**WEEK - 4**

**R- 7.1** :

def findSecondLastNode(head):

temp = head

# If the list is empty or contains less than 2 nodes

if (temp == None or temp.next == None):

return -1

# Traverse the linked list

while (temp != None):

# Check if the current node is the second last node or not

if (temp.next.next == None):

return temp.data

# If not then move to the next node

temp = temp.next

**R - 7.3** :

Def getCount(self, node):

if (not node): # Base case

return 0

else:

return 1 + self.getCountRec(node.next)

**R - 7.6** :

def checkSameOrDifferentCL(L, x, y):

temp = L.\_tail # head

# tsum = temp.data

temp = temp.\_next

xInList = False

yInList = False

while(temp is not L.\_tail):

if temp is x:

xInList = True

if temp is y:

yInList = True

temp = temp.\_next

return xInList and yInList

**R - 7.28** :

def reverseLL(node):

if (node == None):

return node

if (node.next == None):

return node

node1 = reverse(node.next)

node.next.next = node

node.next = None

return node1