Q.1 Write a program to create a new tree set, add some colors (string) and print out the tree set.

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


public class A1_SetA_Pg2

{

    public static void main(String args[])

    {

      int n;
      String friendName;

      Scanner sc = new Scanner(System.in);

      LinkedList<String> list = new LinkedList<String>();

      System.out.print("Enter Number of friends : ");
      n = sc.nextInt();


      for(int i=0;i<n;i++)

      {

            System.out.print("Enter Friend Number - "+(i+1)+" : ");
            friendName = sc.next();

            list.add(friendName);

      }


      System.out.println("----- Your Friend List ----");

      Iterator<String> itr = list.iterator();

      while(itr.hasNext())

      {

            System.out.println(itr.next());
      }

    }

}
```

Q.2) Create the hash table that will maintain the mobile number and student name. Display the contact list.

```
import java.util.*;
import java.util.Set;
import java.util.TreeSet;

public class hashTable

{


    public static void main(String args[])

    {


            Hashtable<String,String> ht =new Hashtable<String,String>();

            ht.put("Sachin Tendulkar","7822902144");
            ht.put("Rahul Dravid","8933902144");
            ht.put("Mithali Raj","922902144");
            ht.put("Smrithi Mandhana","982502144");

            System.out.println("\n---- All the Elements in List----");
            // print hashTable Elements one by one
            Iterator itr = ht.entrySet().iterator();


            while(itr.hasNext())

            {

                System.out.println(itr.next());
            }


            //print only mobile numbers from HashTable

            Iterator itr1 = ht.values().iterator();

            System.out.println("\n\n---- Contact List ----");

            while(itr1.hasNext())

            {

                System.out.println(itr1.next());
            }
            //print only names of students from HashTable

            Iterator itr2 = ht.keySet().iterator();

            System.out.println("\n\n---- Student Name List ----");

            while(itr2.hasNext())

            {

                  System.out.println(itr2.next());
            }


            System.out.println("\n\nTotal Number of Elements in HashTable are : "+ht.size());


            Set<String> ts = new TreeSet<String>();

            System.out.println(ts);


    }


}
```

Q.3) Q.1 a) Accept ‘n’ integers from the user. Store and display integers in sorted order having proper collection class. The collection should not accept duplicate elements.

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


public class A1_SetB_Pg1

{


    public static void main(String args[])

    {
      int n,inputVal;


      Scanner sc =  new Scanner(System.in);

      Set<Integer> ts =  new TreeSet<Integer>();
```

```
            System.out.print("Enter number of Integers : ");

            n = sc.nextInt();

            System.out.print("Enter number of Integers : ");

            for(int i=0; i<n; i++)

            {

                    System.out.print("Enter Number " + (i+1)+" : ");

                    inputVal = sc.nextInt();

                    ts.add(inputVal);

            }

            Iterator<Integer> itr = ts.iterator();

            while(itr.hasNext())

            {

                    System.out.println(itr.next());

            }
          }

}
```

Q.4) Write a program in which thread sleep for 6 sec in the loop in reverse order from 100 to 1 and change the name of thread.

```
public class assign3Q2 extends Thread

{


        public void run()
        {

            for(int i=100;i>0;i--)

            {

                    System.out.println(i);

                    try{

                            Thread.sleep(6000);
            }
            catch(Exception e)
            {

                  System.out.println(e);
            }
      }

    }



    public static void main(String args[])

    {


      assign3Q2 t1 = new assign3Q2();

      t1.start();
    }


}
```

Q.5) Program to define a thread for printing text on output screen for ‘n’ number of times. Create 3 threads and run them. Pass the text ‘n’ parameters to the thread constructor.
Example:
i. First thread prints “COVID19” 10 times.
ii. Second thread prints “LOCKDOWN2020” 20 times .
iii. Third thread prints “VACCINATED2021” 30 times .

```
public class assign2 extends Thread{

    String msg = " ";
    int n;
    assign2(String msg,int n)
    {
      this.msg = msg;
      this.n = n;
    }

    public void run()
    {
      for(int i=1;i<=n;i++)
      {
            System.out.println(msg + " " + i + " times");

            try {
```

```
                        Thread.sleep(1000);
                }
                catch(Exception e)
                {
                        System.out.println(e);
                }
            }
        }

        public static void main(String args[])
        {
           int n = Integer.parseInt(args[0]);

           assign2 t1 = new assign2("Covid-19",n);

           t1.start();

           assign2 t2 = new assign2("Lockdown 2020",n+10);

           t2.start();

           assign2 t3 = new assign2("Vaccinated 2021",n+20);

           t3.start();


        }

}
```

Q.6) Write a program to create a new tree set, add some colors (string) and print out the tree set.

```
import java.util.*;
import java.util.Set;
import java.util.TreeSet;



public class treeSetPg

{

        public static void main(String args[])

        {

           Set<String> t = new TreeSet<String>();

           t.add("Red");
           t.add("Blue");
```

```
            t.add("Green");
            t.add("Orange");
            t.add("Purple");
            t.add("Yellow");
            t.add("Cyan");

            System.out.println("---- Colors You Added ----");

            Iterator<String> itr = t.iterator();

            while(itr.hasNext())

            {

                        System.out.println(itr.next());

            }
        }
}
```

Q.7) Write a Java program to accept the details of Employee (EmpNo, EmpName, Salary) and display it.

```
import java.sql.*;
import java.util.Scanner;


public class empData {

        public static void main(String args[])

        {

            Connection con = null;
            Statement stm = null;
            ResultSet rs = null;

            Scanner sc = new Scanner(System.in);

            int empno,salary,i;
            String empname;


            try {

                        Class.forName("com.mysql.cj.jdbc.Driver");

                        con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb","root","Password123#@!");
```

```
        if(con == null)

        {

            System.out.println("Connection Failed!");

        }

        else

        {
            System.out.println("===== Connection Sucessful =====");

            System.out.print("Enter Employee Number : ");
            empno = sc.nextInt();

            System.out.print("Enter Employee Name : ");
            empname = sc.next();

            System.out.print("Enter Employee Salary : ");
            salary = sc.nextInt();
```

Q.7) Write a Java program to accept the details of Employee (EmpNo, EmpName, Salary) and display it.

```
            stm = con.createStatement();

            i = stm.executeUpdate("INSERT INTO emp 
values("+empno+",'"+empname+"',"+salary+")");

            System.out.println("====== Employee Table Record ====");

            rs = stm.executeQuery("Select * from emp");
```

Q.7) Write a Java program to accept the details of Employee (EmpNo, EmpName, Salary) and display it.

```
            while(rs.next())
            {

                System.out.println("Employee Id : "+rs.getInt(1)+"\tEmployee Name : 
"+rs.getString(2)+"\tEmployee Salary : "+rs.getInt(3));

            }

        }

    }
    catch(Exception e)
    {
        System.out.println(e);
    }
    }

}
```