Name: Dur Muhammad

Roll No: ARI-F23-0064

A I –B

TASK 1

```
 9    *     }
10    * }
11    */
12   public class Solution {
13       public ListNode getIntersectionNode(ListNode headA, ListNode headB) {
14           if(headA==null || headB==null){
15               return null;
16           }
17           ListNode A=headA;
18           ListNode B=headB;
19           while(A != B){
20               A=(A==null)? headB:A.next;
21               B=(B==null)? headA:B.next;
22           }
23           return A;
24       }
```

Saved

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

intersectVal =

8

listA =

[4,1,8,4,5]

listB =

Task 2

```
 9   * }
10   */
11  class Solution {
12      public ListNode deleteDuplicates(ListNode head) {
13          ListNode current=head;
14          while(current != null && current.next!=null){
15              if(current.val==current.next.val){
16                  current.next=current.next.next;
17              }
18              else{
19                  current=current.next;
20              }
21          }
22          return head;
23      }
24  }
```

Saved

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

head =

[1,1,2]

Output

[1,2]

Expected

[1,2]

```
 */
class Solution {
    public ListNode mergeTwoLists(ListNode list1, ListNode list2) {


ListNode list = new ListNode(-1);
ListNode current = list;
while(list1 != null && list2 != null) {
    if(list1.val < list2.val) {
        current.next = list1;
        list1 = list1.next;
    }else {
        current.next = list2;
        list2 = list2.next;
    }
    current = current.next;
}
if(list1 != null) {
    current.next = list1;
}
if(list2 != null) {
    current.next = list2;
}
return list.next;
    }
    }

```

Testcase | Test Result

Input

list1 =

[1,2,4]

list2 =

[1,3,4]

Output

[1,1,2,3,4,4]

Expected

[1,1,2,3,4,4]

Task 4

```
 */
class Solution {
    public ListNode addTwoNumbers(ListNode l1, ListNode l2) {


ListNode list = new ListNode(0);
ListNode current = list;
int carry = 0;

while(l1 != null || l2 != null || carry != 0) {
    int val1 = (l1 != null) ? l1.val : 0;
    int val2 = (l2 != null) ? l2.val : 0;
    int sum = val1+val2+carry;
    carry = sum/10;
    int digit = sum%10;
    current.next = new ListNode(digit);
    current = current.next;
    if(l1 != null) {
        l1 = l1.next;
    }
    if(l2 != null) {
        l2 = l2.next;
    }
}
    return list.next;
    }
}

```

Input

l1 =

[2,4,3]

l2 =

[5,6,4]

Output

[7,0,8]

Expected