Exercise 3:

PROGRAM 3(a)

MAX\_SIZE=5

stack=[]

top=-1

def push(book\_title):

global top

if top>=MAX\_SIZE-1:

print("stack overflow!")

else:

top+=1

stack.append(book\_title)

print(f"book'{book\_title}'pushed")

def pop():

global top

if top==-1:

print("stack underflow!")

else:

remove\_book=stack.pop()

print(f"book'{remove\_book}' popped")

top-=1

def peek():

if top==-1:

print("stack is empty")

else:

print(f" top book-'{stack[top]}'")

def display():

if top==-1:

print("stack is empty")

else:

print("top to bottom:")

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

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

push("harry potter")

push("wings of fire")

push("rabbit and turtle")

push("the merchant")

push("love")

push("extra love")

display()

peek()

pop()

pop()

display()

peek()

OUTPUT:

book'harry potter'pushed

book'wings of fire'pushed

book'rabbit and turtle'pushed

book'the merchant'pushed

book'love'pushed

stack overflow!

top to bottom:

5.love

4.the merchant

3.rabbit and turtle

2.wings of fire

1.harry potter

top book-'love'

book'love' popped

book'the merchant' popped

top to bottom:

3.rabbit and turtle

2.wings of fire

1.harry potter

top book-'rabbit and turtle'

PROGRAM 3 (b)

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):

newnode=Node(data)

newnode.next=self.top

self.top=newnode

def pop(self):

if self.is\_empty():

return None

else:

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 of the stack:",stack.peek())

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

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

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

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

OUTPUT:

peek of the stack: 3

popped data: 3

popped data: 2

peek of the stack: 1

Is empty: False