1. **Given two sorted linked lists, write a recursive procedure that produces a single linked list**

package main;

class Node {

public int data;

public Node next;

public Node(int data) {

this.data = data;

next = null;

}

public void displayData() {

System.out.print(this.data + " ");

}

}

class MergedLinkedList {

public static Node mergedLinkedListRecursion(Node node1, Node node2) {

if (node1 == null) {

return node2;

}

if (node2 == null) {

return node1;

}

if (node1.data <= node2.data) {

node1.next = mergedLinkedListRecursion(node1.next, node2);

return node1;

} else {

node2.next = mergedLinkedListRecursion(node1, node2.next);

return node2;

}

}

public static void displayLinkedList(Node n) {

Node current = n;

while (current != null) {

current.displayData();

current = current.next;

}

}

}

class LinkedList {

public Node first;

private Node mergedList;

public LinkedList() {

this.first = null;

}

public void insertFirst(int data) {

Node temp = new Node(data);

temp.next = this.first;

this.first = temp;

}

}

public class Main {

public static void main(String[] args) {

LinkedList list1 = new LinkedList();

list1.insertFirst(5);

list1.insertFirst(3);

list1.insertFirst(1);

LinkedList list2 = new LinkedList();

list2.insertFirst(6);

list2.insertFirst(4);

list2.insertFirst(2);

Node merged = MergedLinkedList.mergedLinkedListRecursion(list1.first, list2.first);

MergedLinkedList.displayLinkedList(merged);

}

}

Output:

1 2 3 4 5 6