# 2. Add Two Numbers

You are given two non-empty linked lists representing two non-negative integers. The digits are stored in reverse order, and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

## SOLUTION IN JAVA

class Solution {

// Add Two Numbers (Java improved)

public ListNode addTwoNumbers(ListNode l1, ListNode l2) {

ListNode dummyHead = new ListNode(0);

ListNode curr = dummyHead;

int carry = 0;

while (l1 != null || l2 != null || carry != 0) {

int x = (l1 != null) ? l1.val : 0;

int y = (l2 != null) ? l2.val : 0;

int sum = carry + x + y;

carry = sum / 10;

curr.next = new ListNode(sum % 10);

curr = curr.next;

if (l1 != null) l1 = l1.next;

if (l2 != null) l2 = l2.next;

}

return dummyHead.next;

}

}