```java
package linkedlist;

// Java program for flattening a Linked List
class LinkedList22
{
    Node head; // head of list

    /* Linked list Node*/
    class Node
    {
        int data;
        Node right, down;
        Node(int data)
        {
            this.data = data;
            right = null;
            down = null;
        }
    }

    // An utility function to merge two sorted linked lists
    Node merge(Node a, Node b)
    {
        // if first linked list is empty then second
        // is the answer
        if (a == null)   return b;

        // if second linked list is empty then first
        // is the result
        if (b == null)   return a;

        // compare the data members of the two linked lists
        // and put the larger one in the result
        Node result;

        if (a.data < b.data)
        {
            result = a;
            result.down = merge(a.down, b);
        }

        else
        {
            result = b;
            result.down = merge(a, b.down);
        }

        result.right = null;
        return result;
    }

    Node flatten(Node root)
    {
        // Base Cases
        if (root == null || root.right == null)
            return root;

        // recur for list on right
        root.right = flatten(root.right);

        // now merge
        root = merge(root, root.right);

        // return the root
        // it will be in turn merged with its left
        return root;
    }

    /* Utility function to insert a node at beginning of the
    linked list */
    Node push(Node head_ref, int data)
    {
        /* 1 & 2: Allocate the Node &
                Put in the data*/
        Node new_node = new Node(data);

        /* 3. Make next of new Node as head */
        new_node.down = head_ref;

        /* 4. Move the head to point to new Node */
        head_ref = new_node;

        /*5. return to link it back */
        return head_ref;
    }

    void printList()
    {
        Node temp = head;
        while (temp != null)
        {
            System.out.print(temp.data + " ");
            temp = temp.down;
        }
        System.out.println();
    }

    /* Driver program to test above functions */
    public static void main(String args[])
    {
        LinkedList22 L = new LinkedList22();

        /* Let us create the following linked list
            5 -> 10 -> 19 -> 28
            | |  |   |
            V V  V   V
            7 20 22 35
            |        |   |
            V        V   V
            8        50  40
            |            |
            V            V
            30           45
        */

        L.head = L.push(L.head, 30);
        L.head = L.push(L.head, 8);
        L.head = L.push(L.head, 7);
        L.head = L.push(L.head, 5);

        L.head.right = L.push(L.head.right, 20);
        L.head.right = L.push(L.head.right, 10);

        L.head.right.right = L.push(L.head.right.right, 50);
        L.head.right.right = L.push(L.head.right.right, 22);
        L.head.right.right = L.push(L.head.right.right, 19);

        L.head.right.right.right = L.push(L.head.right.right.right, 45);
        L.head.right.right.right = L.push(L.head.right.right.right, 40);
        L.head.right.right.right = L.push(L.head.right.right.right, 35);
        L.head.right.right.right = L.push(L.head.right.right.right, 20);
        // flatten the list
        L.head = L.flatten(L.head);

        L.printList();
    }
} /* This code is contributed by Rajat Mishra */
```