

Experiment 5

Aim: Write a menu driven program for data structure using built in function for link list, stack and queue.

THEORY:

- List the importance and working of each data structure

Ans.

- **Stacks:** Stacks are simple data structures which allow addition and removal of elements in a specific manner. Whenever there is an addition of some element, it is sent to the highest spot of the stacks. And the element that is on the topmost of the stack can be removed first. The concept is similar to a pile of items.

- **Queue:** Queue is another type of abstract data structure, or linear data structure. In this type of structure, when an element is to be added, it is done from the rear end, known as tail. When an element is to be deleted, it is done from the front end, known as head. It follows the concept of FIFO data structure, which means first in first out. The element that is entered first will be removed first as well. The process of making addition of elements to the data structure is known as Enqueue. The process of removing an element from the data structure is known as Dequeue.

- **Linked list** is a sequential structure that consists of a sequence of items in linear order which are linked to each other. Hence, you have to access data sequentially and random access is not possible. Linked lists provide a simple and flexible representation of dynamic sets.

- Explain addition, deletion, searching in each data structure.

Ans.

1. STACKS:

Follows Last In First Out (LIFO) operation.

Addition: Insert an element on to the top of the stack. This operation is called the push operation.

Deletion: Delete the topmost element and return it. This operation is called the pop operation.

Searching: Return the top element of the stack without deleting it. We can retrieve

only the top element of stack. This operation is called a Peek operation.

2. QUEUE:

Follows First In First Out (FIFO) operation.

Addition: Insert an element to the end of the queue. This operation is called the enqueue operation.

Deletion: Delete the element from the beginning of the queue. This operation is called the deque operation.

Searching: Return the element at the front position. This operation is called a Peek operation in a queue.

3. LINKED LIST:

Addition: Insert a key to the linked list. An insertion can be done in 3 different ways; insert at the beginning of the list, insert at the end of the list and insert in the middle of the list.

Deletion: Removes an element **x** from a given linked list. You cannot delete a node by a single step. A deletion can be done in 3 different ways; delete from the beginning of the list, delete from the end of the list and delete from the middle of the list.

Searching: Find the first element with the key **k** in the given linked list by a simple linear search and returns a pointer to this element

IMPLEMENTATION:

1. Menu driven program to implement stack.

```
stack = []
```

```
while True:
```

```
    choice = int(input("Menu: 1.Push 2.Pop 3.Peek 4.Exit
```

```
Enter Choice: "))
```

