Java Assignment-4

1. ArrayList using all collections:

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

public class Arraylist {
	public static void main(String[] args)
	{
		ArrayList<String> list=new ArrayList<String>();
		list.add("Icecreams");
		list.add("Chocolates");
		list.add("Bytes");
		list.add("LittleHearts");
		System.out.println(list);
		//get
		list.get(2);
		System.out.println("Access an Item: "+list);
		//set
	 list.set(3, "Shawerma");
	 System.out.println("Setting the change: "+list);
	 //size
	 list.size();
		System.out.println("Size of Array"+list);
		 //sort
		Collections.sort(list);
		System.out.println("Sorting the List: "+list);
		//NaturalOrder
		list.sort(Comparator.naturalOrder());
		System.out.println("NaturalOrder: "+list);
		 //Reverse Order
		list.sort(Comparator.reverseOrder());
		System.out.println("ReverseOrder: "+list);
		//Index
		int pos =list.indexOf("Bytes");
		System.out.println("Indexof Specified Element: "+pos);
	   String arr[] = new String[list.size()];
	   //toArray
	   arr=list.toArray(arr);
	   System.out.print("Array: ");
	   //toString
	 String S = list.toString();
	 System.out.println("String: " + S);
	 //ensureCapacity
	 list.ensureCapacity(10);
	   //lastIndexOf
	 int element = list.lastIndexOf("Icecreams");
	   System.out.println("the lastIndexof of" + " Icecream is " + element);
	   //clone
	   ArrayList list2=new ArrayList();
		list2 =(ArrayList)list.clone();
		System.out.println("After using Clone: "+list2);
```

```
            list2.add("Sandwich");
            list2.add("pizza");
        //retainAll
            list2.retainAll(list);
        System.out.println("\nAfter Applying retainAll()"+" method to
list2\n"+list2);
        System.out.println("\nAfter Applying retainAll()"+" method to
list\n"+list);
        //containsAll
        boolean ca = list.containsAll(list2);
        System.out.println("ArrayList 1 contains all elements of ArrayList 2: " +
ca);
            //isEmpty
        boolean res=list.isEmpty();
            System.out.println("Checking EMpty Status: "+res);
            //contains
            System.out.println("Using Contains: "+list.contains("Paneer"));
            System.out.println(list.contains("Bytes"));
            //Iterator
            for(int i=0;i<list.size();i++)
            {
                  System.out.println("Using for loop: "+list.get(i));
            }
              list.retainAll(list2);
              System.out.println("iterating the elements after retaining the
elements of list2");
               //remove
             list2.remove(2);
             //clear
             list2.clear();
              System.out.println("After Remove: "+ list2);
      }
}
```

Output:

```
[Icecreams, Chocolates, Bytes, LittleHearts]
Access an Item: [Icecreams, Chocolates, Bytes, LittleHearts]
Setting the change: [Icecreams, Chocolates, Bytes, Shawerma]
Size of Array[Icecreams, Chocolates, Bytes, Shawerma]
Sorting the List: [Bytes, Chocolates, Icecreams, Shawerma]
NaturalOrder: [Bytes, Chocolates, Icecreams, Shawerma]
ReverseOrder: [Shawerma, Icecreams, Chocolates, Bytes]
Indexof Specified Element: 3
Array: String: [Shawerma, Icecreams, Chocolates, Bytes]
the lastIndexof of Icecream is 1
After using Clone: [Shawerma, Icecreams, Chocolates, Bytes]

After Applying retainAll() method to list2
[Shawerma, Icecreams, Chocolates, Bytes]

After Applying retainAll() method to list
[Shawerma, Icecreams, Chocolates, Bytes]
ArrayList 1 contains all elements of ArrayList 2: true
Checking EMpty Status: false
Using Contains: false
true
Using for loop: Shawerma
Using for loop: Icecreams
```

```
Using for loop: Chocolates
Using for loop: Bytes
iterating the elements after retaining the elements of list2
After Remove: []
```

2.LinkedList using all its Collections.

```
import java.util.*;
public class Linkedlist {

	public static void main(String[] args) {

		LinkedList<String> ll=new LinkedList<>();
		ll.add("After");
		ll.add("Vampire Dairies");
		ll.add("MoneyHeist");
		ll.add("FiveFeetApart");
		ll.add("You");
		ll.add("NoteBook");
		ll.add("Twilight");
		System.out.println("linKedlist is"+ll);
		//get
		String ws=ll.get(3);
		System.out.println("Series are"+ws);
		  //remove
		  ll.remove("NoteBook");
	   System.out.println("After invoking remove(object) method: "+ll);
	   ll.remove(0);
	   System.out.println("After invoking remove(index) method: "+ll);
	   LinkedList<String> ll1=new LinkedList<String>();
	   ll1.add("BrokenHeart");
	   ll1.add("Bridgerton");
	   //addAll
	   ll.addAll(ll1);
	   System.out.println("Updated list : "+ll);
	   //removeAll
	   ll.removeAll(ll1);
	   System.out.println("After invoking removeAll() method: "+ll);
	  //removeFirst
	   ll.removeFirst();
	   System.out.println("After invoking removeFirst() method: "+ll);
	  //removeLast
	   ll.removeLast();
	   System.out.println("After invoking removeLast() method: "+ll);
	   //removeFirstOccurence
	   ll.removeFirstOccurrence("Gaurav");
	   System.out.println("After invoking removeFirstOccurrence() method:
"+ll);
	  //removeLastOccurence
	   ll.removeLastOccurrence("Harsh");
	   System.out.println("After invoking removeLastOccurrence() method: "+ll);
	   //clear
	   ll.clear();
	   System.out.println("After invoking clear() method: "+ll);
	   //getFirst
	   System.out.println("The first element is: " + ll1.getFirst());
	   //getLast
	   System.out.println("The LAst element is: " + ll1.getLast());
```

```
        //addFirst
        ll1.addFirst("LAura");
        System.out.println("The add first element is: " + ll1);
        //addLast
        ll1.addLast("Paris");
        System.out.println("The add last element is: " + ll1);

  }


    }
```

Output:

```
linKedlist is[After, Vampire Dairies, MoneyHeist, FiveFeetApart, You, NoteBook,
Twilight]
Series areFiveFeetApart
After invoking remove(object) method: [After, Vampire Dairies, MoneyHeist,
FiveFeetApart, You, Twilight]
After invoking remove(index) method: [Vampire Dairies, MoneyHeist, FiveFeetApart,
You, Twilight]
Updated list : [Vampire Dairies, MoneyHeist, FiveFeetApart, You, Twilight,
BrokenHeart, Bridgerton]
After invoking removeAll() method: [Vampire Dairies, MoneyHeist, FiveFeetApart,
You, Twilight]
After invoking removeFirst() method: [MoneyHeist, FiveFeetApart, You, Twilight]
After invoking removeLast() method: [MoneyHeist, FiveFeetApart, You]
After invoking removeFirstOccurrence() method: [MoneyHeist, FiveFeetApart, You]
After invoking removeLastOccurrence() method: [MoneyHeist, FiveFeetApart, You]
After invoking clear() method: []
The first element is: BrokenHeart
The LAst element is: Bridgerton
The add first element is: [LAura, BrokenHeart, Bridgerton]
The add last element is: [LAura, BrokenHeart, Bridgerton, Paris]
```

3.PriorityQueue Implementation

```
import java.util.*;
public class PriorQ {

      public static void main(String[] args) {
            PriorityQueue<Integer> pq=new PriorityQueue<>();
            pq.add(46);
            pq.add(32);
            pq.add(42);
            pq.add(63);
            pq.add(1);
            pq.add(80);
            System.out.println(pq);
            pq.offer(6);
            System.out.println(pq);
       System.out.println("Poll Method  " + pq.poll());
       int element = pq.peek();
       System.out.println("Accessed Element: " + element);
       Iterator iterator = pq.iterator();
        while (iterator.hasNext()) {
           System.out.print(iterator.next() + " ");
```

```
        }
        boolean ret_val = pq.contains(6);
        System.out.println("\n\nPriority queue contains '6' "
                           + "or not?: " + ret_val);

        // get array equivalent of PriorityQueue with toArray ()
        Object[] numArr = pq.toArray();
        System.out.println("\nArray Contents: ");
        for (int i = 0; i < numArr.length; i++)
            System.out.print(numArr[i].toString() + " ");

    }
}
```

Output:

```
[1, 32, 42, 63, 46, 80]
[1, 32, 6, 63, 46, 80, 42]
Poll Method  1
Accessed Element: 6
6 32 42 63 46 80
Priority queue contains '6' or not?: true
Array Contents:
6 32 42 63 46 80
```