



# DATA STRUCTURES

---

## LAB FILE

**SUBMITTED BY - VANSH RANA**

**ROLL NO - 2401420011**

**PROGRAM - B.Tech CSE (Data Science)**

**SEMESTER - III**

**SUBMITTED TO -**

## INDEX

S. NO.	Experiment Title	Page No.
1.	Browser History Navigation System (Using Stack Concept)	
2.	Ticketing System Using Queue (Linear Queue Implementation)	
3.	Singly Linked List Operations (Insert, Delete, Search, Display)	
4.	Circular Singly Linked List (Insert, Search, Delete, Display)	
5.	Reverse a String Using Stack	
6.	Check Balanced Parentheses Using Stack	
7.	Lab Project: Inventory Stock Management System	

# EXPERIMENT 1

## Browser History Navigation System (Using Stack Concept)

CODE:-

The screenshot shows a code editor interface with the following details:

- Title Bar:** Shows the path "DSA\_Lab\_Programs" and the file name "browser\_history.py".
- Explorer Panel:** Displays a folder structure under "DSA\_LAB\_PROGRAMS" containing "browser\_history.py".
- Code Editor:** The main area contains Python code for a browser history system using stacks. The code defines functions for visiting pages, going back, and going forward.

```
# Browser History using Stack
back_stack = []
forward_stack = []
current_page = None

def visit(page):
    global current_page
    if current_page:
        back_stack.append(current_page)
    current_page = page
    forward_stack.clear()
    print("Visited:", page)

def go_back():
    global current_page
    if not back_stack:
        print("No page to go back.")
        return
    forward_stack.append(current_page)
    current_page = back_stack.pop()
    print("Back to:", current_page)

def go_forward():
    global current_page
    if not forward_stack:
        print("No forward page.")
        return
    back_stack.append(current_page)
    current_page = forward_stack.pop()
    print("Forward to:", current_page)

# Demo
visit("Google")
visit("YouTube")
visit("Instagram")
go_back()
go_back()
go_forward()
```

- Status Bar:** Shows "Ln 40, Col 1" and other status indicators like "Spaces: 4", "UTF-8", "LF", "Python", "Finish Setup", "Python 3.13.7", and "Go Live".

## OUTPUT:-

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

(base) vanshrana@Vanshs-MacBook-Air DSA_Lab_Programs % python3 filename.py
ms/filename.py': [Errno 2] No such file or directory
● (base) vanshrana@Vanshs-MacBook-Air DSA_Lab_Programs % python3 browser_history.py

Visited: Google
Visited: YouTube
Visited: Instagram
Back to: YouTube
Back to: Google
Forward to: YouTube
○ (base) vanshrana@Vanshs-MacBook-Air DSA_Lab_Programs %
```

## EXPERIMENT 2

### Ticketing System Using Queue (Linear Queue Implementation)

## CODE:-

```
DSA_Lab_Programs
ticket_system.py > ...
1 # Ticket system using queue
2
3 queue = []
4
5 def issue_ticket(name):
6     queue.append(name)
7     print("Ticket issued to:", name)
8
9 def serve_customer():
10    if queue:
11        print("Serving:", queue.pop(0))
12    else:
13        print("No one in queue")
14
15 def show_queue():
16    print("Current Queue:", queue)
17
18 issue_ticket("Amit")
19 issue_ticket("Riya")
20 issue_ticket("John")
21 show_queue()
22 serve_customer()
23 show_queue()
```

## OUTPUT:-

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

● (base) vanshrana@Vanshs-MacBook-Air DSA_Lab_Programs % python3 ticket_system.py

Ticket issued to: Amit
Ticket issued to: Riya
Ticket issued to: John
Current Queue: ['Amit', 'Riya', 'John']
Serving: Amit
Current Queue: ['Riya', 'John']
○ (base) vanshrana@Vanshs-MacBook-Air DSA_Lab_Programs % █
```

