

Python

Om Ghanshyam Bhamare

Div: A Roll No: 3

SE

Experiment 5

Aim: Menu driven program for data structure using built in function for queue, dequeue, stack and linked list

Code:

```
#Om Ghanshyam Bhamare

#Div A Roll No: 3

#AIM: Menu driven program for data structure using built in function for
#queue, dequeue, stack and Linked List.

import collections

def repetStack(lst):

    print("\n\nSelect Stack Operations: ")

    print("\n1]Add Element\n2]Remove Element\n3]Search\n4]Peep\n5]Show")

    choiceStack=int(input("Enter Choice: "))

    if choiceStack==1:

        addStack(lst)

    elif choiceStack==2:

        removeStack(lst)

    elif choiceStack==3:

        serachStack(lst)
```

```

elif choiceStack==4:

    peepStack(lst)

elif choiceStack==5:

    showStack(lst)

else:

    print("Invalid choice")

    repetStack(lst)


def addStack(lst):

    num = int(input("\nEnter Element added to stack: "))

    lst.append(num)

    repetStack(lst)


def removeStack(lst):

    lst.pop()

    repetStack(lst)


def serachStack(lst):

    num=int(input("\nEnter Number to search: "))

    if num in range(len(lst)):

        print("Number is present at index: {}".format(lst.index(num)))

    else:

        print("Number is not present")

```

```
repetStack(lst)
```

```
def peepStack(lst):
```

```
    print("Element at top : ",lst[-1])
```

```
    repetStack(lst)
```

```
def showStack(lst):
```

```
    print("Elements in Stack : ",lst)
```

```
    repetStack(lst)
```

```
def stackOp():
```

```
    print("\nStack Operations")
```

```
    lst=[]
```

```
    size=int(input("Enter Size of Stack: "))
```

```
    print("Enter Element in Stack")
```

```
    for i in range(0,size):
```

```
        a=int(input())
```

```
        lst.append(a)
```

```
    print("Element in stack: ",lst)
```

```
    repetStack(lst)
```

```
#Queue Operations....
```

```

def repetQueue(lst):

    print("\n\nSelect Queue Operations: ")

    print("\n1]Add Element\n2]Remove Element\n3]Search\n4]Show")

    choiceQueue=int(input("Enter Choice: "))

    if choiceQueue==1:

        addEndQueue(lst)

    elif choiceQueue==2:

        removeQueue(lst)

    elif choiceQueue==3:

        serachQueue(lst)

    elif choiceQueue==4:

        showQueue(lst)

    else:

        print("Invalid choice")

        repetQueue(lst)

def addEndQueue(lst):

    num=int(input("\nEnter element to enter to Queue: "))

    lst.append(num)

    repetQueue(lst)

def removeQueue(lst):

    if lst==[]:

        print("\nQueue is empty you can not remove elemnt....")

```

```

    else:

        lst.pop(0)

    repetQueue(lst)

def serachQueue(lst):

    num=int(input("\nEnter Number to search: "))

    if num in range(len(lst)):

        print("\nNumber is present at index: {}".format(lst.index(num)))

    else:

        print("\nNumber is not present")

    repetQueue(lst)

    pass

def showQueue(lst):

    print("\nElements in Queue are: ",lst)

    repetQueue(lst)

    pass

def queueOp():

    print("\nQueueOperations")

    lst=[]

    size=int(input("Enter Size of Queue: "))

    print("Enter Element in Queue")

    for i in range(0,size):

```

```

        a=int(input())

        lst.append(a)

    print("Element in Queue: ",lst)

    repetQueue(lst)

#Deque Operations...

def repetDequeue(list):

    print("\n\nSelect Deque Operations: ")

    print("\n1]Add Element at Front \n2]Add Element at End \n3]Remove Element at F
ront\n4]Remove Element at End\n5]Search\n6]Show")

    choiceDeQueue=int(input("Enter Choice: "))

    if choiceDeQueue==1:

        addAtFrontDequeue(list)

    elif choiceDeQueue==2:

        addAtLastDequeue(list)

    elif choiceDeQueue==3:

        removeAtFrontDequeue(list)

    elif choiceDeQueue==4:

        removeAtEndDequeue(list)

    elif choiceDeQueue==5:

        serachDequeue(list)

    elif choiceDeQueue==6:

