EX: -9 Python program to implement linked list(single linked list)

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

1 usage
class Linkedlist:
    def __init__(self):
        self.head=None

    def Insert(self,x):
        newnode=Node(x)
        newnode.next=self.head
        self.head=newnode

    def Delete(self):
        temp=self.head
        if temp:
            self.head=temp.next
            temp.next=None
            print("Deleted Node ="+str(temp.data))
        else:
            print("list is empty,")

    def Display(self):
        temp=self.head
        if(temp):
            print("Linkedlist---> ", end="")
            while(temp):
                print(str(temp.data)+"  ",end="")
                temp=temp.next
```

```
        else:
            print("list is empty")
```

```python
def searchNode(self):
    flag=0
    x=int(input("Enter which node you wnt to search="))
    temp=self.head

    if(temp):

        while(temp):
            if x==temp.data:
                flag=1
                break
            temp=temp.next

        if (flag==1):
            print("element is found in list")
        else:
            print("element not found in list")
    else:
        print("list is empty")
```

OUTPUT: -

```
"C:\Program Files\Python312\python.exe" C:\Users\dell\Downloads\9.py
----------Menu--------
1. Insert node at begining
2. delete
3. Display
4. search
enter your choice=1
Enter value for node=100
----------Menu--------
1. Insert node at begining
2. delete
3. Display
4. search
enter your choice=1
Enter value for node=50
----------Menu--------
1. Insert node at begining
2. delete
3. Display
4. search
enter your choice=1
Enter value for node=30
----------Menu--------
1. Insert node at begining
2. delete
3. Display
4. search
enter your choice=2
Deleted Node =30
```

```
----------Menu--------
1. Insert node at begining
2. delete
3. Display
4. search
enter your choice=3
Linkedlist---> 50  100  ----------Menu--------
1. Insert node at begining
2. delete
3. Display
4. search
enter your choice=4
Enter which node you wnt to search=50
element is found in list
```

EX: -10 Python program to implement stack data structure

```python
# program to impliment stack list

ch=0
1 usage
def createStack():
    STACK=[]
    return STACK

1 usage
def Push(x):
    STACK.append(x)
    print("Element pushed into stack=",x)

1 usage
def Pop():
    if len(STACK)==0:
        print("stack is empty")
    else:
        print("poped element from stack=",STACK.pop())


def Display():
    print("Stack")
    for i in range(len(STACK)-1,-1,-1):
        print("   ",STACK[i])
```

```python
# ---------main--------------------
STACK=createStack();

while ch <4:
    print("--------Stack operation----------")
    print("1 Push")
    print("2 Pop  ")
    print(" ------------------------------ ")

    ch=int(input("enter your choice="))
    if ch==1:
        x=int(input("enter value for push="))
        Push(x);
    elif ch==2:
        Pop()

    print("STACK ELEMENTS")
    Display()
```

OUTPUT: -

```
--------Stack operation----------
1 Push
2 Pop
 --------------------------------
enter your choice=1
enter value for push=10
Element pushed into stack= 10
STACK ELEMENTS
Stack
    10
--------Stack operation----------
1 Push
2 Pop
 --------------------------------
enter your choice=1
enter value for push=20
Element pushed into stack= 20
STACK ELEMENTS
Stack
    20
    10
--------Stack operation----------
1 Push
2 Pop
 --------------------------------
enter your choice=1
enter value for push=30
Element pushed into stack= 30
```

```
STACK ELEMENTS
Stack
    30
    20
    10
--------Stack operation----------
1 Push
2 Pop
 --------------------------------
enter your choice=2
poped element from stack= 30
STACK ELEMENTS
Stack
    20
    10
--------Stack operation----------
1 Push
2 Pop
 --------------------------------
enter your choice=2
poped element from stack= 20
STACK ELEMENTS
Stack
    10
```

EX: -12 Python program to implement factorialof the number

```
def fact(n):
    if n==1:
        return 1
    else:
        return n*fact(n-1)

a=int(input("Enter a number="))
print("\n\nFactorial=",fact(a))
```

OUTPUT: -

```
Enter a number=5



Factorial= 120

Process finished with exit code 0
```

EX: -13 Python program to implement bracket matching