## EXPERIMENT 3

### **Ticketing System Using Queue (Linear Queue Implementation)**

## CODE:-

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER** sidebar: Shows a folder named "DSA\_LAB\_PROGRAMS" containing several Python files: balanced\_parentheses.py, browser\_history.py, circular\_linked\_list.py, inventory\_system.py, reverse\_string.py, singly\_linked\_list.py, and ticket\_system.py.
- Editor Area:** The active file is "singly\_linked\_list.py". The code implements a singly linked list with methods for insertion, deletion, search, and display.
- Bottom Status Bar:** Shows the current file path as "DSA\_Lab\_Programs", line 61, column 1, and other status indicators like "Spaces: 4", "UTF-8", and "Python".

```
class Node:  
    def __init__(self, data):  
        self.data = data  
        self.next = None  
  
class List:  
    def __init__(self):  
        self.head = None  
  
    def insert_first(self, data):  
        n = Node(data)  
        n.next = self.head  
        self.head = n  
  
    def delete_value(self, val):  
        temp = self.head  
        if temp and temp.data == val:  
            self.head = temp.next  
            return  
  
        prev = None  
        while temp and temp.data != val:  
            prev = temp  
            temp = temp.next  
        if temp:  
            prev.next = temp.next  
  
    def search(self, val):  
        t = self.head  
        while t:  
            if t.data == val:  
                return True  
            t = t.next  
        return False  
  
    def display(self):  
        t = self.head  
        while t:  
            print(t.data, end=" -> ")  
            t = t.next  
        print("None")  
  
lst = List()  
lst.insert_first(5)
```

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER** sidebar: Shows the same "DSA\_LAB\_PROGRAMS" folder as the first screenshot.
- Editor Area:** The "singly\_linked\_list.py" file now contains additional code at the bottom to demonstrate its functionality.
- Bottom Status Bar:** Shows the current file path as "DSA\_Lab\_Programs", line 51, column 1, and other status indicators.

```
class List:  
    def __init__(self):  
        print("None")  
  
    def insert_first(self, data):  
        lst = List()  
        lst.insert_first(5)  
        lst.insert_first(15)  
        lst.insert_first(25)  
        lst.display()  
        lst.delete_value(15)  
        lst.display()  
        print("Search 25:", lst.search(25))  
51
```

## OUTPUT:-

The screenshot shows the terminal window with the following details:

- Tab Bar:** PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), PORTS.
- Output Content:** The terminal shows the command "python3 singly\_linked\_list.py" being run, followed by the list's state and a search result.

```
PROBLEMS      OUTPUT      DEBUG CONSOLE      TERMINAL      PORTS  
● (base) vanshrana@Vanshs-MacBook-Air DSA_Lab_Programs % python3 singly_linked_list.py  
25 -> 15 -> 5 -> None  
25 -> 5 -> None  
Search 25: True  
○ (base) vanshrana@Vanshs-MacBook-Air DSA_Lab_Programs %
```

## **EXPERIMENT 4**

**Circular Singly Linked List (Insert, Search, Delete, Display)**

## CODE:-

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows a folder named "DSA\_LAB\_PROGRAMS" containing several Python files: balanced\_parentheses.py, browser\_history.py, circular\_linked\_list.py (selected), inventory\_system.py, reverse\_string.py, singly\_linked\_list.py, and ticket\_system.py.
- Code Editor:** Displays the content of the "circular\_linked\_list.py" file. The code defines a Circular linked list structure with methods for insertion, display, and a constructor. It also includes a main block for testing the class.
- Status Bar:** Shows the current line (Ln 39, Col 1), spaces used (Spaces: 4), encoding (UTF-8), and file type (Python).

## OUTPUT:-

```
● (base) vanshrana@Vanshs-MacBook-Air DSA_Lab_Programs % python3 circular_linked_list.py
100 -> 200 -> 300 -> (head)
○ (base) vanshrana@Vanshs-MacBook-Air DSA_Lab_Programs %
```

# EXPERIMENT 5

## Reverse a String Using Stack

## CODE:-

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar, which lists several Python files under a folder named 'DSA\_Lab\_Programs'. The file 'reverse\_string.py' is currently selected and its content is displayed in the main editor area. The code uses a stack (represented by a list) to reverse a given string.

```
reverse_string.py >...
1 def reverse(text):
2     st = []
3     for ch in text:
4         st.append(ch)
5     rev = ""
6     while st:
7         rev += st.pop()
8     return rev
9
10 print(reverse("welcome"))
11
```

## OUTPUT:-

```
● (base) vanshrana@Vanshs-MacBook-Air DSA_Lab_Programs % python3 reverse_string.py
emoclew
○ (base) vanshrana@Vanshs-MacBook-Air DSA_Lab_Programs %
Ln 11, Col 1  Spaces: 4  UTF-8  LF  {ø}
```

