```
package A7;
import java.util.Collections;
import java.util.LinkedList;
import java.util.List;
import java.util.ListIterator;
/**
 *
 * @author Lakshmi Priya
 */
public class TestLinkedList {
    public static void main(String[] args) {
        LinkedList<Integer> ll=new LinkedList<Integer>();
        System.out.println("Adding 1 in first position...");
        ll.addFirst(1);
        System.out.println("LinkedList after adding element: "+ll);

        System.out.println("Adding 2 in last position...");
        ll.addLast(2);
        System.out.println("LinkedList after adding element: "+ll);

        System.out.println("Adding 3 in 0th position...");
        ll.add(0,3);
        System.out.println("LinkedList after adding element: "+ll);

        System.out.println("Adding 4 in 1st position...");
        ll.add(1,4);
        System.out.println("LinkedList after adding element: "+ll);

        System.out.println("Adding 5 in 1st position...");
        ll.add(1,5);
        System.out.println("LinkedList after adding all elements:
"+ll);

        int remove=3;

        System.out.println("\nRemoved first element:
"+ll.pollFirst());
        System.out.println("Removed last element : "+ll.pollLast());
        System.out.println("LinkedList after removing elements:
"+ll);


        if(ll.contains(3)){
            ll.remove(remove);
            System.out.println("\nLinkedList after removing element
3: "+ll);
        }
        else
            System.out.println("\nElement "+remove+" cannot be
removed as it is not present in list");

        LinkedList<Integer> llcpy = null;
        llcpy=(LinkedList<Integer>) ll.clone();

        System.out.println("\nOriginal LinkedList : "+ll);
        System.out.println("Copied LinkedList   :"+llcpy);
```

```
        Collections.sort(llcpy);
        System.out.println("\nAfter sorting: "+llcpy);

        System.out.println("\nIndex of element 1: "+ll.indexOf(1));
        System.out.println("Index of element 9: "+ll.indexOf(9));

        ListIterator itr=llcpy.listIterator();

        System.out.println("\nSort without using algorithms: ");
        System.out.println("Forward order: ");
        while(itr.hasNext()){
            System.out.print(itr.next()+"\t");
        }

        System.out.println("\nReverse order: ");
        while(itr.hasPrevious()){
            System.out.print(itr.previous()+"\t");
        }

        System.out.println("\n\nSort using algorithms: ");
        System.out.println("Forward order: ");
        Collections.sort(llcpy);
        System.out.println(llcpy);

        System.out.println("Reverse order: ");
        Collections.sort(llcpy, Collections.reverseOrder());
        System.out.println(llcpy);

        int sum=0;
        while(itr.hasNext()){
            sum+=(int)itr.next();
        }

        System.out.println("\nSum of elements: "+sum);
        System.out.println("\nAverage of elements:
"+(double)sum/ll.size());

        List<Integer> subll=null;
        subll=ll.subList(2,ll.size());
        System.out.println("\nNew LinkedList from 3rd position:
"+subll);

        LinkedList<Integer> newlll=new LinkedList<Integer>();
        newlll.addAll(ll);
        newlll.addAll(subll);

        System.out.println("\nNew LinkedList after combination:
"+newlll);

    }
}
```