

```

import java.io.*;
import java.util.*;

class TopSort{
    String myFile = "";
    static Vertex [] Myvert = new Vertex[50];
    static int [] ndeg = new int[50];
    static int VSize = 0;

    public static void main(String[] args){
        if (readfile()){
            if(sort()){
                for(int z = 1; z <= VSize; z++){
                    System.out.print("v"+ z + " = " + Myvert[z].topNum);
                    if(z < VSize)
                        System.out.print(", ");
                }
            }
        }
    }

    static boolean readfile(){
        try{
            System.out.print("Please enter a file name: ");
            BufferedReader MyBufferedReader = new BufferedReader (new InputStreamReader(System.in));
            String s = MyBufferedReader.readLine();
            BufferedReader MyBufferedReader2 = new BufferedReader (new FileReader(s));
            s = MyBufferedReader2.readLine() ;
            while (s != null) {
                StringTokenizer t = new StringTokenizer(s);
                int newid = Integer.parseInt(t.nextToken());
                int a = t.countTokens();
                if ( a != 0 ){
                    int[] varray = new int[a];
                    for (int i = 0 ; i < a ; i++){
                        int N = Integer.parseInt(t.nextToken());
                        ++ndeg[N];
                        varray[i] = N;
                    }
                    Myvert[newid] = new Vertex(newid, varray);
                    s = MyBufferedReader2.readLine() ;
                    VSize++;
                }else {
                    Myvert[newid] = new Vertex(newid);
                    VSize++;
                    s = MyBufferedReader2.readLine();
                }
            }
            for(int j = 1; j <= VSize; j++)
                for(int k = 1; k <= ndeg[j]; k++)
                    Myvert[j].addIndegree();

        }catch(Exception E){System.out.println("Data File does not exist or is of improper
            format."); return false;}

        return true;
    }

    static boolean sort() {
        int counter = 0;
        Vertex v, w;
        QueueAr q = new QueueAr();
        for (int i = 1; i <= ( VSize ) ; i++){
            v = Myvert[i];
            if ( v.indegree == 0){
                q.enqueue(v);
            }
        }
        int j = 1;

        while(!q.isEmpty()) {
            Myvert[j] = (Vertex)(q.dequeue());
            Myvert[j].topNum = ++counter;
            for(int u = 0; u < Myvert[j].adjacents.length ; u++){

```

```
        int temp2 = Myvert[j].adjacents[u];
        Myvert[temp2].indegree--;
        if (Myvert[temp2].indegree == 0 )
            q.enqueue(Myvert[temp2]);
    }
    j++;
}
if(counter != VSize){
    System.out.println("Cycle Found!");
    return false;
}
return true;
}
}
```