# EXPERIMENT 6

## Check Balanced Parentheses Using Stack

### CODE:-

```
balancing_parentheses.py > ...
1 def balanced(s):
2     st = []
3     pairs = {'(': ')', '[': ']', '{': '}'}
4
5     for ch in s:
6         if ch in "([{":
7             st.append(ch)
8         elif ch in ")]}":
9             if not st or st.pop() != pairs[ch]:
10                 return False
11     return len(st) == 0
12
13 print(balanced("(a+b)*(c+d)"))
14 print(balanced("[]"))
15
16
```

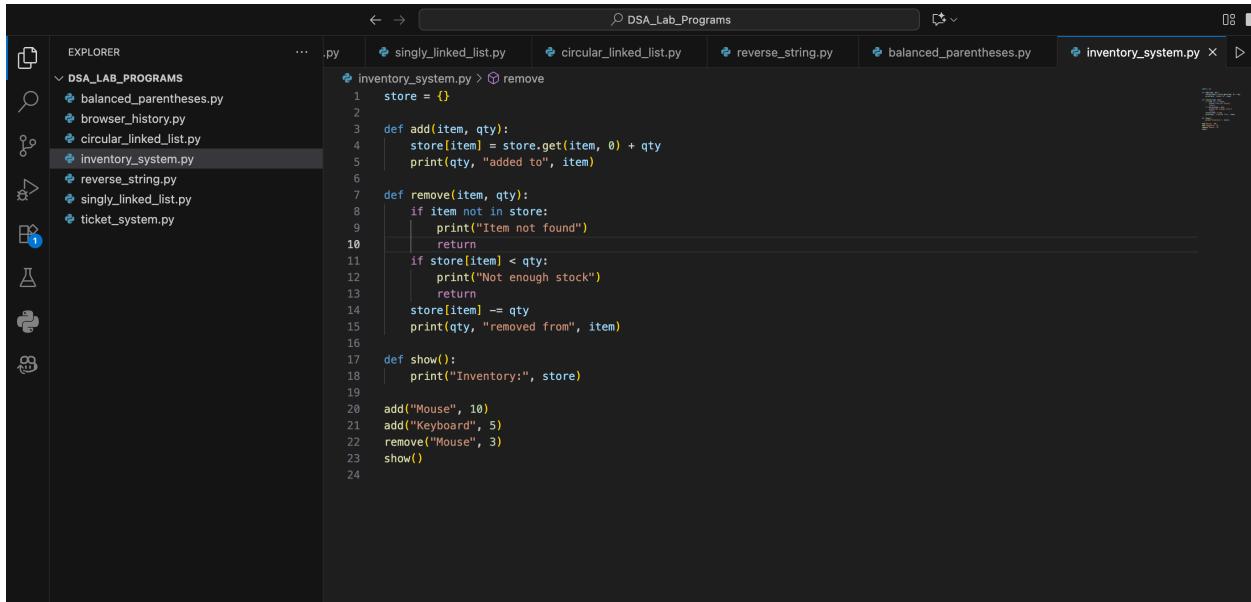
## OUTPUT:-

```
● (base) vanshrana@Vanshs-MacBook-Air DSA_Lab_Programs % python3 balanced_parentheses.py
True
False
○ (base) vanshrana@Vanshs-MacBook-Air DSA_Lab_Programs %
```

# EXPERIMENT 7

## Check Balanced Parentheses Using Stack

CODE:-



The screenshot shows a code editor interface with the title bar "DSA\_Lab\_Programs". The left sidebar (EXPLORER) lists several Python files: balanced\_parentheses.py, browser\_history.py, circular\_linked\_list.py, inventory\_system.py (which is selected), reverse\_string.py, singly\_linked\_list.py, and ticket\_system.py. The right pane shows the content of the selected file, inventory\_system.py:

```
store = {}

def add(item, qty):
    store[item] = store.get(item, 0) + qty
    print(qty, "added to", item)

def remove(item, qty):
    if item not in store:
        print("Item not found")
        return
    if store[item] < qty:
        print("Not enough stock")
        return
    store[item] -= qty
    print(qty, "removed from", item)

def show():
    print("Inventory:", store)

add("Mouse", 10)
add("Keyboard", 5)
remove("Mouse", 3)
show()
```

## OUTPUT:-

```
● (base) vanshrana@Vanshs-MacBook-Air DSA_Lab_Programs % python3 inventory_system.py
10 added to Mouse
5 added to Keyboard
3 removed from Mouse
Inventory: {'Mouse': 7, 'Keyboard': 5}
○ (base) vanshrana@Vanshs-MacBook-Air DSA_Lab_Programs % █
```

Ln 10, Col 15 Spaces: 4 UTF-8 L