'''

# Experiment 5 : STACK, QUEUE AND LINKED LIST IN PYTHON

Name:  Khan Arshad Abdulla

Roll No: 20CO24

2021-2022

THORY:

1.Stack

Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO(Last In First Out) or FILO

(First In Last Out).In programming terms, putting an item on top of the stack is called push and removing an item is called pop.There are some basic operations that allow us to perform different actions on a stack.

Push: Add an element to the top of a stack

Pop: Remove an element from the top of a stack

IsEmpty: Check if the stack is empty

IsFull: Check if the stack is full

Peek: Get the value of the top element without removing it

2.Queue

A Queue is a linear structure which follows a particular order in which the operations are performed. The order is First In First Out (FIFO). A good

example of a queue is any queue of consumers for a resource where the consumer that came first is served first. The difference between stacks and queues is in removing. In a stack we remove the item the most recently added; in a queue, we remove the item the least recently added.

In programming terms, putting items in the queue is called enqueue, and removing items from the queue is called dequeue.

A queue is an object (an abstract data structure - ADT) that allows the following operations:

Enqueue: Add an element to the end of the queue

Dequeue: Remove an element from the front of the queue

IsEmpty: Check if the queue is empty

IsFull: Check if the queue is full

Peek: Get the value of the front of the queue without removing it

3.Linked List

A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. In simple words, a linked list consists of

nodes where each node contains a data field and a reference(link) to the next node in the list.

Following are the basic operations supported by a list.

Insertion − Adds an element at the beginning of the list.

Deletion − Deletes an element at the beginning of the list.

Display − Displays the complete list.

Search − Searches an element using the given key.

Delete − Deletes an element using the given key.

'''

```
from collections import deque



class Stack:
    def __init__(self):
        self.data = list()

    def push(self, d):
        try:
            self.data.append(d)
        except Exception:
            return False
        return True
```

```
    def pops(self):
        if not self.data:
            return "Stack is empty"
        return +str(self.data.pop())

    def display(self):
        if not self.data:
            print('Stack is empty.')
        else:
            print('Stack Content ->', self.data[::-1])

    def __str__(self):
        if not self.data:
            return 'Stack is empty.'
        else:
            return 'Stack Content -> '+str(self.data[::-1])


class Queue:
    def __init__(self):
        self.data = deque()

    def insert(self, d):
        self.data.append(d)

    def remove(self):
        if self.data:
            return "Removed Element is "+str(self.data.popleft())
        else:
            return "Cant remove from an empty Queue."
```

```
    def display(self):
        if not self.data:
            print('Queue is empty.')
        else:
            print('Queue Content ->', end=" ")
            for i in self.data:
                print(i, end="\t")
            print()


class Node:
    def __init__(self, data=None):
        self.data = data
        self.next = None


class Linked_List:
    def __init__(self, data=None):
        if data:
            self.head = Node(data)
        else:
            self.head = None

    def insertAtBeg(self, d):
        newnode = Node(d)
        newnode.next = self.head
        self.head = newnode

    def display(self):
        if self.head:
```

```
        temp = self.head
        print('Linked List:')
        while temp:
            print(temp.data)
            temp = temp.next
    else:
        print('Linked List is empty.')

# Insert at end
def insertAtEnd(self, d):
    newNode = Node(d)
    if(self.head == None):
        self.head = newNode
        return
    else:
        temp = self.head
        while(temp.next != None):
            temp = temp.next
        temp.next = newNode

# delete first element
def removeFirst(self):
    if(self.head != None):
        self.head = self.head.next

# delete last element
def removeLast(self):
    if(self.head != None):
        if(self.head.next == None):
            self.head = None
        else:
```

```
            temp = self.head
            while(temp.next.next != None):
                temp = temp.next
        lastNode = temp.next
        temp.next = None
        lastNode = None
    else:
        return "List is empty"

# Insert after
def insertAfter(self, key, d):
    temp = self.head
    while temp and temp.data != key:
        temp = temp.next
    if temp:
        newnode = Node(d)
        newnode.next = temp.next
        temp.next = newnode
        return "Node inserted."
    return "Key Node does not exist."

def removeBefore(self, key):
    temp = self.head
    while temp and temp.data != key:
        prev = temp
        temp = temp.next
    if temp == self.head:
        return "Cannot remove before first node."
    elif temp:
        if prev == self.head:
            self.head = prev.next
```

```
            else:
                temp = self.head
                while temp.next != prev:
                    temp = temp.next
                temp.next = prev.next
                prev.next = None
            return "Node Removed."
        return "Key Node does not exist."

# modify main function to have MENU with all required options


def main():
    while True:
        print("\n\t\tMENU\t\t\n1.Stack \n2.Queue \n3.Linked List\n4.Exit")
        ch = int(input("Enter your choice: "))
        print("\n")
        if ch == 1:
            print("\t\tStack")
            st = Stack()
            while True:
                print("1.Push \n2.Pop \n3.Display \n4.Exit")
                ch = int(input("Enter your choice: "))
                if ch == 1:

                    d = int(input("Enter the number to be pushed :"))
                    st.push(d)
                    print("Data successfully pushed")
                    print("\n")

                elif ch == 2:
```

```
            if st.data:
                st.pops()
                print("Succesfully popped")
                print("\n")
            else:
                print("List is empty")
                print("\n")
        elif ch == 3:
            st.display()
        elif ch == 4:
            break
        else:
            print("Please enter a valid choice")
elif ch == 2:
    print("\t\tQueue")
    q = Queue()
    while True:
        print("1.Insert \n2.Remove \n3.Display\n4.Exit")
        ch = int(input("Enter your choice: "))
        if ch == 1:
            d = int(input("Enter the number to be inserted :"))
            q.insert(d)
            print("Data successfully inserted")
            print("\n")
        elif ch == 2:
            if q.data:
                print("removed element is :", q.remove())
            else:
                print("Queue is empty")
                print("\n")
        elif ch == 3:
```

```
            q.display()
        elif ch == 4:
            break
        else:
            print("please enter a valid choice")
    elif ch == 3:
        print("\t\tLinked List")
        l = Linked_List()
        while True:
            print("1.Insert at beginning \n2.Insert at end \n3.Insert after\n4.Remove first element \n5.Remove last element \n6.Remove element before \n7.Display \n8.Exit")
            ch = int(input("Enter your choice: "))
            if ch == 1:
                d = int(input("Enter the number to be inserted :"))
                l.insertAtBeg(d)
                print("Data successfully inserted")
                print("\n")
            elif ch == 2:
                d = int(input("Enter the number to be inserted :"))
                l.insertAtEnd(d)
                print("Data successfully inserted")
                print("\n")
            elif ch == 3:
                key = int(input("Enter the position :"))
                d = int(input("Enter the number to be inserted :"))
                print(l.insertAfter(key, d))
                print("\n")

            elif ch == 4:
                if l.head:
                    l.removeFirst()
```

```
            print("Data successfully removed")
            print("\n")
        else:
            print("List is empty")
            print("\n")

    elif ch == 5:
        if l.head:
            l.removeLast()
            print("Data successfully removed")
            print("\n")
        else:
            print("List is empty")
            print("\n")

    elif ch == 6:
        if l.head:
            key = int(
                input("Enter the element to be removed before :"))
            print(l.removeBefore(key))
            print("\n")
        else:
            print("List is empty")
            print("\n")

    elif ch == 7:
        l.display()
    elif ch == 8:
        break
    else:
        print("please enter a valid choice")
```

```
if __name__ == '__main__':
    main()

'''
```

