**ASSIGNMENT 2**

**Name:** Kapil Wavale

Div.: D

**Batch:** B1

**Roll. No.: 79**

**Prn :** 1211021

BFS

Code:

import java.util.\*;

import javax.swing.plaf.synth.SynthOptionPaneUI;

public class BFS {

    static class Edge{

            int source;

            int destination;

            int weight;

            public Edge(int s,int d,int w)

            {

                this.destination=d;

                this.source=s;

                this.weight=w;

            }

    }

    public static void Creation(ArrayList<Edge>graph[],int v)

    {

        for(int i=0;i<v;i++)

        {

            graph[i]=new ArrayList<>();

        }

        //0-vertex

        graph[0].add(new Edge(0,1,1));

         graph[0].add(new Edge(0,2,1));

        //1-vertex

        graph[1].add(new Edge(1, 3, 1));

        graph[1].add(new Edge(1, 0, 1));

        //2-vertex

        graph[2].add(new Edge(2, 0, 1));

        graph[2].add(new Edge(2, 4, 1));

        //3-vertex

        graph[3].add(new Edge(3, 1, 1));

        graph[3].add(new Edge(3, 4, 1));

         graph[3].add(new Edge(3, 5, 1));

         // 4-vertex

         graph[4].add(new Edge(4, 2, 1));

         graph[4].add(new Edge(4, 3, 1));

         graph[4].add(new Edge(4, 5, 1));

         //5-vertex

         graph[5].add(new Edge(5, 3, 1));

         graph[5].add(new Edge(5, 4, 1));

         graph[5].add(new Edge(5, 6, 1));

         graph[6].add(new Edge(6, 5, 1));

    }

    public static void BreathfirstSearch(ArrayList<Edge>graph[])

    {

        Queue<Integer>q=new LinkedList<>();

        boolean visit[]=new boolean[graph.length];

        q.add(0);//we select vertex as source

        while(!q.isEmpty())

        {

            int curr=q.remove();

            if(!visit[curr]) // visited curr

            {

                System.out.println(curr+" ");

                visit[curr]=true;

                for(int i=0;i<graph[curr].size();i++)//add all neighbour all curr

                {

                    Edge e=graph[curr].get(i);

                    q.add(e.destination);

                }

            }

        }

    }

    public static void main(String args[])

    {

         int v=7;

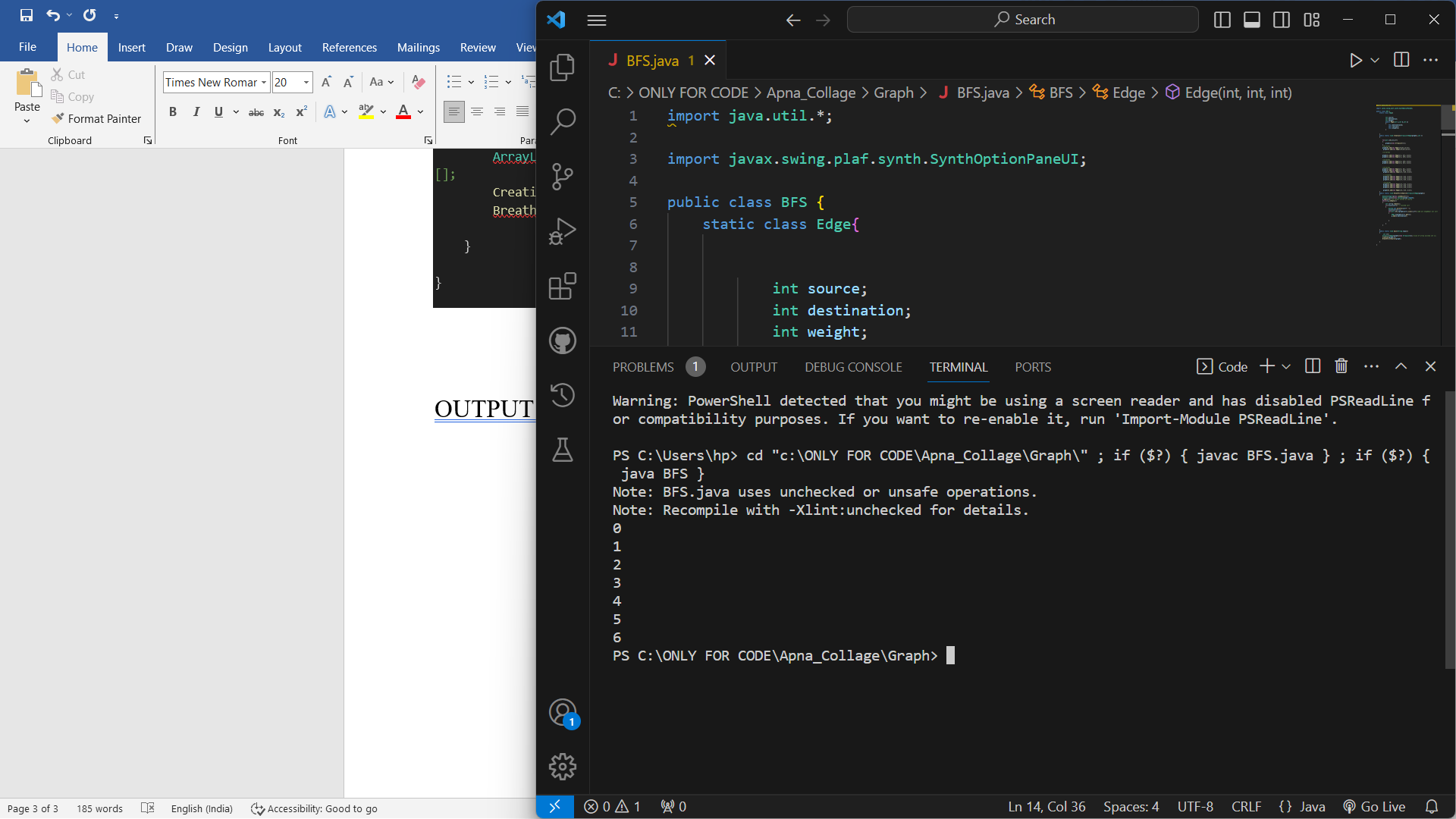
        ArrayList<Edge>graph[]=new ArrayList[v];//size of array a[]=new int [];