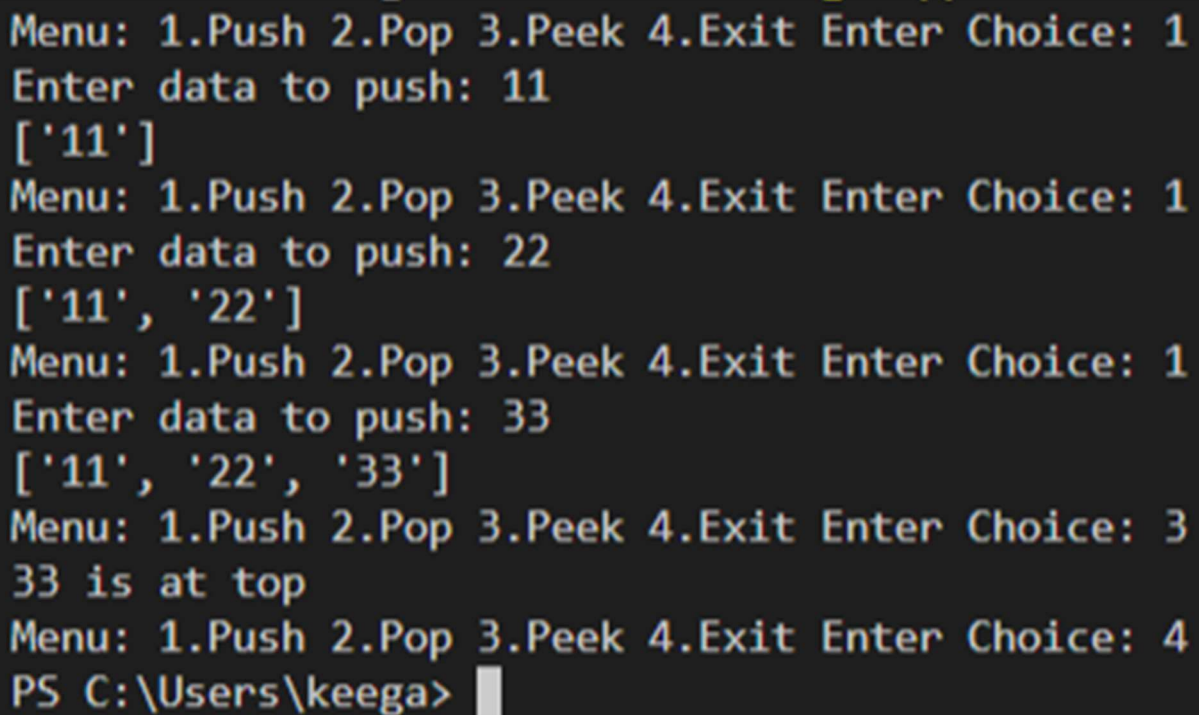
```
    if choice == 1:
```

```
        data = input("Enter data to push: ")
```

```
        stack.append(data)
```

```
        print(stack)
```

```
elif choice == 2:
    if stack == []:
        print("Underflow!! condition")
    else:
        stack.pop()
        print("Element removed!")
        print(stack)
elif choice == 3:
    if stack == []:
        print("Underflow!! condition")
    else:
        print(str(stack[-1]) + " is at top")
elif choice == 4:
    break
```



```
Menu: 1.Push 2.Pop 3.Peek 4.Exit Enter Choice: 1
Enter data to push: 11
['11']
Menu: 1.Push 2.Pop 3.Peek 4.Exit Enter Choice: 1
Enter data to push: 22
['11', '22']
Menu: 1.Push 2.Pop 3.Peek 4.Exit Enter Choice: 1
Enter data to push: 33
['11', '22', '33']
Menu: 1.Push 2.Pop 3.Peek 4.Exit Enter Choice: 3
33 is at top
Menu: 1.Push 2.Pop 3.Peek 4.Exit Enter Choice: 4
PS C:\Users\keega>
```

2. Menu driven program to implement queue

```
stack = []
```

```
while True:
```

```
    choice = int(input("Menu: 1.Enqueue 2.Dequeue 3.Peek 4.Exit
```

```
Enter Choice: "))
```

```
    if choice == 1:
```

```
        data = input("Enter data to enqueue: ")
```

```
        stack.append(data)
```

```
        print(stack)
```

```
    elif choice == 2:
```

```
        if stack == []:
```

```
            print("Underflow!! condition")
```

```
        else:
```

```
            stack.pop(0)
```

```
            print("Element removed!")
```

```
            print(stack)
```

```
    elif choice == 3:
```

```
        if stack == []:
```

```
            print("Underflow!! condition")
```

```
        else:
```

```
            print(str(stack[0]) + " is at front")
```

```
    elif choice == 4:
```

```
        break
```

```
Menu: 1.Enqueue 2.Dequeue 3.Peek 4.Exit
Enter Choice: 1
Enter data to enqueue: 11
['11']
Menu: 1.Enqueue 2.Dequeue 3.Peek 4.Exit
Enter Choice: 1
Enter data to enqueue: 22
['11', '22']
Menu: 1.Enqueue 2.Dequeue 3.Peek 4.Exit
Enter Choice: 1
Enter data to enqueue: 33
['11', '22', '33']
Menu: 1.Enqueue 2.Dequeue 3.Peek 4.Exit
Enter Choice: 2
Element removed!
['22', '33']
Menu: 1.Enqueue 2.Dequeue 3.Peek 4.Exit
Enter Choice: 3
22 is at front
Menu: 1.Enqueue 2.Dequeue 3.Peek 4.Exit
Enter Choice: 4
PS C:\Users\keega\Desktop\Keegan\SFIT\SEM 4\PP\Exp 4>
```

```
Menu: 1.Enqueue 2.Dequeue 3.Peek 4.Exit
Enter Choice: 1
Enter data to enqueue: 11
['11']
Menu: 1.Enqueue 2.Dequeue 3.Peek 4.Exit
Enter Choice: 2
Element removed!
[]
Menu: 1.Enqueue 2.Dequeue 3.Peek 4.Exit
Enter Choice: 2
Underflow!! condition
Menu: 1.Enqueue 2.Dequeue 3.Peek 4.Exit
Enter Choice: 3
Underflow!! condition
Menu: 1.Enqueue 2.Dequeue 3.Peek 4.Exit
Enter Choice: 4
PS C:\Users\keega\Desktop\Keegan\SFIT\SEM 4\PP\Exp 4>
```