Output:

STACK

MENU

1.Stack

2.Queue

3.Linked List

4.Exit

Enter your choice: 1

Stack

1.Push

2.Pop

3.Display

4.Exit

Enter your choice: 1

Enter the number to be pushed :34

Data successfully pushed

1.Push

2.Pop

3.Display

4.Exit

Enter your choice: 1

Enter the number to be pushed :24

Data successfully pushed

1.Push

2.Pop

3.Display

4.Exit

Enter your choice: 3

Stack Content -> [24, 34]

Queue

1.Insert

2.Remove

3.Display

4.Exit

Enter your choice: 1

Enter the number to be inserted :56

Data successfully inserted

1.Insert

2.Remove

3.Display

4.Exit

Enter your choice: 1

Enter the number to be inserted :43

Data successfully inserted

1.Insert

2.Remove

3.Display

4.Exit

Enter your choice: 2

removed element is : Removed Element is 56

1.Insert

2.Remove

3.Display

4.Exit

Enter your choice: 3

Queue Content -> 43

Linked List

1.Insert at beginning

2.Insert at end

3.Insert after

4.Remove first element

5.Remove last element

6.Remove element before

7.Display

8.Exit

Enter your choice: 1

Enter the number to be inserted :23

Data successfully inserted

1.Insert at beginning

2.Insert at end

3.Insert after

4.Remove first element

5.Remove last element

6.Remove element before

7.Display

8.Exit

Enter your choice: 2

Enter the number to be inserted :45

Data successfully inserted

1.Insert at beginning

2.Insert at end

3.Insert after

4.Remove first element

5.Remove last element

6.Remove element before

7.Display

8.Exit

Enter your choice: 3

Enter the position :23

Enter the number to be inserted :24

Node inserted.

1.Insert at beginning

2.Insert at end

3.Insert after

4.Remove first element

5.Remove last element

6.Remove element before

7.Display

8.Exit

Enter your choice: 7

Linked List:

23

24

45

1.Insert at beginning

2.Insert at end

3.Insert after

4.Remove first element

5.Remove last element

6.Remove element before

7.Display

8.Exit

Enter your choice: 6

Enter the element to be removed before :24

Node Removed.

Conclusion: In this experiment we have successfully implemented Stack, Queue and Linked List in Python.

'''