#Name: Shihab Muhtasim

#ID: 21301610

#Sec: 1

#TASK 1 LINKED LIST STACK

class Node:

def \_\_init\_\_(self,elem,next):

self.elem=elem

self.next=next

class Stack:

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

self.top=None

def push(self,top):

if self.top==None:

self.top=Node(top,None)

else:

n=Node(top,self.top)

self.top=n

def pop(self):

if self.top==None:

return "Underflow"

else:

val=self.top.elem

self.top=self.top.next

return val

def peek(self):

if self.top==None:

return "Underflow"

else:

val=self.top.elem

return val

#----------part2-----------

exp="1+2]\*[3\*3+{4–5(6(7/8/9)+10)–11+(12\*8)]+14"

s=Stack()

idx=Stack()

c=0

status=None

for i in exp:

if i =="[" or i=="{" or i=="(":

s.push(i)

c+=1

idx.push(c)

elif i =="]" or i=="}" or i==")":

val=s.pop()

c+=1

x=idx.pop()

if val=='(' and i!=')':

flag=False

status="not closed"

char=val

break

elif val=='[' and i!=']':

flag=False

status="not closed"

char=val

break

elif val=='{' and i!='}':

flag=False

status="not closed"

char=val

break

elif val=='Underflow':

flag=False

status='not opened'

char=i

break

else:

c+=1

temp=s.pop()

if temp!="Underflow" and flag==True:

flag=False

status='not closed'

for i in range(len(exp)):

if exp[i]==temp:

c=i+1

break

if flag==True:

print(exp)

print("This expression is correct.")

elif flag==False and status=="not opened":

print(exp)

print("This expression is NOT correct.")

print(f"Error at character # {c}. ‘{char}‘ - {status}.")

else:

print(exp)

print("This expression is NOT correct.")

print(f"Error at character # {x}. ‘{char}‘ - {status}.")

#TASK 2 ARRAY STACK

class Stack:

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

self.stk=[None]\*len

self.top=0

def push(self,elem):

if self.top>=len(self.stk):

print("Overflow")

else:

self.stk[self.top]=elem

self.top+=1

def pop(self):

if self.top<=0:

return "Underflow"

else:

self.top-=1

val=self.stk[self.top]

self.stk[self.top]=None

return val

#----------part2-----------

exp="1+2\*[3\*3+{4–5(6(7/8/9)+10)–11+(12\*8)]+14"

s=Stack(len(exp))

idx=Stack(len(exp))

c=0

status=None

for i in exp:

if i =="[" or i=="{" or i=="(":

s.push(i)

c+=1

idx.push(c)

elif i =="]" or i=="}" or i==")":

val=s.pop()

c+=1

x=idx.pop()

if val=='(' and i!=')':

flag=False

status="not closed"

char=val

break

elif val=='[' and i!=']':

flag=False

status="not closed"

char=val

break

elif val=='{' and i!='}':

flag=False

status="not closed"

char=val

break

elif val=='Underflow':

flag=False

status='not opened'

char=i

break

else:

c+=1

temp=s.pop()

if temp!="Underflow" and flag==True:

flag=False

status='not closed'

for i in range(len(exp)):

if exp[i]==temp:

c=i+1

break

if flag==True:

print(exp)

print("This expression is correct.")

elif flag==False and status=="not opened":

print(exp)

print("This expression is NOT correct.")

print(f"Error at character # {c}. ‘{char}‘ - {status}.")

else:

print(exp)

print("This expression is NOT correct.")

print(f"Error at character # {x}. ‘{char}‘ - {status}.")