LAB-4

1.Min and Max in a List in Java

Sol:

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class MinMax {
    public static void main(String[] args) {
        List<Integer> numbers = new ArrayList<>();
        numbers.add(5);
        numbers.add(8);
        numbers.add(3);
        numbers.add(12);
        numbers.add(7);
        int minValue = Collections.min(numbers);
        int maxValue = Collections.max(numbers);

        System.out.println("List: " + numbers);
        System.out.println("Minimum value: " + minValue);
        System.out.println("Maximum value: " + maxValue);
    }
}
```

o/p

```
List: [5, 8, 3, 12, 7]
Minimum value: 3
Maximum value: 12
```

2. Split a List into Two Halves in Java

Sol:

```
import java.util.ArrayList;
import java.util.List;

public class SplitList {
    public static void main(String[] args) {
        List<Integer> numbers = new ArrayList<>();
        numbers.add(1);
        numbers.add(2);
        numbers.add(3);
        numbers.add(4);
        numbers.add(5);
        numbers.add(6);

        int middle = numbers.size() / 2;
        List<Integer> firstHalf = new ArrayList<>(numbers.subList(0, middle));
        List<Integer> secondHalf = new ArrayList<>(numbers.subList(middle, numbers.size()));

        System.out.println("Original List: " + numbers);
        System.out.println("First Half: " + firstHalf);
        System.out.println("Second Half: " + secondHalf);
    }
}
```

o/p

```
Original List: [1, 2, 3, 4, 5, 6]
First Half: [1, 2, 3]
Second Half: [4, 5, 6]
```

3. Remove Duplicates from ArrayList in Java

Sol:

```
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;

public class RemoveDup {
    public static void main(String[] args) {
        List<Integer> numbersWithDuplicates = new ArrayList<>();
        numbersWithDuplicates.add(1);
        numbersWithDuplicates.add(2);
        numbersWithDuplicates.add(3);
        numbersWithDuplicates.add(2);
        numbersWithDuplicates.add(4);
        numbersWithDuplicates.add(3);

        HashSet<Integer> uniqueNumbersSet = new HashSet<>(numbersWithDuplicates);
        List<Integer> numbersWithoutDuplicates = new ArrayList<>(uniqueNumbersSet);

        System.out.println("Original List with Duplicates: " + numbersWithDuplicates);
        System.out.println("List without Duplicates: " + numbersWithoutDuplicates);
    }
}
```

o/p

```
Original List with Duplicates: [1, 2, 3, 2, 4, 3]
List without Duplicates: [1, 2, 3, 4]
```

4.Add Element at First and Last Position of LinkedList in Java

Sol:

```
import java.util.LinkedList;

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

      LinkedList<String> linkedList = new LinkedList<>();
      linkedList.add("Apple");
      linkedList.add("Banana");
      linkedList.add("Orange");

      linkedList.addFirst("Grapes");
      linkedList.addLast("Pineapple");

      System.out.println("Final LinkedList: " + linkedList);
   }
}
```

o/p

```
Final LinkedList: [Grapes, Apple, Banana, Orange, Pineapple]
```