#DSA-Exer-2

class Node:

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

self.\_\_data=data

self.\_\_next=None

def get\_data(self):

return self.\_\_data

def set\_data(self,data):

self.\_\_data=data

def get\_next(self):

return self.\_\_next

def set\_next(self,next\_node):

self.\_\_next=next\_node

class LinkedList:

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

self.\_\_head=None

self.\_\_tail=None

def get\_head(self):

return self.\_\_head

def get\_tail(self):

return self.\_\_tail

def add(self,data):

new\_node=Node(data)

if(self.\_\_head is None):

self.\_\_head=self.\_\_tail=new\_node

else:

self.\_\_tail.set\_next(new\_node)

self.\_\_tail=new\_node

def display(self):

temp=self.\_\_head

while(temp is not None):

print(temp.get\_data())

temp=temp.get\_next()

#You can use the below \_\_str\_\_() to print the elements of the DS object while debugging

def \_\_str\_\_(self):

temp=self.\_\_head

msg=[]

while(temp is not None):

msg.append(str(temp.get\_data()))

temp=temp.get\_next()

msg=" ".join(msg)

msg="Linkedlist data(Head to Tail): "+ msg

return msg