        Creation(graph,v);

        BreathfirstSearch(graph);

    }

}

OUTPUT :

DFS

import java.util.\*;

import java.util.ArrayList;

public class DFS {

     static class Edge{

            int source;

            int destination;

            int weight;

            public Edge(int s,int d,int w)

            {

                this.destination=d;

                this.source=s;

                this.weight=w;

            }

    }

    public static void Creation(ArrayList<Edge>graph[],int v)

    {

        for(int i=0;i<v;i++)

        {

            graph[i]=new ArrayList<>();

        }

        //0-vertex

        graph[0].add(new Edge(0,1,1));

         graph[0].add(new Edge(0,2,1));

        //1-vertex

        graph[1].add(new Edge(1, 3, 1));

        graph[1].add(new Edge(1, 0, 1));

        //2-vertex

        graph[2].add(new Edge(2, 0, 1));

        graph[2].add(new Edge(2, 4, 1));

        //3-vertex

        graph[3].add(new Edge(3, 1, 1));

        graph[3].add(new Edge(3, 4, 1));

         graph[3].add(new Edge(3, 5, 1));

         // 4-vertex

         graph[4].add(new Edge(4, 2, 1));

         graph[4].add(new Edge(4, 3, 1));

         graph[4].add(new Edge(4, 5, 1));

         //5-vertex

         graph[5].add(new Edge(5, 3, 1));

         graph[5].add(new Edge(5, 4, 1));

         graph[5].add(new Edge(5, 6, 1));

         graph[6].add(new Edge(6, 5, 1));

    }

    public static void Depthfirstsearch(ArrayList<Edge>[]graph,int curr, boolean visit[])

    {

       //visit

       System.out.println(curr+" ");

       visit[curr]=true;

       for(int i=0;i<graph[curr].size();i++)//neighbour ke liye

       {

        Edge e=graph[curr].get(i);

        if(!visit[e.destination])

        {

            Depthfirstsearch(graph,e.destination,visit);

        }

       }

    }

    public static void main(String[] args) {

        int v=7;

        ArrayList<Edge>graph[]=new ArrayList[v];

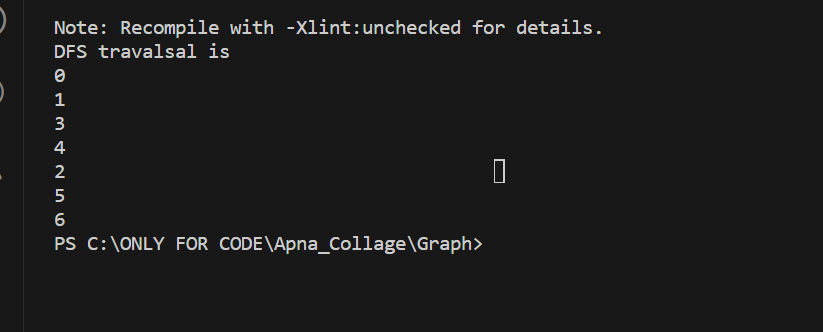
         Creation(graph,v);

         System.out.println("DFS travalsal is");

        Depthfirstsearch(graph, 0, new boolean[v]);

    }

}

Output: