1. **Write a Java program to create a new array list, add some Movie names (string) and print out thecollection.**
   **-Write a Java program to insert an element into the array list at the first -position.**
   **-Write a Java program to retrieve an element (at a specified index) from a given array list.**
   **-Write a Java program to update specific array element by given element.**
   **-Write a Java program to remove the third element from a array list.**
   **-Write a Java program to search an element in a array list.**
   **-Write a Java program to sort a given array list.**
   **-Write a Java program to reverse elements in a array list.**
   **-Write a Java program to empty an array list.**

```
import java.util.*;

class CollectionsQ1{

    public static void main(String[] args){

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

        list.add("Saawariya");
        list.add("Barfi");
        list.add("Wake up Sid");
        list.add("Tamasha");
        list.add("Brahmastra");

        System.out.println("\nWrite a Java program to create a new array list, add some Movie names (string) and print out the collection.");
        System.out.println(list.size());
        System.out.println(list);

        System.out.println("\nWrite a Java program to insert an element into the array list at the first -position.");
        list.add(0, "Ae dil hai Mushkil");
        System.out.println(list.size());
        System.out.println(list);

        System.out.println("\nWrite a Java program to retrieve an element (at a specified index) from a given array list.");
```

```
        System.out.println(list.get(4));

        System.out.println("\nWrite a Java program to update specific array element by given element.");
        list.set(3, "Roy");
        System.out.println(list.size());
        System.out.println(list);

        System.out.println("\nWrite a Java program to remove the third element from a array list.");
        list.remove(2);
        System.out.println(list.size());
        System.out.println(list);

        System.out.println("\nWrite a Java program to search an element in a array list.");
        System.out.println(list.contains("Roy"));
        System.out.println(list.contains("Barfi"));

        System.out.println("\nWrite a Java program to sort a given array list.");
        Collections.sort(list);
        System.out.println(list);

        System.out.println("\nWrite a Java program to reverse elements in a array list.");
        Collections.sort(list, Collections.reverseOrder());
        System.out.println(list);

        System.out.println("\nWrite a Java program to empty an array list.");
        System.out.println(list.removeAll(list));
        System.out.println(list);
    }


}
```

**1. Write a Java program to append the specified element to the end of a linked list of names.**

**-Write a Java program to iterate through all elements in a linked list starting at the specified position.**

**-Write a Java program to iterate a linked list in reverse order.**

**-Write a Java program to insert the specified element at the specified position in the linked list.**

**-Write a Java program to insert elements into the linked list at the first and last position.**

**-Write a Java program to insert the specified element at the front of a linked list.**

**-Write a Java program to insert some elements at the specified position into a linked list.**

**-Write a Java program to get the first and last occurrence of the specified elements in a linked list.**

**-Write a Java program to remove first and last element from a linked list.**

**-Write a Java program of swap two elements in a linked list.**

**-Write a Java program to join two linked lists.**

**-Write a Java program to check if a particular element exists in a linked list.**

**-Write a Java program to convert a linked list to array list.**

**-Write a Java program to compare two linked lists.**

**-Write a Java program to test an linked list is empty or not.**

**-Write a Java program to replace an element in a linked list.**

```
import java.util.*;

class CollectionsQ2{

	public static void main(String[] args){

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

		list.add("Bediskar");
		list.add("Rajesh");
		list.add("Gopal");
		list.add("Himanshu");
		list.add("Kalyani");
		list.add("Nayan");
		list.add("Rohit");
		list.add("Tejas");
		list.add("Snehal");
		list.add("Siddhi");

		System.out.println("\nWrite a Java program to append the specified element to the end of a linked list of names.");
		list.addLast("Shrutika");
		System.out.println(list.size());
```

```
		System.out.println(list);


		System.out.println("\nWrite a Java program to iterate through all elements in a linked list starting at the specified position.");
		Iterator list1 = list.listIterator(0);
		while(list1.hasNext()){
			System.out.println(list1.next());
		}


		System.out.println("\nWrite a Java program to iterate a linked list in reverse order.");
		Iterator list2 = list.descendingIterator();
		System.out.println("\nReverse order:");
		while(list2.hasNext()){
			System.out.println(list2.next());
		}


		System.out.println("\nWrite a Java program to insert the specified element at the specified position in the linked list.");
		list.add(2, "Kalewad Sir");
		System.out.println(list);


		System.out.println("\nWrite a Java program to insert elements into the linked list at the first and last position.");
		list.addFirst("CDAC");
		list.addLast("Bye!!!");
		System.out.println(list);


		System.out.println("\nWrite a Java program to insert the specified element at the front of a linked list.");
		list.offerFirst("Hi!!!");
		System.out.println(list);


		System.out.println("\nWrite a Java program to insert some elements at the specified position into a linked list.");
		LinkedList<String> list3 = new LinkedList<>();//making list again we want to add not single but multiple elements
		list3.add("Mumbai");
		list3.add("Juhu");
```

```
        list.addAll(2, list3);//addAll to add everything in the list
        System.out.println(list);

        System.out.println("\nWrite a Java program to get the first and last occurrence of the specified elements in a linked list.");
        System.out.println(list.getFirst());
        System.out.println(list.getLast());

        System.out.println("\nWrite a Java program to remove first and last element from a linked list.");
        list.removeFirst();
        list.removeLast();
        System.out.println(list);

        System.out.println("\nWrite a Java program of swap two elements in a linked list.");
        Collections.swap(list, 0, 1);
        System.out.println(list);

        System.out.println("\nWrite a Java program to join two linked lists.");
        LinkedList<String> list4 = new LinkedList<>();
        list4.add("Bye!!!");
        list4.add("See you soon!!!");

        LinkedList<String> listjoin = new LinkedList<>();
        listjoin.addAll(list);
        listjoin.addAll(list4);
        System.out.println(listjoin);

        System.out.println("\nWrite a Java program to check if a particular element exists in a linked list.");
        System.out.println(list.contains("Shrutika"));//true
        System.out.println(list.contains("Hi!!!"));//false

        System.out.println("\nWrite a Java program to convert a linked list to array list.");

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

        for(String str : newlist){
                System.out.println(str);
```

```
            }

            System.out.println("\nWrite a Java program to compare two linked lists.");
            LinkedList<String> listcomp = new LinkedList<>();
            for (String comp : listjoin){
      listcomp.add(list.contains(comp) ? "Yes" : "No");
            }
        System.out.println(listcomp);

            System.out.println("\nWrite a Java program to test an linked list is empty or not.");
            System.out.println(list.isEmpty());//false
            list4.removeAll(list4);
            System.out.println(list4.isEmpty());//true

            System.out.println("\nWrite a Java program to replace an element in a linked list.");

            list.set(1, "Replace");
            System.out.println(list);

        }
}
```

