1. Create a method that takes an ArrayList of Integer and returns a sorted copy of that ArrayList with no duplicates.

Sample Input: {5, 7, 4, 6, 5, 6, 9, 7}

Sample Output: {4, 5, 6, 7, 9}

import java.util.ArrayList;  
public class Remove\_Duplicate\_and\_sort\_list {  
public static ArrayList<Integer> sort\_directlt(ArrayList<Integer> list){  
   
 Collections.sort(list);  
   
 return list;  
 }  
}

2. Determine the output:

14

MMM[-1, 7, 8]

3.

a. Create a class named Point3D that contains 3 instance variables x, y, and z.

b. Create a constructor that sets the variables. Also, create get and set methods for each variable.

c. Create a toString() method.

d. Make Point3D implement Comparable<Point3D>.

Also, create a compareTo(Point3D other) method that compares based on

the x-coordinate, then y-coordinate for tiebreakers, then z-coordinate for tiebreakers.

For example, (1, 2, 5) comes before (2, 1, 4), which comes before (2, 2, 3), which comes before (2, 2, 4).

e. Create a TreeSet of Point3D and insert a few elements including duplicates.

Verify that the TreeSet removes all the duplicates.

Also verify that the TreeSet orders the elements properly.

import java.util.Iterator;  
import java.util.TreeSet;  
class Point3D implements Comparable<Point3D>  
{  
 int x, y, z;  
   
 public Point3D(int x, int y, int z)  
 {  
 this.x = x;  
 this.y = y;  
 this.z = z;  
 }  
 public int compareTo(Point3D obj)  
 {  
 if(x == obj.x)  
 {  
 if(y == obj.y)   
 {  
 if(z == obj.z)  
 {  
 return 0;  
 }  
 else if(z > obj.z)  
 {  
 return 1;  
 }  
 else  
 {  
 return -1;  
 }  
 }  
 else if(y > obj.y)  
 {  
 return 1;  
 }  
 else  
 {  
 return -1;  
 }  
 }  
 else if(x > obj.x)   
 {  
 return 1;  
 }  
 else  
 {  
 return -1;  
 }  
 }  
   
 public String toString()  
 {  
 return "(" + x + ", " + y + ", " + z + ")";  
 }  
}  
  
public class testTreeset   
{  
 public static void main(String[] args)   
 {  
 TreeSet<Point3D> ts = new TreeSet<>();   
 ts.add(new Point3D(1, 2, 5));  
 ts.add(new Point3D(2, 2, 3));  
 ts.add(new Point3D(1, 2, 5));  
 ts.add(new Point3D(2, 2, 4));  
 ts.add(new Point3D(2, 1, 4));  
   
 Iterator<Point3D> itr = ts.iterator();   
 while(itr.hasNext())  
 {   
 System.out.println(itr.next());   
 }  
  
 }  
}M

4. Determine the output:

05679  
MMM57906