1. class Node:

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

self.data=data

self.next=None

class LinkedList:

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

self.head=None #starting point of LL

def insert(self,data):

new\_node=Node(data)

if self.head is None:

self.head=new\_node

else:

temp=self.head

while temp.next:

temp=temp.next

temp.next=new\_node

def display(self):

temp=self.head

while temp:

print(temp.data,end="->")

temp=temp.next

print("None")

ll=LinkedList()

ll.insert(10)

ll.insert(20)

ll.insert(30)

ll.insert(40)

ll.display()

1. class Node:

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

self.data=data

self.next=None

class LinkedList:

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

self.head=None #starting point of LL

def insert(self,data):

new\_node=Node(data)

if self.head is None:

self.head=new\_node

else:

temp=self.head

while temp.next:

temp=temp.next

temp.next=new\_node

def display(self):

temp=self.head

while temp:

print(temp.data,end=

"->")

temp=temp.next

print("None")

def add\_ll(self):

temp=self.head

sum=0

while temp:

sum=sum+temp.data

temp=temp.next

return sum

ll=LinkedList()

ll.insert(10)

ll.insert(20)

ll.insert(30)

ll.insert(40)

ll.display()

print("sum of elements in ll is ")

ans=ll.add\_ll()

print(ans)

1. class Node:

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

self.data=data

self.next=None

class LinkedList:

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

self.head=None #starting point of LL

def insert\_end(self,data):

new\_node=Node(data)

if self.head is None:

self.head=new\_node

else:

temp=self.head

while temp.next:

temp=temp.next

temp.next=new\_node

def display(self):

temp=self.head

while temp:

print(temp.data,end=

"->")

temp=temp.next

print("None")

def add\_ll(self):

temp=self.head

sum=0

while temp:

sum=sum+temp.data

temp=temp.next

return sum

def Count(self):

pass

def insert\_begnning(self,data):

new\_node=Node(data)

new\_node.next=self.head

self.head=new\_node

def delete\_begining(self):

self.head=self.head.next

ll=LinkedList()

ll.insert\_end(10)

ll.insert\_begnning(8888)

ll.display()

ll.insert\_end(20)

ll.insert\_end(30)

ll.insert\_end(40)

ll.delete\_begining()

ll.display()

print(" sum of elements in ll is ")

ans=ll.add\_ll()

print(ans)

1. class Node:

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

self.data=data

self.next=None

class LinkedList:

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

self.head=None #starting point of LL

def insert\_end(self,data):

new\_node=Node(data)

if self.head is None:

self.head=new\_node

else:

temp=self.head

while temp.next:

temp=temp.next

temp.next=new\_node

def display(self):

temp=self.head

while temp:

print(temp.data,end=

"->")

temp=temp.next

print("None")

def add\_ll(self):

temp=self.head

sum=0

while temp:

sum=sum+temp.data

temp=temp.next

return sum

def Count(self):

pass

def insert\_begnning(self,data):

new\_node=Node(data)

new\_node.next=self.head

self.head=new\_node

def delete\_begining(self):

self.head=self.head.next

def delete\_end(self):

temp=self.head

while temp.next.next:

temp=temp.next

temp.next=None

ll=LinkedList()

ll.insert\_end(10)

ll.insert\_begnning(8888)

ll.display()

ll.insert\_end(20)

ll.insert\_end(30)

ll.insert\_end(40)

ll.delete\_begining()

ll.display()

print(" sum of elements in ll is ")

ans=ll.add\_ll()

print(ans)

1. class Node:

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

self.data=data

self.next=None

class LinkedList:

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

self.head=None #starting point of LL

def insert\_end(self,data):

new\_node=Node(data)

if self.head is None:

self.head=new\_node

else:

temp=self.head

while temp.next:

temp=temp.next

temp.next=new\_node

def display(self):

temp=self.head

while temp:

print(temp.data,end= "->")

temp=temp.next

print("None")

def add\_ll(self):

temp=self.head

sum=0

while temp:

sum=sum+temp.data

temp=temp.next

return sum

def Count(self):

pass

def insert\_begnning(self,data):

new\_node=Node(data)

new\_node.next=self.head

self.head=new\_node

def delete\_begining(self):

self.head=self.head.next

def delete\_end(self):

temp=self.head

while temp.next.next:

temp=temp.next

temp.next=None

def insert\_position(self,pos,data):

if(pos==0):

self.insert\_begnning(data)

else:

new\_node=Node(data)

temp=self.head

for i in range(pos-1):

if temp is None:

print("Position out of bounds")

return

temp=temp.next

temp.next=temp.next.next

def delete\_value(self,value):

if self.head.data==value:

self.head=self.head.next

return

temp=self.head

while temp.next and temp.next.data !=value:

temp=temp.next

if temp.next is None:

print("value not found")

return

temp.next=temp.next.next

ll=LinkedList()

ll.insert\_begnning(10)

ll.insert\_begnning(20)

ll.insert\_begnning(30)

ll.insert\_position(3,900)

ll.insert\_end(40)

ll.delete\_position(2)

ll.delete\_value

ll.display()