```

```
        showDequeue(list)

    else:

        print("Invalid choice")

        repetQueue(list)

def addAtFrontDequeue(list):

    collist=collections.deque(list)

    num=int(input("Enter Element add to Front"))

    collist.appendleft(num)

    list=collist

    repetDequeue(list)

    pass

def addAtLastDequeue(list):

    collist=collections.deque(list)

    num=int(input("Enter Element add to Last"))

    collist.append(num)

    list=collist

    repetDequeue(list)

    pass

def removeAtFrontDequeue(list):

    if list==[]:

        print("Dequeue is empty: ")
```

else:

collist=collections.deque(list)

collist.popleft()

list=collist

repetDequeue(list)

pass

def removeAtEndDequeue(list):

if list==[]:

print("Dequeue is empty: ")

else:

collist=collections.deque(list)

collist.pop()

list=collist

repetDequeue(list)

pass

def serachDequeue(list):

collist=collections.deque(list)

num=int(input("\nEnter Number to search: "))

if num *in* collist:

print("Number is present at index: {0}".format(list.index(num)))

else:

print("Number is not present")


```
list=collist

repetDequeue(list)

pass


def showDequeue(list):

    print("\nElements in Queue are: ",list)

    repetDequeue(list)

    pass


def dequeueOp():

    print("\nDequeue Operations")

    list=[]

    size=int(input("Enter Size of Dequeue: "))

    print("Enter Element in Dequeue")

    for i in range(0,size):

        a=int(input())

        list.append(a)

    print("Element in Dequeue: ",list)

    repetDequeue(list)

    pass
```

```

def repetLL(lst):

    print("\n\nSelect Stack Operations: ")

    print("\n1]Add Element\n2]Add Element at Given Position\n3]Remove Element\n4]R
eplace Element\n5]Search\n6]Show")

    choiceLL=int(input("Enter Choice: "))

    if choiceLL==1:

        addLL(lst)

    elif choiceLL==2:

        addAnyLL(lst)

    elif choiceLL==3:

        removeLL(lst)

    elif choiceLL==4:

        replaceLL(lst)

    elif choiceLL==5:

        searchLL(lst)

    elif choiceLL==6:

        showLL(lst)

    else:

        print("Invalid choice")

        repetLL(lst)

```

```
def addLL(lst):

    num=int(input("Enter Element add to Linked lst: "))

    lst.append(num)

    repetLL(lst)

def addAnyLL(lst):

    num1=int(input("Enter Index where to add element: "))

    num2=int(input("Enter element to add Linked lst: "))

    lst.insert(num1,num2)

    repetLL(lst)

def removeLL(lst):

    num=int(input("Enter Element to remove: "))

    lst.remove(num)

    repetLL(lst)

def replaceLL(lst):

    num1=int(input("Enter Element to Old value "))

    index=lst.index(num1)

    num2=int(input("Enter Element to New value "))

    lst.insert(index,num2)

    lst.remove(num1)

    repetLL(lst)
```

```

def searchLL(lst):

    num=int(input("\nEnter Number to search: "))

    if num in range(len(lst)):

        print("Number is present at index: {}".format(lst.index(num)))

    else:

        print("Number is not present")

    repetLL(lst)

def showLL(lst):

    print("Linked lst: ",lst)

    repetLL(lst)

def linkedLstOp():

    print("\nLinked lst Operations")

    lst=[]

    size=int(input("Enter Linked lst: "))

    print("Enter Element in Linked lst")

    for i in range(0,size):

        a=int(input())

        lst.append(a)

    print("Element in Linked lst: ",lst)

    repetLL(lst)

pass

```

```
print("Select any from following: \n")

print("1]Stack\n2]Queue\n3]Deque\n4]Linked lst\n")

choice = int(input("Enter Your choice: "))

if choice==1:
    stackOp()

elif choice==2:
    queueOp()

elif choice==3:
    dequeueOp()

elif choice==4:
    linkedlstOp()

else:
    print("Plz Enter Proper Choice")
```

Output:

```
sk/dataStructure.py
C:\Users\DELL>python c:/Users/DELL/Desktop/De
sk/dataStructure.py
Select any from following:

1]Stack
2]Queue
3]Deque
4]Linked list

Enter Your choice: 1

Stack Operations
Enter Size of Stack: 5
Enter Element in Stack
1
2
3
4
5
Element in stack: [1, 2, 3, 4, 5]

Select Stack Operations:

1]Add Element
2]Remove Element
3]Search
4]Peep
5]Show
Enter Choice: 1

Enter Element added to stack: 12
```

Select Stack Operations:

- 1]Add Element
- 2]Remove Element
- 3]Search
- 4]Peep
- 5]Show

Enter Choice: 5

Elements in Stack : [1, 2, 3, 4, 5, 12]

Select Stack Operations:

- 1]Add Element
- 2]Remove Element
- 3]Search
- 4]Peep
- 5]Show

Enter Choice: 2

Select Stack Operations:

