```
1.public class insert {

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

  //add elements
  linkedList.add("k");
  linkedList.add("s");
  linkedList.add("h");

  //print linkedList
  System.out.println("LinkedList is: "+linkedList);

  //insert elements in first and last position
  linkedList.addFirst("A");
  linkedList.addLast("u");

  //print updated list
  System.out.println("list after adding last and last elements: "+linkedList);

 }

}
```

Output:

```
LinkedList is: [k, s, h]
list after adding last and last elements: [A, k, s, h, u]
```

```
2.public class Indexing {

 public static void main(String[] args) {
    LinkedList<String> linklist=new LinkedList<String>();
    linklist.add("Akshu");
    linklist.add("Shigvan");
    linklist.add("0221");
    System.out.println("linklist is: "+linklist);
    for(int i=0; i<linklist.size();i++) {
     System.out.println("position of elements: "+i+" "+linklist.get(i));
    }

 }

}
```

Output:

```
linklist is: [Akshu, Shigvan, 0221]
position of elements: 0 Akshu
position of elements: 1 Shigvan
position of elements: 2 0221
```

```
3.public class Displayonebyone {

 public static void main(String[] args) {
  LinkedList<String> linkedlist = new LinkedList<>();
  linkedlist.add("red");
  linkedlist.add("blue");
  linkedlist.add("green");
  linkedlist.add("white");

  System.out.println("The Linked list is: "+linkedlist);

  linkedlist.add(2,"redish");
  System.out.println("the list after: "+linkedlist);

  for(String m:linkedlist) {
   System.out.println(m);
  }

 }

}
```

Output:

```
The Linked list is: [red, blue, green, white]
the list after: [red, blue, redish, green, white]
red
blue
redish
green
white
```

```
4.public class Remove {
public static void main (String[] args) {

        // Creating an LinkedList
        LinkedList<String> list = new LinkedList<String>();

        //add elements in the list
        list.add("good");
        list.add("achieve");
        list.add("goals");
        list.add("2020");
        list.add("2021");

        // Displaying the list
        System.out.println("LinkedList:" + list);

        System.out.println("The last element is removed:" + list.removeLast());

        System.out.println("Final LinkedList: "+ list);

        System.out.println("The first element is removed:" + list.removeFirst());

        System.out.println("Final LinkedList:\t" + list);
 }
}
```

Output:

```
LinkedList:[good, achieve, goals, 2020, 2021]
```

```
The last element is removed:2021
Final LinkedList: [good, achieve, goals, 2020]
The first element is removed:good
Final LinkedList: [achieve, goals, 2020]
```

```
5.public class specificlist {
     public static int binarySearch(int[] nums, int flag) {
       int hi_num = nums.length - 1;
       int lo_num = 0;
       while (hi_num >= lo_num) {
          int guess = (lo_num + hi_num) >>> 1;
          if (nums[guess] > flag) {
             hi_num = guess - 1;
          } else if (nums[guess] < flag) {
             lo_num = guess + 1;
          } else {
             return guess;
          }
       }
       return -1;
    }

    public static void main(String[] args) {
       int[] nums = {1, 5, 6, 7, 8, 11};
       int search_num = 7;
       int index = binarySearch(nums, search_num);
       if (index == -1) {
          System.out.println(search_num + " is not in the array");
       } else {
          System.out.println(search_num + " is at index " + index);
       }
 }
}
```

Output:

```
7 is at index 3
```

```
6.public class BinarySearch {
 static int [] nums;

 public static void main(String[] args) {
  nums = new int[]{3,2,4,5,6,6,7,8,9,9,0,9};
  int result = Linear_Search(nums, 6);
  if(result == -1)
  {
   System.out.print("Not present in the array!");
  }
  else
  System.out.print("Number found at index "+result);
 }

 private static int Linear_Search(int [] nums,int search)
 {
  for(int i=0;i<nums.length;i++)
  {
   if(nums[i]==search)
   {
    return i;

   }
  }
  return -1;

 }
}
```

Output:
```
Number found at index 4
```

```
7.public class LinearSearch {
    public static int binarySearch(int[] nums, int flag) {
      int hi_num = nums.length - 1;
      int lo_num = 0;
      while (hi_num >= lo_num) {
         int guess = (lo_num + hi_num) >>> 1;
         if (nums[guess] > flag) {
            hi_num = guess - 1;
         } else if (nums[guess] < flag) {
            lo_num = guess + 1;
         } else {
            return guess;
         }
      }
      return -1;
   }

   public static void main(String[] args) {
      int[] nums = {1, 5, 6, 7, 8, 11};
      int search_num = 7;
      int index = binarySearch(nums, search_num);
      if (index == -1) {
         System.out.println(search_num + " is not in the array");
      } else {
         System.out.println(search_num + " is at index " + index);
      }
   }
}
```

Output:
```
7 is at index 3
```