1. **Write a Java program to append the specified element to the end of a hash set for Employee Id and Employee name.**
2. **-Write a Java program to get the number of elements in a hash set.**
3. **-Write a Java program to convert a hash set to an array.**
4. **-Write a Java program to convert a hash set to a tree set.**
5. **-Write a Java program to convert a hash set to a List/ArrayList.**
6. **-Write a Java program to remove all of the elements from a hash set.**

```
import java.util.*;

class Employee implements Comparable<Employee>{
        Integer employeeId;
        String name;

        Employee(Integer employeeId, String name){
            this.employeeId = employeeId;
            this.name = name;
        }

        @Override
        public String toString(){
            return this.employeeId+", "+this.name;
        }

        @Override
        public boolean equals(Object o){
            if(o instanceof Employee){
                    Employee temp = (Employee)o;
                    if(this.employeeId.equals(temp.employeeId)&&
                    this.name.equals(temp.name)){
                            return true;
                    }
            }
            return false;
```

```
	}

	@Override
	public int hashCode(){
		int prime = 17;
		prime = prime + prime*this.employeeId.hashCode();
		prime = prime + prime*this.name.hashCode();
		return prime;
	}

	@Override
	public int compareTo(Employee e){
		//return this.employeeId.compareTo(e.employeeId);
		return this.name.compareTo(e.name);

	}


}

class CollectionsQ3{
	public static void main(String[] args){

		Employee e1 = new Employee(104, "Aakash");
		Employee e2 = new Employee(101, "Sagar");

		HashSet<Employee> set = new HashSet<>();

		System.out.println("\nWrite a Java program to append the specified element to the end of a hash set for Employee Id and Employee name.");
		set.add(e1);
		set.add(e2);

		System.out.println(set);
```

```
        System.out.println("\nWrite a Java program to get the number of elements in a hash set.");
        System.out.println(set.size());

        System.out.println("\nWrite a Java program to convert a hash set to an array.");

        /* public Object[] toArray()
or
        public <T> T[] toArray(T[] a) */

        Employee arr[] = new Employee[set.size()];
        set.toArray(arr);
        for(Employee e : arr){
                System.out.println(e);
        }

        System.out.println("\nWrite a Java program to convert a hash set to a tree set");
        Set<Employee> treeset = new TreeSet<Employee>(set);
        for(Employee e : treeset){
                System.out.println(e);
        }
        System.out.println(treeset);

        System.out.println("\nWrite a Java program to convert a hash set to a List/ArrayList.");
        List<Employee> list = new ArrayList<Employee>(set);

        for(Employee e : list){
                System.out.println(e);
        }
        System.out.println(list);

        System.out.println("\nWrite a Java program to remove all of the elements from a hash set.");
        set.clear();

        System.out.println(set);
```

```
        }



}
```

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

   **-Write a Java program to iterate through all elements in a tree set.**

   **-Write a Java program to add all the elements of a specified tree set to another tree set.**

   **-Write a Java program to create a reverse order view of the elements contained in a given tree set.**

   **-Write a Java program to find the numbers less than 7 in a tree set.**

```
import java.util.*;

class CollectionsQ4{

    public static void main(String[] args){
        TreeSet<String> set = new TreeSet<String>();

        System.out.println("\nWrite a Java program to create a new tree set, add some fruits (string) and print out the tree set.");
        set.add("Apple");
        set.add("Pineapple");
        set.add("Orange");
        set.add("Banana");
        set.add("Chikku");
        set.add("Mango");

        System.out.println(set);

        System.out.println("\nWrite a Java program to iterate through all elements in a tree set.");
        for(String s : set){
            System.out.println(s);
        }

        System.out.println("\nWrite a Java program to add all the elements of a specified tree set to another tree set.");
        TreeSet<String> set1 = new TreeSet<String>();
        set1.add("Lime");
        set1.add("Strawberry");
        set1.add("Blue Berry");
        set.addAll(set1);
        System.out.println(set);
```

```
        System.out.println("\nWrite a Java program to create a reverse order view of the elements contained in a given tree set.");
        Iterator set2 = set.descendingIterator();
        while(set2.hasNext()){
            System.out.println(set2.next());
        }

        System.out.println("\nWrite a Java program to find the numbers less than 7 in a tree set.");

    }
}
```