

```

import java.util.LinkedList;

public class Kosraju {

    public static void main(String[] args) {
        int V = 7;
        LinkedList<LinkedList<Integer>> adj = new LinkedList<>();
        for (int i = 0; i < V; i++) {
            adj.add(new LinkedList<>());
        }
        addEdge(0, 1, adj);
        addEdge(0, 2, adj);
        addEdge(1, 3, adj);
        addEdge(4, 1, adj);
        addEdge(6, 4, adj);
        addEdge(5, 6, adj);
        addEdge(5, 2, adj);
        addEdge(6, 0, adj);

        System.out.println(motherVertex(adj, V));
    }

    public static int motherVertex(LinkedList<LinkedList<Integer>> g, int v) {
        boolean[] vis = new boolean[v];
        int V = -1;
        for (int i = 0; i < v; i++) {
            if (!vis[i]) {
                dfs_util(g, i, vis);
                V = i;
            }
        }

        boolean[] check = new boolean[v];
        dfs_util(g, V, check);
        for (boolean val : check) {
            if (!val)
                return -1;
        }
        return V;
    }

    public static void addEdge(int u, int v, LinkedList<LinkedList<Integer>> adj) {
        adj.get(u).add(v);
    }

    public static void dfs_util(LinkedList<LinkedList<Integer>> g, int v, boolean[] vis) {
        vis[v] = true;
    }
}

```

```
    for (var U : g.get(v)) {  
        if (!vis[U]) {  
            dfs_util(g, U, vis);  
        }  
    }  
}  
  
}
```