INSERTING LINKED LIST:

class Node:

def \_init\_(self,data):

self.data = data

self.next = None

class LinkedList:

def \_init\_(self):

self.head = None #starting poing of linked list

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 #moves temp node

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()

o/p:

10->20->30->40->None

SUM OF THE LINKED LIST:

class Node:

def \_init\_(self,data):

self.data = data

self.next = None

class LinkedList:

def \_init\_(self):

self.head = None #starting poing of linked list

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 #moves temp node

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 the linked list")

ans=ll.add\_ll()

print(ans)

O/P:

10->20->30->40->None

sum of elements in the linked list

100

INSERT THE ELEMENT AT THE BEGINING:

class Node:

def \_init\_(self,data):

self.data = data

self.next = None

class LinkedList:

def \_init\_(self):

self.head = None #starting poing of linked list

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 #moves temp node

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\_begining(self,data):

new\_node=Node(data)

new\_node.next=self.head

self.head=new\_node

ll=LinkedList()

ll.insert\_end(10)

ll.insert\_begining(7000)

ll.insert\_end(20)

ll.insert\_end(30)

ll.insert\_end(40)

ll.display()

print("sum of elements in the linked list")

ans=ll.add\_ll()

print(ans)

class Node:

def init(self,data):

self.data = data

self.next = None

class LinkedList:

def init(self):

self.head = None #starting poing of linked list

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 #moves temp node

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\_begining(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\_begining(data)

else:

new\_node=Node(data)

temp=self.head

for \_ in range(pos-1):

if temp is None:

print("position out of bounds")

return

temp=temp.next

new\_node.next=temp.next

temp.next=new\_node

def delete\_position(self,pos):

if(pos==0):

self.delete-begining(data)

else:

temp = self.head

for \_ in range(pos-1):

if temp.next is None:

print("position out of bound")

return

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\_begining(10)

ll.insert\_begining(20)

ll.insert\_begining(30)

ll.insert\_position(3,900)

ll.insert\_end(40)

ll.delete\_position(2)

ll.delete\_value(20)

ll.display()

print("sum of elements in the linked list")

ans=ll.add\_ll()

print(ans)

def check\_password\_strength(password):

score = 0