DAY-3:

STACKS AND QUEUES:

PROGRAM:

stack = []

top = -1

MAX\_SIZE = 5

def push(element):

global top

if top == MAX\_SIZE - 1:

print("Stack is full")

else:

stack.append(element)

top += 1

print(f"Pushed {element} into stack")

push(10)

push(20)

push(30)

push(40)

push(50)

push(60)

print("Stack contents:", stack)

OUTPUT:

Pushed 10 into stack

Pushed 20 into stack

Pushed 30 into stack

Pushed 40 into stack

Pushed 50 into stack

Stack is full

Stack contents: [10, 20, 30, 40, 50]

PROGRAM:

def push(value):

top=-1

if(top==4):

return"Stack is full"

else:

top=top+1

return stack.append(value)

stack=[10]

push(20)

push(30)

print(stack)

OUTPUT:

[10, 20, 30]

PROGRAM POP:

def push(value):

top=1

if(top==3):

return"Stack is full"

else:

top=top+1

return stack.append(value)

stack=[10]

push(20)

push(30)

stack.pop()

stack.pop()

print(stack)

OUTPUT:

[10]

PROGRAM:

def push(value):

top=1

if(top==3):

return"Stack is full"

else:

top=top+1

return stack.append(value)

def pop():

top=5

if(top!=-1):

return stack.pop()

else:

top=-1

return "Stack is empty"

stack=[10,20,30,40,50]

pop()

pop()

push(20)

push(30)

print(stack)

OUTPUT:

[10, 20, 30, 20, 30]

PROGRAM:

stack=[]

top=-1

def push(value):

global top

stack.append(value)

top+=1

def pop():

global top

if top==-1:

print("stack is empty nothing to pop")

else:

stack.pop()

top-=1

def peek():

if top==-1:

return "Stack is empty. No top element"

else:

return f"top element={stack[top]}"

def display():

if(top==-1):

print("empty")

else:

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

print(stack[i])

push(10)

push(30)

push(50)

push(70)

pop()

pop()

print(peek())

display()

OUTPUT:

top element=30

30

10

Q. Initiate a while loop user enters 1 push 2 pop 3 peek 4 Display all the elements in the stack 5 exit

stack = []

def push():

element = input("Enter element to push: ")

stack.append(element)

print(f"pushed onto the stack.")

def pop():

if not stack:

print("Stack is empty. Cannot pop.")

else:

popped = stack.pop()

print(f"Popped element:")

def peek():

if not stack:

print("Stack is empty.")

else:

print(f"Top element is: ")

def display():

if not stack:

print("Stack is empty.")

else:

print("Stack elements are:")

for item in reversed(stack):

print(item)

while True:

choice = input("Enter your choice : ")

if choice == '1':

push()

elif choice == '2':

pop()

elif choice == '3':

peek()

elif choice == '4':

display()

elif choice == '5':

print("Exiting program.")

break

else:

print("Invalid choice. Please enter a number between 1 and 5.")

OUTPUT:  
Menu:

1. Push

2. Pop

3. Peek

4. Display

5. Exit

Enter your choice (1-5): 1

Enter element to push: 10

10 pushed onto the stack.

Menu:

1. Push

2. Pop

3. Peek

4. Display

5. Exit

Enter your choice (1-5): 1

Enter element to push: 20

20 pushed onto the stack.

Menu:

1. Push

2. Pop

3. Peek

4. Display

5. Exit

Enter your choice (1-5): 4

Stack elements are (from top to bottom):

20

10

QUEUE:

PROGRAM:

class Queue:

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

self.queue = []

self.front = -1

self.rear = -1

def enqueue(self, value):

if self.front == -1:

self.front = 0

self.rear += 1

self.queue.append(value)

print(f"Enqueued {value}")

def dequeue(self):

if self.is\_empty():

print("Queue is empty! Cannot dequeue.")

return

value = self.queue.pop(0)

self.rear -= 1

if self.rear < 0:

self.front = -1

print(f"Dequeued {value}")

def is\_empty(self):

return self.front == -1

def display(self):

if self.is\_empty():

print("Queue is empty.")

else:

print("Queue elements:", self.queue)

# Create queue object

q = Queue()

# Menu-driven code

while True:

print("\nMenu:\n1. Enqueue\n2. Dequeue\n3. Display\n4. Exit")

choice = input("Enter your choice: ")

if choice == '1':

value = input("Enter value to enqueue: ")

q.enqueue(value)

elif choice == '2':

q.dequeue()

elif choice == '3':

q.display()

elif choice == '4':

print("Exiting program.")

break

else:

print("Invalid choice. Please enter a number between 1 and 4.")

OUTPUT:

Enter your choice: 1

Enter value to enqueue: 10

Enqueued 10

Enter your choice: 3

Queue elements: ['10']