Name: S. Santhosh kumar.....
Ph no: 7305118197.....

//Question 1:.....

```
import java.util.LinkedList;
public class Exercise {
	public static void main(String[] args) {
		
		LinkedList <String> l_list = new LinkedList <String> ();
		
		l_list.add("Red");
		l_list.add("Green");
		l_list.add("Black");
		l_list.add("White");
		l_list.add("Pink");
		System.out.println("Original linked list: ");
		System.out.println("Let add the Yellow color after the Red Color: " + l_list);
		l_list.add(1, "Yellow");
		
		System.out.println("The linked list:" + l_list);
	}
}
```

//Output:.....

```
Original linked list:
Let add the Yellow color after the Red Color: [Red, Green, Black, White
, Pink]
The linked list:[Red, Yellow, Green, Black, White, Pink]
```

//Question 2:.....

```
import java.util.ArrayList;

public class ArraylistTest {
   public static void main(String[] args)
   {

      ArrayList<Integer> arr = new ArrayList<Integer>(10);

 arr.add(1);

      ans = arr.isEmpty();
      if (ans == true)
         System.out.println("The ArrayList is empty");
      else
         System.out.println("The ArrayList is not empty");
   }
}
```

//Output:.....

```
The ArrayList is not empty
```

//Question 3:.....

```
import java.util.HashSet;
```

```
import java.util.List;
import java.util.ArrayList;
class ConvertHashSetToArrayList{
  public static void main(String[] args) {

    HashSet<String> hset = new HashSet<String>();

    hset.add("Steve");
    hset.add("Matt");
    hset.add("Govinda");
    hset.add("John");
    hset.add("Tommy");


    System.out.println("HashSet contains: "+ hset);


    List<String> list = new ArrayList<String>(hset);


    System.out.println("ArrayList contains: "+ list);
  }
}
//Output:.....

HashSet contains: [Tommy, Matt, Steve, Govinda, John]
ArrayList contains: [Tommy, Matt, Steve, Govinda, John]

//Question 4:.....


import java.util.*;
class Sortmap {

 
static Map<String, Integer> map = new HashMap<>();

public static void sortbykey()
{
  ArrayList<String> sortedKeys
    = new ArrayList<String>(map.keySet());

  Collections.sort(sortedKeys);

  
  for (String x : sortedKeys)
    System.out.println("Key = " + x
              + ", Value = " + map.get(x));
}

 
public static void main(String args[])
{
  
  map.put("Jayant", 80);
  map.put("Abhishek", 90);
  map.put("Anushka", 80);
  map.put("Amit", 75);
  map.put("Danish", 40);
```

□□
□□sortbykey();
□}
}

//Output:.....

Key = Abhishek, Value = 90
Key = Amit, Value = 75
Key = Anushka, Value = 80
Key = Danish, Value = 40
Key = Jayant, Value = 80

//Question 5:.....

```
import java.util.TreeMap;
import java.util.Set;
import java.util.Map;
import java.util.Iterator;
public class TreeMapExample {

  public static void main(String[] args) {


    TreeMap<String, String> treemap = new TreeMap<String, String>();


    treemap.put("Key1","Item1");
    treemap.put("Key2","Item2");
    treemap.put("Key3","Item3");
    treemap.put("Key4","Item4");
    treemap.put("Key5","Item5");

    Set set = treemap.entrySet();

    Iterator it = set.iterator();

    while(it.hasNext()) {
      Map.Entry me = (Map.Entry)it.next();
      System.out.print("Key is: "+me.getKey() + " & ");
      System.out.println("Value is: "+me.getValue());
    }
  }
}
```

//Output:.....

Key is: Key1 & Value is: Item1
Key is: Key2 & Value is: Item2
Key is: Key3 & Value is: Item3
Key is: Key4 & Value is: Item4
Key is: Key5 & Value is: Item5

//Question 6:.....

```
import java.util.*;
```

//using comparable interface:.....

```
class Student implements Comparable<Student>{
int rollno;
String name;
int age;
Student(int rollno,String name,int age){
this.rollno=rollno;
this.name=name;
this.age=age;
}

public int compareTo(Student st){
if(age==st.age)
return 0;
else if(age>st.age)
return 1;
else
return -1;
}
}

import java.util.*;

public class TestSort1{
public static void main(String args[]){
ArrayList<Student> al=new ArrayList<Student>();
al.add(new Student(101,"Vijay",23));
al.add(new Student(106,"Ajay",27));
al.add(new Student(105,"Jai",21));

Collections.sort(al);
for(Student st:al){
System.out.println(st.rollno+" "+st.name+" "+st.age);
}
}
}

//Output:.....

105 Jai 21
101 Vijay 23
106 Ajay 27

//using comparator interface:.....

class Student{
int rollno;
String name;
int age;
Student(int rollno,String name,int age){
this.rollno=rollno;
this.name=name;
this.age=age;
}
}

class AgeComparator implements Comparator{
public int compare(Object o1,Object o2){
Student s1=(Student)o1;
Student s2=(Student)o2;
```

```
if(s1.age==s2.age)
return 0;
else if(s1.age>s2.age)
return 1;
else
return -1;
}
}
class Simple{
public static void main(String args[]){

ArrayList al=new ArrayList();
al.add(new Student(101,"Vijay",23));
al.add(new Student(106,"Ajay",27));
al.add(new Student(105,"Jai",21));
System.out.println("Sorting by age");

Collections.sort(al,new AgeComparator());
Iterator itr2=al.iterator();
while(itr2.hasNext()){
Student st=(Student)itr2.next();
System.out.println(st.rollno+" "+st.name+" "+st.age);
}


}
}

//Output:.....
 Sorting by age
      105 Jai 21
      101 Vijay 23
      106 Ajay 27
```