Task 1:

Complete the recursive function flattenList which will take a nested python list and an empty list as parameters. The function should convert the nested list into a new flat list sequentially. Check the input-output to understand the sequence pattern.

def flattenList(given\_list, output\_list):

# To Do

given\_list = [1, [2, [3, [4], 5], 6], 7, 8, [9, [[10, 11], 12], 13], 14, [15, [16, [17]]]] output\_list = flattenList(given\_list, []) # Initial empty list is sent for update

output\_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]

Your code must work for any depth of the nested linked list. You can not use any loop in the function. You can use built-in functions like append, type, etc.

Ungraded Task: Do this now for Singly Linked List Instead of Python’s built-in list.

Hint:

Your node class

class Node:

def \_\_init\_\_(self, next, bottom, val):

self.next = next # for next item

self.bottom = bottom # for nested item check

self.val = val # The integer value.