**Program to Count the Number of Occurrences of an Element in the Linked List using Recursions**

class Node:

def \_\_init\_\_(self, data):

self.data = data

self.next = None

class LinkedList:

def \_\_init\_\_(self):

self.head = None

self.last\_node = None

def append(self, data):

if self.last\_node is None:

self.head = Node(data)

self.last\_node = self.head

else:

self.last\_node.next = Node(data)

self.last\_node = self.last\_node.next

def display(self):

current = self.head

while current:

print(current.data, end = ' ')

current = current.next

def count(self, key):

return self.count\_helper(self.head, key)

def count\_helper(self, current, key):

if current is None:

return 0

if current.data == key:

return 1 + self.count\_helper(current.next, key)

else:

return self.count\_helper(current.next, key)

a\_llist = LinkedList()

for data in [9, 4, 9, 4, 7, 3, 9, 1, 2, 3]:

a\_llist.append(data)

print('The linked list: ', end = '')

a\_llist.display()

print()

key = int(input('Enter data item: '))

count = a\_llist.count(key)

print('{0} occurs {1} time(s) in the list.'.format(key, count))

