Question :

You are the P.E.T Master in a school. Students are already standing in height order. There are 2 rows of students standing in a line. You want to join the two rows as a single row and height order should be maintained.

Input Description :

Heights of Students standing in two rows.

Output Description :

Combined students in sorted order.

Tags :

Linked List, Array

Solution :

```
class Node:
    '''Initializes a Node for Singly Linked List'''
    def __init__(self, data):
        self.data = data
        self.next = None

class LinkedList:
    ''' Initializes a Linked List'''
    def __init__(self):
        self.head = None

    def printLinkedList(self):
        ''' Prints LinkedList '''
        temp = self.head
        while temp is not None:
            if temp.next is not None:
                print(temp.data, end = " ")
            else:
                print(temp.data, end = "")
            temp = temp.next
```

```
def createLinkedList(lst, n):
    ''' Creates a LinkedList '''
    ll = LinkedList()
    temp = ll.head
    for i in range(n):
        new_node = Node(lst[i])
        if ll.head is None:
            ll.head = new_node
            temp = new_node
        else:
            temp.next = new_node
            temp = new_node

    return ll

def merge_linked_list(ll1, ll2):
    ll3 = LinkedList()
    temp1 = ll1.head
    temp2 = ll2.head
    while temp1.next is not None and temp2.next is not None:
        if temp1.data <= temp2.data:
            new_node = Node(temp1.data)
            temp1 = temp1.next
        else:
            new_node = Node(temp2.data)
            temp2 = temp2.next
        if ll3.head is None:
            ll3.head = new_node
            temp = new_node
        else:
            temp.next = new_node
            temp = new_node
    while temp1.next is not None:
```

```
        new_node = Node(temp1.data)
        temp1 = temp1.next
        if ll3.head is None:
            ll3.head = new_node
            temp = new_node
        else:
            temp.next = new_node
            temp = new_node
    while temp2 is not None:
        new_node = Node(temp2.data)
        temp2 = temp2.next
        if ll3.head is None:
            ll3.head = new_node
            temp = new_node
        else:
            temp.next = new_node
            temp = new_node
    return ll3

n = int(input())
ll1 = createLinkedList([int(x) for x in input().split()], n)

m = int(input())
ll2 = createLinkedList([int(x) for x in input().split()], n)

merged_linkedlist = merge_linked_list(ll1, ll2)
merged_linkedlist.printLinkedList()
```

Test Cases:

Test case 1:
Input :
5
1 3 5 7 9
5

2 4 6 8 10
Output :
1 2 3 4 5 6 7 8 9 10

Test case 2:
Input :
3
1 5 9
4
2 4 6 8
Output :
1 2 4 5 6 8 9

Test case 3:
Input :
1
1
1
1
Output :
1 1

Test case 4:
Input :
2
11 13
4
2 4 6 8
Output :
2 4 6 8 11 13

Test case 5:
Input :
3
1 2 3
3
1 2 3
Output :
1 1 2 2 3 3