3. Menu driven program for Linked List

```
import collections
```

```
linked_list = collections.deque()
```

```
while True:
```

```
    choice = int(input("Linked List Menu: 1.Insert 2.Delete 3.Search 4.Exit
```

```
Enter Choice:"))
```

```
    if choice == 1:
```

```
        pos = int(input("Enter the position where you want to insert: "))
```

```
        data = input("Enter data to insert: ")
```

```
        linked_list.insert(pos, data)
```

```
        print(linked_list)
```

```
elif choice == 2:
    if linked_list == []:
        print("Nothing in linked list to delete!")
    else:
        dlt = input("Which data to delete: ")
        if dlt in linked_list:
            linked_list.remove(dlt)
            print(linked_list)
        else:
            print("Element is not there!")
elif choice == 3:
    if linked_list == []:
        print("Nothing in linked list!")
    else:
        ser = input("Which data to search: ")
        if ser in linked_list:
            print(linked_list.index(ser, 0, len(linked_list)))
        else:
            print("Element is not there!")
elif choice == 4:
    break
```

Insert:

```
Linked List Menu: 1.Insert  2.Delete  3.Search  4.Exit
Enter Choice:1
Enter the position where you want to insert: 0
Enter data to insert: 11
deque(['11'])
Linked List Menu: 1.Insert  2.Delete  3.Search  4.Exit
Enter Choice:1
Enter the position where you want to insert: 1
Enter data to insert: 22
deque(['11', '22'])
Linked List Menu: 1.Insert  2.Delete  3.Search  4.Exit
Enter Choice:1
Enter the position where you want to insert: 3
Enter data to insert: 33
deque(['11', '22', '33'])
Linked List Menu: 1.Insert  2.Delete  3.Search  4.Exit
Enter Choice:1
Enter the position where you want to insert: 4
Enter data to insert: 44
deque(['11', '22', '33', '44'])
```

Delete:

```
Linked List Menu: 1.Insert  2.Delete  3.Search  4.Exit
Enter Choice:2
Which data to delete: 22
deque(['11', '33', '44'])
Linked List Menu: 1.Insert  2.Delete  3.Search  4.Exit
Enter Choice:2
Which data to delete: 44
deque(['11', '33'])
```


Academic Year: **2020-21**

Subject: **Skill Base Lab Course: Python Programming**

Class/ Branch/ Sem: **SE CMPN B**

PID :192120

Roll No.: 42

Search:

```
Linked List Menu: 1.Insert  2.Delete  3.Search  4.Exit
Enter Choice:3
Which data to search: 11
0
Linked List Menu: 1.Insert  2.Delete  3.Search  4.Exit
Enter Choice:3
Which data to search: 33
1
Linked List Menu: 1.Insert  2.Delete  3.Search  4.Exit
Enter Choice:3
Which data to search: 40
Element is not there!
```

Conclusion:

Successfully learnt to read and write in files using Python features.