### Add two numbers represented by linked lists

Given two decimal numbers represented by two linked lists of size **N** and **M**respectively. The task is to return a linked list that represents the sum of these two numbers.

For example, the number **190** will be represented by the linked list, **1->9->0->null,**similarly **25**by **2->5->null.**Sum of these two numbers is190 + 25 =**215,**which will be represented by**2->1->5->null.**You are required to return the head of the linked list**2->1->5->null.**

**Example 1:**

**Input:**

N = 2

valueN[] = {4,5}

M = 3

valueM[] = {3,4,5}

**Output:** 3 9 0

**Explanation:** For the given two linked

list (4 5) and (3 4 5), after adding

the two linked list resultant linked

list will be (3 9 0).

**Example 2:**

**Input:**

N = 2

valueN[] = {6,3}

M = 1

valueM[] = {7}

**Output:** 7 0

**Explanation:** For the given two linked

list (6 3) and (7), after adding the

two linked list resultant linked list

will be (7 0).

**Expected Time Complexity:**O(N+M)  
**Expected Auxiliary Space:**O(Max(N,M)) for the resultant list.

**Constraints:**  
1 <= N, M <= 5000

**Company Tags**

[**Flipkart**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Flipkart) [**Morgan Stanley**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Morgan%20Stanley) [**Accolite**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Accolite) [**Amazon**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Amazon) [**Microsoft**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Microsoft) [**Snapdeal**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Snapdeal) [**MakeMyTrip**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=MakeMyTrip) [**Qualcomm**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Qualcomm)

//{ Driver Code Starts

// driver

import java.util.\*;

class Node {

int data;

Node next;

Node(int d) {

data = d;

next = null;

}

}

class CodingMaxima{

static void printList(Node n){

while(n!=null){

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

n = n.next;

}

System.out.println();

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int T = sc.nextInt();

while (T-- > 0) {

int n = sc.nextInt();

int val = sc.nextInt();

Node first = new Node(val);

Node tail = first;

for(int i=0; i<n-1; i++)

{

val = sc.nextInt();

tail.next = new Node(val);

tail = tail.next;

}

int m = sc.nextInt();

val = sc.nextInt();

Node second = new Node(val);

tail = second;

for(int i=0; i<m-1; i++)

{

val = sc.nextInt();

tail.next = new Node(val);

tail = tail.next;

}

Solution g = new Solution();

Node res = g.addTwoLists(first, second);

printList(res);

}

}

}

// } Driver Code Ends

class Solution{

public static Node reverse(Node head) {

Node current = head;

Node previous = null;

Node next = null;

while (current != null)

{

next = current.next;

current.next = previous;

previous = current;

current = next;

}

head = previous;

return head;

}

static Node addTwoLists(Node first, Node second){

first = reverse(first);

second = reverse(second);

Node dummy = new Node(0);

Node temp = dummy;

int carry = 0;

while (first != null || second != null || carry == 1){

int sum = 0;

if (first != null){

sum += first.data;

first = first.next;

}

if (second != null){

sum += second.data;

second = second.next;

}

sum += carry;

carry = sum/10;

Node newnode = new Node(sum%10);

temp.next = newnode;

temp = temp.next;

}

return reverse(dummy.next);

}

}