Exercise:3.1.1

max\_size=5

stack=[]

top=-1

def push(book\_title):

global top

if top >= max\_size-1:

print(" stack over flow! cannot push more books.")

else:

top +=-1

stack.append(book\_title)

print(f"book'(book title)' pushed onto the stack.")

def pop():

global top

if top==-1:

print("stack under flow ! cannot pop any book.")

else:

removed\_book=stack.pop()

print(f"book'(removed\_book)' popped from the stack.")

top==-1

def peek():

if top==-1:

print("stack is empty. no book to peek.")

else:

print(f"top book on the stack:'(stack[top])'")

def display():

if top==-1:

print("stack is empty.")

else:

print("book in stack (top to bottom):")

for i in range (top,-1,-1):

print(f"{i+1}.{stack[i]}")

push("The sum also rises")

push("The jungle book")

push("All around the world in 18 days")

push("Chirst topper")

push("The name last")

push("The genius wallet")

display()

peek()

pop()

pop()

display()

peek()

output :

book'(book title)' pushed onto the stack.

book'(book title)' pushed onto the stack.

book'(book title)' pushed onto the stack.

book'(book title)' pushed onto the stack.

book'(book title)' pushed onto the stack.

book'(book title)' pushed onto the stack.

book in stack (top to bottom):

top book on the stack:'(stack[top])'

book'(removed\_book)' popped from the stack.

book'(removed\_book)' popped from the stack.

book in stack (top to bottom):

top book on the stack:'(stack[top])'

Exercise:3.1.2

class Node:

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

self.data=data

self.next=None

class Stack:

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

self.top=None

def is\_empty(self):

return self.top is None

def push(self,data):

new\_node=Node(data)

new\_node.next=self.top

self.top=new\_node

def pop(self):

if self.is\_empty():

return None

popped\_data=self.top.data

self.top=self.top.next

return popped\_data

def peek(self):

if self.is\_empty():

return None

return self.top.data

stack=Stack()

stack.push(1)

stack.push(2)

stack.push(3)

print("peek:",stack.peek())

print("pop:",stack.pop())

print("pop:",stack.pop())

print("peek:",stack.peek())

print("is empty:",stack.is\_empty())

output:

peek: 3

pop: 3

pop: 2

peek: 1

is empty: False