**Program :5a**

```
class InsertionSortLinkedList {
    static class Node {
        int data;
        Node next;

        Node(int data) {
            this.data = data;
            this.next = null;
        }
    }
Node head;
public void insert(int data) {
        Node newNode = new Node(data);
        if (head == null) {
            head = newNode;
            return;
        }
        Node temp = head;
        while (temp.next != null) {
            temp = temp.next;
        }
        temp.next = newNode;
    }
public void insertionSort() {
        if (head == null || head.next == null) return;
        Node sorted = null;
        Node current = head;
while (current != null) {
```

```
Node next = current.next;
            sorted = sortedInsert(sorted, current);
            current = next;
        }
        head = sorted;
    }private Node sortedInsert(Node sorted, Node newNode) {
        if (sorted == null || sorted.data >= newNode.data) {
            newNode.next = sorted;
            sorted = newNode;
        } else {
            Node temp = sorted;
            while (temp.next != null && temp.next.data < newNode.data) {
                temp = temp.next;
            }
            newNode.next = temp.next;
            temp.next = newNode;
        }
        return sorted;
    }public void display() {
        Node temp = head;
        while (temp != null) {
            System.out.print(temp.data + " ");
            temp = temp.next;
        }
        System.out.println();
    }

    public static void main(String[] args) {
        InsertionSortLinkedList list = new InsertionSortLinkedList();
        list.insert(30);
        list.insert(10);
        list.insert(50);
        list.insert(20);
        list.insert(40);

        System.out.println("Original List:");
        list.display();

        list.insertionSort();

        System.out.println("Sorted List:");
        list.display();
    }
}
```

**Output:**

```
Original List:
30 10 50 20 40
Sorted List:
10 20 30 40 50
```

**Program :5b**

```
class RemoveElementLinkedList {
    static class Node {
        int data;
        Node next;
        Node(int data) {
            this.data = data;
            this.next = null;
        }
    }Node head;
public void insert(int data) {
        Node newNode = new Node(data);
        if (head == null) {
            head = newNode;
            return;
        }
        Node temp = head;
        while (temp.next != null) {
            temp = temp.next;
        }
        temp.next = newNode;
    }public void remove(int key) {
        if (head == null) return;
        if (head.data == key) {
            head = head.next;
            return; }
 Node temp = head;
        while (temp.next != null && temp.next.data != key) {
temp = temp.next;
```

```
        }
 if (temp.next != null) {
            temp.next = temp.next.next;}
    }public void display() {
        Node temp = head;
        while (temp != null) {
            System.out.print(temp.data + " ");
            temp = temp.next;
        }
        System.out.println();
    }
public static void main(String[] args) {
        RemoveElementLinkedList list = new RemoveElementLinkedList();
        list.insert(10);
        list.insert(20);
        list.insert(30);
        list.insert(40);
        System.out.println("Original List:");
        list.display();
        list.remove(30);
        System.out.println("After removing 30:");
        list.display();
    }
}
```

**Output:**

```
Original List:

10 20 30 40

After removing 30:

10 20 40
```

**Program:5c**

```
import java.util.HashSet;

class RemoveDuplicatesLinkedList {

    static class Node {

        int data;

        Node next;

        Node(int data) {

            this.data = data;

            this.next = null;

        }}

 Node head;

 public void insert(int data) {

        Node newNode = new Node(data);

        if (head == null) {

            head = newNode;

            return; }

        Node temp = head;

        while (temp.next != null) {

            temp = temp.next;

        }

        temp.next = newNode;}

public void removeDuplicates() {

        HashSet<Integer> seen = new HashSet<>();

        Node current = head;

        Node prev = null;

while (current != null) {

            if (seen.contains(current.data)) {

                prev.next = current.next;

            } else {
```

```
        seen.add(current.data);
        prev = current;
      }
      current = current.next;
    }}
public void display() {
    Node temp = head;
    while (temp != null) {
      System.out.print(temp.data + " ");
      temp = temp.next; }
    System.out.println();}
public static void main(String[] args) {
    RemoveDuplicatesLinkedList list = new RemoveDuplicatesLinkedList();
    list.insert(10);
    list.insert(20);
    list.insert(10);
    list.insert(30);
    list.insert(20);
    System.out.println("Original List:");
    list.display();
    list.removeDuplicates();
    System.out.println("After removing duplicates:");
    list.display();
  }}
```

**Output:**

```
Original List:
10 20 10 30 20
After removing duplicates:
10 20 30
```