Date: Write a program to demonstrate Collection Interface List Interface (ArrayList, LinkedList, Vector).

Exp: 19a

program

```
import java.util.*;
import java.io.*;
class LstsDemo{
    public static void main(String[] args){
        List<Integer> al = new ArrayList<>();
        List<Integer> ll = new LinkedList<>();
        List<Integer> v = new Vector<Integer>();

        for(int i=1; i<=10; i++){
            al.add(i);
            ll.add(i*i);
            v.add(2*i);
        }

        System.out.println("ArrayList: "+ al +"\n"+ "LinkedList: "+ ll
                           +"\n"+ "Vector: "+ v);

    }
}
```

Output:

ArrayList : [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
LinkedList : [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
Vector : [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

Date: write a program to demonstrate Collection interface : Set interface.

Exp: 19b

program:

```
import java.util.*;
import java.lang.io.*;

public class SetDemo {
    public static void main (String [] args) {
        Set <Integer> hs = new HashSet<>();
        Set <Integer> lhs = new LinkedHashSet<>();
        Set <Integer> ts = new TreeSet<>();

        for (int i=0; i<=10; i++) {
            hs.add(i);
        }
        for (int i=0; i<=5; i=i+2) {
            lhs.add(-1);
        }
        if (lhs.contains(-4)) {
            lhs.add(-5);
        }
        ts.add(hs);
        ts.add(lhs);
        System.out.println("HashSet: " + hs + "\n" +
                           "LinkedHashSet: " + lhs + "\n" +
                           "TreeSet: " + ts + "\n");
    }
}
```

Output:

HashSet: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

LinkedHashSet: [0, -2, -4, -5]

TreeSet : [-5, -4, -2, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Date:

Exp: 19dc

Write a program to demonstrate Collection Interface Queue Interface.

program

```
import java.util.*
class QueueDemo{
    public static void main(String[] args){
        BlockingQueue<String> bq = new ArrayBlockingQueue<String>();
        bq.put("Star Wars");
        bq.put("Super Man");
        bq.put("Flash")
        bq.remove();
        System.out.println(bq);
        PriorityQueue<Integer> pq = new PriorityQueue<>();
        pq.add(10);
        pq.add(-10);
        pq.add(5);
        pq.add(3);
        System.out.println("smallest element: "+ pq.peek());
    }
}
```

Output:

[SuperMan, Flash]

~~tes~~ Smallest Element: -10

Date: Write a program to demonstrate the use of HashMap.

Exp: 20a

```
import java.util.HashMap;
public class MapDemo {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<>();
        map.put("Eren", 10);
        map.put("Levi", 12);
        map.put("Erwin", 100);

        System.out.println("Size of map: "+map.size());
        System.out.println(map);
        if (map.containsKey("Levi")){
            System.out.println("The value for Key "Levi" is "+map.get("Levi"));
        }
    }
}
```

Output:

Size of map = 3

{Eren = 10, Levi = 11, Erwin = 100}

value for key Levi is 11

Date:

Exp: 206

Write a program to demonstrate the use of Map interface classes : Hash table.

program

```
import java.io.*;
import java.util.*;
class AddElementsToHashtabe {
    public static void main(String args[]) {
        Hashtable<Integer, String> ht1 = new Hashtable<>();
        Hashtable<Integer, String> ht2 = new Hashtable<>();

        ht1.put(1, "one");
        ht1.put(2, "two");
        ht1.put(3, "three");
        ht2.put(4, "four");
        ht2.put(5, "five");
        ht2.put(6, "six");

        System.out.println("Mappings of ht1: " + ht1);
        System.out.println("Mappings of ht2: " + ht2);

    }
}
```

Output:

Mappings of HT1 : {3 = three, 2 = two, 1 = one}

Mappings of HT2 : {6 = six, 5 = five, 4 = four}

Date: Write a program to demonstrate the usea of TreeMap

Exp: 20c

```
class TreeMapDemo {
public static void main (String[] args) {
        TreeMap<Integer, String> tm = new TreeMap<>();

        tm.put (10, "Geeks");
        tm.put (15, "IronMan");
        tm.put (16, "SpiderWoman");
        tm.put (36, "Green");
        tm.put (1, "you");

        Iterator<Integer, String>
        for (Map.Entry <String, Integer> entry : tm.entrySet()) {
         System.out.println ("Key: "+tm.getKey() + "value: "+tm.getValue());
        }
        }
}
```

Output:

```
Key : 1     Value = You
Key : 10    value = Geeks
Key : 15    value = Iron Man
Key : 36    Value = Green
Key : 160   value = Spider Woman
```

Date :

20 d

Write a program to demonstrate the use of Map Interface : Properties.

program

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

public class PropertiesDemo {
    public static void main(String[] args) {
        FileReader reader = new FileReader("db.properties");
        Properties p = new Properties();
        p.load(reader);
        System.out.println(p.getProperty("username"));
        System.out.println(p.getProperty("password"));
    }
}
```

input file: db.properties

```
username = coder
password = I like cats.
```

Output

```
Coder
I like cats.
```