1 /\*\*  
 2 \* @author Amar Bessedik   
 3 \* This program user executes Kruskal's algorithm on 3 different graphs   
 4 \* and computes their Minimum Spanning Trees.   
 5 \* CSC560 - Kruskal Project.  
 6 \*/  
 7 public class KruskalUser  
 8 {  
 9 static Edge[] E;//Hold a graph's edges  
10 static int V, i;//"V" is # of vertices, "i" is a counter of titles.  
11   
12 public static void main(String[] args)  
13 {  
14 final String[] Title =  
15 {  
16 "1ST GRAPH: ", "2ND GRAPH: ", "3RD GRAPH: "  
17 };  
18   
19 final String path = "C:\\Users\\Amar-cs\\Desktop\\CSC560\\Kruskals\_project\\";

20 Graph G1 = new Graph(path + "data1");//Graph 1.  
21 Graph G2 = new Graph(path + "data2");//Graph 2.  
22 Graph G3 = new Graph(path + "data3");//Graph 3.  
23   
24   
25 Graph[] graphs = {G1, G2, G3};//Array of graphs.  
26   
27 Kruskal k;//Kuskal instance.  
28   
29 for (Graph G : graphs)  
30 {  
31 k = new Kruskal(G);//Kruskal's object.  
32 System.out.println(Title[i++]);//print graph's title.  
33   
34 get\_params(G);//extract edges and # of vertices from the graph.  
35   
36 k.kruskal(V, E);//Run Kruskal's Algorithm on G's parameters.  
37 }//end for  
38 }//end main  
39   
40 private static void get\_params(Graph G)  
41 {  
42 E = G.getEdges();// get edges of G.  
43 V = G.getVertices();//get # of vertices of G.  
44 }//end get\_params  
45 }//end main class  
46