```python
#write python program to implement bracket matching using stack
1 usage
def bracketsbalanced(expr):
    STACK=[]
    for char in expr:
        if char in['(','{','[']:
            STACK.append(char)
        else:
            if not STACK:
                return False
            current_char=STACK.pop()
            if current_char=='(':
                if char!=')':
                    return False

            if current_char=='{':
                if char!='}':
                    return False

            if current_char=='[':
                if char!=']':
                    return False
    if STACK:
        return False
    return True
```

```python
if __name__=="__main__":
    expr=input("Enter expression = ")
    if bracketsbalanced(expr):
        print("balanced")
    else:
        print("not balanced")
```

OUTPUT: -

```
"C:\Program Files\Python312\python.exe" "C:\Users\dell\Downloads\program 12.py"
Enter expression = {()[}]
not balanced
```

```
Enter expression = []()
balanced
```

EX: -14 Python program to implement Tower of hanoi

```
def towerofhanoi(n, rodA, rodB, rodC):
    if n==0:
        return
    towerofhanoi(n-1, rodA, rodC,rodB)
    print("move disk", n,"rodA", rodA,"rodB", rodB)
    towerofhanoi(n-1,rodC,rodB,rodA)
n=int(input("enter number of disks:"))
towerofhanoi(n, rodA: 'P', rodB: 'Q', rodC: 'R')
```

OUTPUT: -

```
enter number of disks:3
move disk 1 rodA P rodB Q
move disk 2 rodA P rodB R
move disk 1 rodA Q rodB R
move disk 3 rodA P rodB Q
move disk 1 rodA R rodB P
move disk 2 rodA R rodB Q
move disk 1 rodA P rodB Q

Process finished with exit code 0
```

EX: -15 Python program to implement queue data structure

```
def create_Queue():
    Queue=[]
    return Queue

def insert(Queue,item):
    Queue.append(item)

def Delete(Queue):
    if(len(Queue)==0):
        return "Queue is empty"
    else:
        return (Queue.pop(0))

1 usage
def selectOption(ch):
    if ch==1:
        x=int(input("Enter the item to insert:"))
        insert(Queue,str(x))
    elif ch==2:
        print("Delete item:",Delete(Queue))
Queue=create_Queue()
ch=0
while ch<3:
    print("1 insert")
    print("2 delete")
    print("3 exit")
    ch=int(input("Enter your choice="))
    selectOption(ch)
    print(Queue)
```

OUTPUT: -

```
"C:\Program Files\Python312\python.exe" C:\Users\dell\Downloads\program15.py
1 insert
2 delete
3 exit
Enter your choice=1
Enter the item to insert:20
['20']
1 insert
2 delete
3 exit
Enter your choice=1
Enter the item to insert:30
['20', '30']
1 insert
2 delete
3 exit
Enter your choice=1
Enter the item to insert:10
['20', '30', '10']
1 insert
2 delete
3 exit
Enter your choice=2
Delete item: 20
['30', '10']
```

```
1 insert
2 delete
3 exit
Enter your choice=2
Delete item: 30
['10']
1 insert
2 delete
3 exit
Enter your choice=3
['10']

Process finished with exit code 0
```

EX: -16 Python program to implement Priority queue

```python
class PriorityQueue(object):
    def __init__(self):
        self.queue=[]

    def __str__(self):
        return '  '.join([str(i) for i in self.queue])

    def isEmpty(self):
        return len(self.queue)==0

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

    def delete(self):



            max=0
            for i in range(len(self.queue)):
                if self.queue[i]>self.queue[max]:
                    max=i

            x=self.queue[max]
            del self.queue[max]
            return x
```

```python
if __name__ == '__main__':
    myQueue = PriorityQueue()
    myQueue.insert(100)
    myQueue.insert(13)
    myQueue.insert(90)
    myQueue.insert(65)
    myQueue.insert(22)
    myQueue.insert(95)
    print("Queue Elements  = ", myQueue)

    print("Delete from Queue = ")
    while not myQueue.isEmpty():
        print(myQueue.delete())
```

OUTPUT: -

```
"C:\Program Files\Python312\python.exe" C:\Users\dell\Downloads\program16.py
Queue Elements  =  100  13  90  65  22  95
Delete from Queue =
100
95
90
65
22
13

Process finished with exit code 0
```