- 1]Add Element
- 2]Remove Element
- 3]Search
- 4]Peep
- 5]Show

Enter Choice: 2

Select Stack Operations:

- 1]Add Element
- 2]Remove Element
- 3]Search
- 4]Peep

```
Enter Choice: 5
Elements in Stack : [1, 2, 3, 4]
```

```
Select Stack Operations:
```

```
1]Add Element
2]Remove Element
3]Search
4]Peep
5]Show
Enter Choice: 3
```

```
Enter Number to search: 12
Number is not present
```

```
Select Stack Operations:
```

```
1]Add Element
2]Remove Element
3]Search
4]Peep
5]Show
Enter Choice: 3
```

```
Enter Number to search: 2
Number is present at index: 1
```

```
Select Stack Operations:
```

```
1]Add Element
2]Remove Element
3]Search
4]Peep
5]Show
Enter Choice: 4
```



```
Enter Choice: 4
Element at top : 4

Select Stack Operations:

1]Add Element
2]Remove Element
3]Search
4]Peep
5]Show
Enter Choice: 5
Elements in Stack : [1, 2, 3, 4]
```

```
Select Stack Operations:

1]Add Element
2]Remove Element
3]Search
4]Peep
5]Show
Enter Choice: 
```

```
C:\Users\DELL>python c:/Users/DELL/Desktop/Desk/dataStructure.py
Select any from following:
```

```
1]Stack
2]Queue
3]Deque
4]Linked list
```

```
Enter Your choice: 2
```

```
QueueOperations
```

```
Enter Size of Queue: 5
```

```
Enter Element in Queue
```

```
5
```

```
4
```

```
3
```

```
2
```

```
1
```

```
Element in Queue: [5, 4, 3, 2, 1]
```

```
Select Queue Operations:
```

```
1]Add Element
```

```
2]Remove Element
```

```
3]Search
```

```
4]Show
```

```
Enter Choice: 1
```

```
Enter element to enter to Queue: 10
```

```
Select Queue Operations:
```

```
1]Add Element
```

```
2]Remove Element
```

```
2]Remove Element
3]Search
4]Show
Enter Choice: 4
```

Elements in Queue are: [5, 4, 3, 2, 1, 10]

Select Queue Operations:

```
1]Add Element
2]Remove Element
3]Search
4]Show
Enter Choice: 2
```

Select Queue Operations:

```
1]Add Element
2]Remove Element
3]Search
4]Show
Enter Choice: 2
```

Select Queue Operations:

```
1]Add Element
2]Remove Element
3]Search
4]Show
Enter Choice: 4
```

Elements in Queue are: [3, 2, 1, 10]

Select Queue Operations:

- 1]Add Element
- 2]Remove Element
- 3]Search
- 4]Show

Enter Choice: 3

Enter Number to search: 12

Number is not present

Select Queue Operations:

- 1]Add Element
- 2]Remove Element
- 3]Search
- 4]Show

Enter Choice: 3

Enter Number to search: 1

Number is present at index: 2

Select Queue Operations:

- 1]Add Element
- 2]Remove Element
- 3]Search
- 4]Show

Enter Choice:

```
C:\Users\DELL>python c:/Users/DELL/Desktop/Desk/dataStructure.py
Select any from following:
```

- 1]Stack
- 2]Queue
- 3]Deque
- 4]Linked list

```
Enter Your choice: 3
```

```
Deque Operations
```

```
Enter Size of Dequeue: 5
```

```
Enter Element in Dequeue
```

```
2
```

```
4
```

```
6
```

```
8
```

```
10
```

```
Element in Dequeue: [2, 4, 6, 8, 10]
```

```
Select Deque Operations:
```

```
1]Add Element at Front
```

```
2]Add Element at End
```

```
3]Remove Element at Front
```

```
4]Remove Element at End
```

```
5]Search
```

```
6]Show
```

```
Enter Choice: 1
```

```
Enter Element add to Front10
```

```
Select Deque Operations:
```

```
1]Add Element at Front
2]Add Element at End
3]Remove Element at Front
4]Remove Element at End
5]Search
6]Show
```

Enter Choice: 6

Elements in Queue are: deque([10, 2, 4, 6, 8, 10])

Select Deque Operations:

```
1]Add Element at Front
2]Add Element at End
3]Remove Element at Front
4]Remove Element at End
5]Search
6]Show
```

Enter Choice: 2

Enter Element add to Last12

Select Deque Operations:

```
1]Add Element at Front
2]Add Element at End
3]Remove Element at Front
4]Remove Element at End
5]Search
6]Show
```

Enter Choice: 6

Elements in Queue are: deque([10, 2, 4, 6, 8, 10, 12])

