def len\_Linkedlist(n):

if not (isinstance(n,Node) or n==None):

return 'Your input is wrong'

elif n==None:

return 0

else:

try:

return 1+len\_Linkedlist(n.pointer)

except:

return 'Your input is wrong'

I also use the Linkedlist class and Node class which has been taught in class.

First Step, I need to check whether the input of the function is a Node. If not the Node, the function will return ‘Your input is wrong ’ and quit automatically. And if n=none, the function will return 0. The final step is recursive. In this area I use try and except method to avoid such situation: the pointer of one of the Node is not Node or none in the following steps.

a=Linkedlist()

a.add\_last(10)

a.add\_last(8)

a.add\_first(9)

a.add\_last(15)

c=3

b=Node(7,c)

print(len\_Linkedlist(a.head))

print(len\_Linkedlist(b))

I have tested 2 situation, for the first, the function will output 4 correctly. For the second, it can avoid the mistake robust.