

    /** function to get all strongly connected components **/
    public List<List<Integer>> getSCComponents(List<Integer>[] graph) {
        V = graph.length;
        this.graph = graph;
        low = new int[V];
        visited = new boolean[V];
        stack = new Stack<Integer>();
        sccComp = new ArrayList<>();

        for (int v = 0; v < V; v++)
            if (!visited[v])
                dfs(v);
        return sccComp;
    }

    public void dfs(int v) {
        low[v] = preCount++;
        visited[v] = true;
        stack.push(v);
        int min = low[v];
        for (int w : graph[v]) {
            if (!visited[w])
                dfs(w);
```

```
41              if (low[w] < min)
42                  min = low[w];
43          }
44          if (min < low[v]) {
45              low[v] = min;
46              return;
47          }
48          List<Integer> component = new ArrayList<Integer>();
49          int w;
50          do {
51              w = stack.pop();
52              component.add(w);
53              low[w] = V;
54          } while (w != v);
55          sccComp.add(component);
56      }
57
58      public static void main(String[] args) {
59          Scanner scan = new Scanner(System.in);
60          System.out.println("Tarjan algorithm Test\n");
61          System.out.println("Enter number of Vertices");
62          int V = scan.nextInt();
63          List<Integer>[] g = new List[V];
64          for (int i = 0; i < V; i++)
65              g[i] = new ArrayList<Integer>();
66
67          /** accept all edges **/
68          System.out.println("\nEnter number of edges");
69          int E = scan.nextInt();
70          System.out.println("Enter "+ E +" x, y coordinates");
71          for (int i = 0; i < E; i++) {
72              int x = scan.nextInt();
73              int y = scan.nextInt();
74              g[x].add(y);
75          }
76          Tarjan t = new Tarjan();
77          System.out.println("\nSCC : ");
78          List<List<Integer>> scComponents = t.getSCComponents(g);
79          System.out.println(scComponents);
80      }
81  }
```