Select Deque Operations:

```
1]Add Element at Front
2]Add Element at End
3]Remove Element at Front
4]Remove Element at End
5]Search
6]Show
Enter Choice: 3
```

Select Deque Operations:

```
1]Add Element at Front
2]Add Element at End
3]Remove Element at Front
4]Remove Element at End
5]Search
6]Show
Enter Choice: 3
```

Select Deque Operations:

```
1]Add Element at Front
2]Add Element at End
3]Remove Element at Front
4]Remove Element at End
5]Search
6]Show
Enter Choice: 6
```

Elements in Queue are: deque([4, 6, 8, 10, 12])

```
1]Add Element at Front
2]Add Element at End
3]Remove Element at Front
4]Remove Element at End
5]Search
6]Show
Enter Choice: 4
```

Select Deque Operations:

```
1]Add Element at Front
2]Add Element at End
3]Remove Element at Front
4]Remove Element at End
5]Search
6]Show
Enter Choice: 6
```

Elements in Queue are: deque([4, 6, 8, 10])

Select Deque Operations:

```
1]Add Element at Front
2]Add Element at End
3]Remove Element at Front
4]Remove Element at End
5]Search
6]Show
Enter Choice: 5
```

```
Enter Number to search: 12
Number is not present
```

Select Deque Operations:

- 1]Add Element at Front
- 2]Add Element at End
- 3]Remove Element at Front
- 4]Remove Element at End
- 5]Search
- 6]Show

Enter Choice: 5

Enter Number to search: 6

Number is present at index: 1

Select Deque Operations:

- 1]Add Element at Front
- 2]Add Element at End
- 3]Remove Element at Front
- 4]Remove Element at End
- 5]Search
- 6]Show

Enter Choice: █

```
C:\Users\DELL>python c:/Users/DELL/Desktop/Desk/dataStructure.py
```

Select any from following:

- 1]Stack
- 2]Queue
- 3]Deque
- 4]Linked lst

Enter Your choice: 4

Linked lst Operations

Enter Linked lst: 6

Enter Element in Linked lst

1

3

5

7

9

11

Element in Linked lst: [1, 3, 5, 7, 9, 11]

Select Stack Operations:

- 1]Add Element
- 2]Add Element at Given Position
- 3]Remove Element
- 4]Replace Element
- 5]Search
- 6]Show

Enter Choice: 1

Enter Element add to Linked lst: 13

Select Stack Operations:

1]Add Element
2]Add Element at Given Position
3]Remove Element
4]Replace Element
5]Search
6]Show
Enter Choice: 6
Linked lst: [1, 3, 5, 7, 9, 11, 13]

Select Stack Operations:

1]Add Element
2]Add Element at Given Position
3]Remove Element
4]Replace Element
5]Search
6]Show
Enter Choice: 2
Enter Index where to add element: 2
Enter element to add Linked lst: 15

Select Stack Operations:

1]Add Element
2]Add Element at Given Position
3]Remove Element
4]Replace Element
5]Search
6]Show
Enter Choice: 6
Linked lst: [1, 3, 15, 5, 7, 9, 11, 13]

Select Stack Operations:

1]Add Element
2]Add Element at Given Position
3]Remove Element
4]Replace Element
5]Search
6]Show
Enter Choice: 3
Enter Element to remove: 5

Select Stack Operations:

1]Add Element
2]Add Element at Given Position
3]Remove Element
4]Replace Element
5]Search
6]Show
Enter Choice: 6
Linked list: [1, 3, 15, 7, 9, 11, 13]

Select Stack Operations:

1]Add Element
2]Add Element at Given Position
3]Remove Element
4]Replace Element
5]Search
6]Show
Enter Choice: 4
Enter Element to Old value 9
Enter Element to New value 21

Select Stack Operations:

1]Add Element
2]Add Element at Given Position
3]Remove Element
4]Replace Element
5]Search
6]Show
Enter Choice: 6
Linked list: [1, 3, 15, 7, 21, 11, 13]

Select Stack Operations:

1]Add Element
2]Add Element at Given Position
3]Remove Element
4]Replace Element
5]Search
6]Show
Enter Choice: 5

Enter Number to search: 12
Number is not present

Select Stack Operations:

1]Add Element
2]Add Element at Given Position
3]Remove Element
4]Replace Element
5]Search
6]Show
Enter Choice: 5

Enter Number to search: 1
Number is present at index: 0

Select Stack Operations:

1]Add Element
2]Add Element at Given Position
3]Remove Element
4]Replace Element
5]Search
6]Show
Enter Choice: 6
Linked list: [1, 3, 15, 7, 21, 11, 13]