```
class Node<S> {
    S data;
    Node<S> next;

    public Node(S data) {
        this.data = data;
        this.next = null;
    }
}

class SinglyLinkedList<S> {
    private Node<S> head;

    public SinglyLinkedList() {
        this.head = null;
    }

    public void addFirst(S s) {
        Node<S> newNode = new Node<>(s);
        newNode.next = head;
        head = newNode;
    }

    public void printList() {
        Node<S> current = head;
        while (current != null) {
            System.out.print(current.data + " ");
            current = current.next;
        }
```

```
                System.out.println();
                {

public static <S> SinglyLinkedList<S> concatenate(SinglyLinkedList<S> L, SinglyLinkedList<S> M) {
                if (L.head == null) {
                    return M;
                {
                if (M.head == null) {
                    return L;
                {

                Node<S> current = L.head;
                while (current.next != null) {
                {
                {
                {

                public class Main {
                public static void main(String[] args) {
                SinglyLinkedList<Integer> listL = new SinglyLinkedList<>();
                listL.addFirst(10);
                listL.addFirst(20);
                listL.addFirst(30);

                SinglyLinkedList<Integer> listM = new SinglyLinkedList<>();
                listM.addFirst(40);
                listM.addFirst(50);
                listM.addFirst(60);
```

```
        System.out.print("List L: ");
        listL.printList();

        System.out.print("List M: ");
        listM.printList();

SinglyLinkedList<Integer> listLPrime = SinglyLinkedList.concatenate(listL, listM);

        System.out.print("List L' after concatenation: ");
        listLPrime.printList();
    {
        {
```