**package** BankMid;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Scanner;

**class** EdgeWeight **implements** Comparable<EdgeWeight>

{

**public int node1**;

**public int node2**;

**public int weight**;

**public int getParent**;

**public** EdgeWeight(**int** node1, **int** node2, **int** weight) {

**this**.**node1** = node1;

**this**.**node2** = node2;

**this**.**weight** = weight;

**this**.**getParent** =0;

}

@Override

**public int** compareTo(EdgeWeight o) {

**if** (**this**.**weight**>o.**weight**) **return** 1;

**else**

**return** -1;

}

}

**class** Find

{

**public int**[] **parent**;

**public int**[] **rank**;

**public int getParent**;

**public** ArrayList<EdgeWeight> **graph**;

**public** Find(**int**[] parent, **int**[] rank, ArrayList<EdgeWeight> graph) {

**this**.**parent** = parent;

**this**.**rank** = rank;

**this**.**graph** = graph;

**this**.**getParent** =0;

}

**public void** print()

{

**for** (EdgeWeight print : **graph**)

{

unionSet(print.**node1**,print.**node2**,print.**weight**);

}

}

**public void** makeSet(**int** length)

{

**for** (**int** i=0;i<length;i++)

{

**parent**[i] = i;

**rank**[i] = 0;

}

}

**public int** findParent(**int** value)

{

**if** (value != **parent**[value])

{

**return parent**[value] = findParent(**parent**[value]);

}**else**

**return** value;

}

**public void** unionSet(**int** node1,**int** node2,**int** weight)

{

**int** parentNode1 = findParent(node1);

**int** parentNode2 = findParent(node2);

**if** (parentNode1 != parentNode2)

{

System.***out***.println(node1+**" "**+node2+**" "**+weight);

}

**if** (**rank**[parentNode1] < **rank**[parentNode2])

{

**parent**[parentNode1] = parentNode2;

}**else if**(**rank**[parentNode1] > **rank**[parentNode2])

{

**parent**[parentNode2] = parentNode1;

}**else**

{

**parent**[parentNode2] = parentNode1;

**rank**[parentNode1] ++;

}

}

}

**public class** KruskalAlgo {

**public static void** main(String[] args) {

ArrayList<EdgeWeight> graph = **new** ArrayList<>();

Scanner input = **new** Scanner(System.***in***);

System.***out***.println(**"Enter number of vertex: "**);

**int** vertex = input.nextInt();

System.***out***.println(**"Enter number of edge: "**);

**int** edge = input.nextInt();

**for** (**int** i=0;i<edge;i++)

{

**int** n1 = input.nextInt();

**int** n2 = input.nextInt();

**int** weight = input.nextInt();

graph.add(**new** EdgeWeight(n1,n2,weight));

}

*/\*graph.add(new EdgeWeight(0, 1, 10));*

*graph.add(new EdgeWeight(0, 2, 6));*

*graph.add(new EdgeWeight(0, 3, 5));*

*graph.add(new EdgeWeight(1, 3, 15));*

*graph.add(new EdgeWeight(2, 3, 4));*

*graph.add(new EdgeWeight(2, 4, 1));\*/*

Collections.*sort*(graph);

**int**[] parent = **new int**[100];

**int**[] rank = **new int**[100];

Find object = **new** Find(parent,rank,graph);

object.makeSet(graph.size());

System.***out***.println(**"After Kruskal is done: "**);

object.print();

}

}