BÀI TẬP THỰC HÀNH CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

LAB MANUAL

Academic Year	:	**2022 - 2023**
Semester	:	**III Semester**

Prepared by

MSc. KhangVQH

WEEK-7

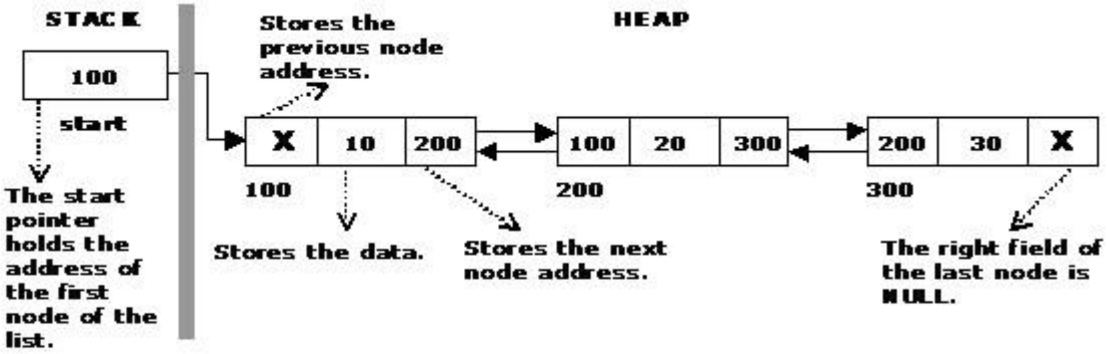
IMPLEMENTATION OF DOUBLE LINKED LIST

7.1 OBJECTIVE:

a. Create a doubly linked list of integers.
b. Delete a given integer from the above doubly linked list.
c. Display the contents of the above list after deletion

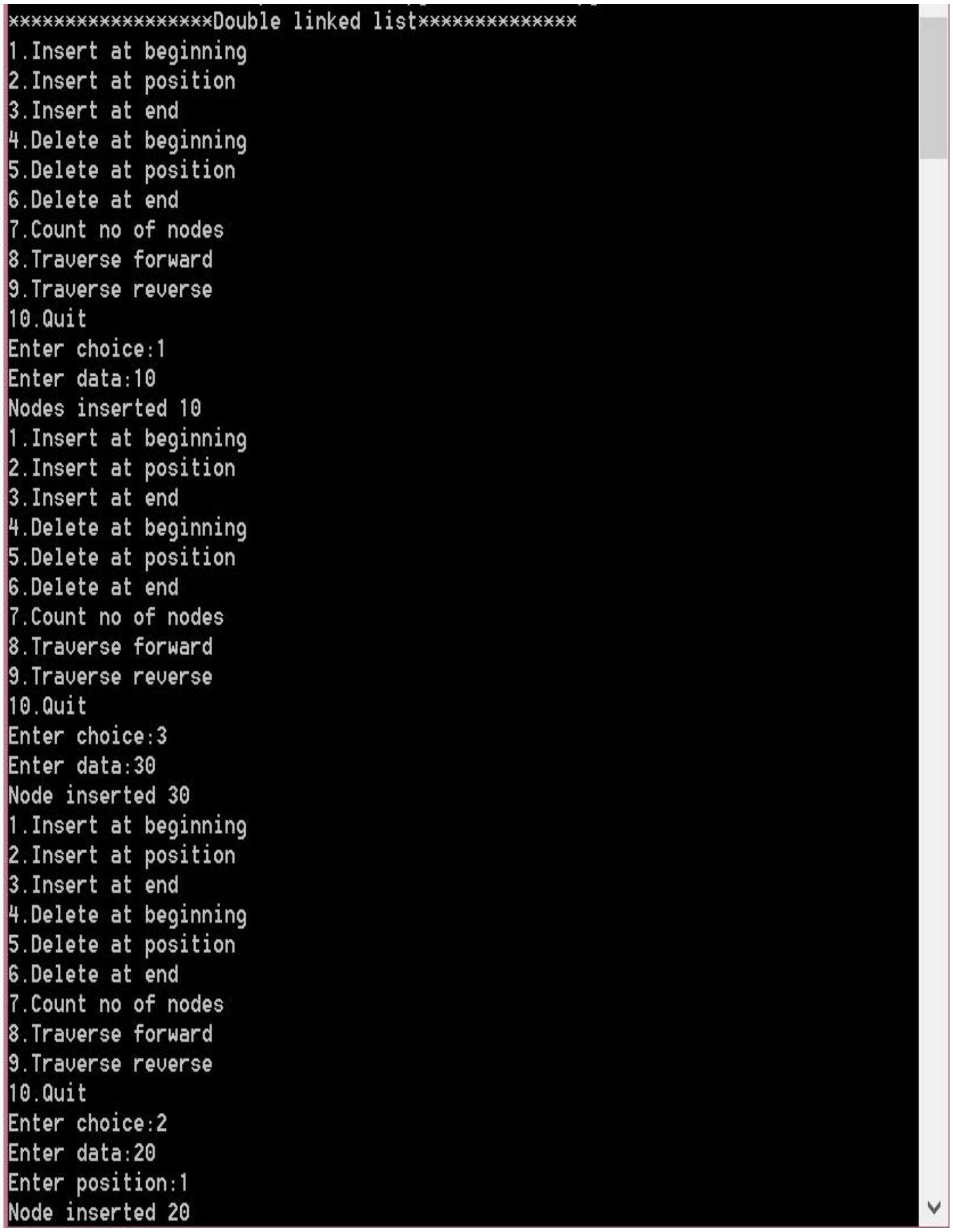
7.2 PROGRAM LOGIC:

1. In a doubly-linked list each node of the list contain two references (or links) – one to the previous node and other to the next node. The previous link of the first node and the next link of the last node points to NULL.
2. A double linked list is a two-way list in which all nodes will have two links. This helps in accessing both successor node and predecessor node from the given node position. It provides bi-directional traversing.
3. Each node contains three fields: Left link, Data and Right link.
4. The left link points to the predecessor node and the right link points to the successor node. The data field stores the required data.
5. The basic operations in a double linked list are: creation, insertion, deletion and traversing.
6. The beginning of the double linked list is stored in a "start" pointer which points to the first node. The first node's left link and last node's right link is set to NULL.

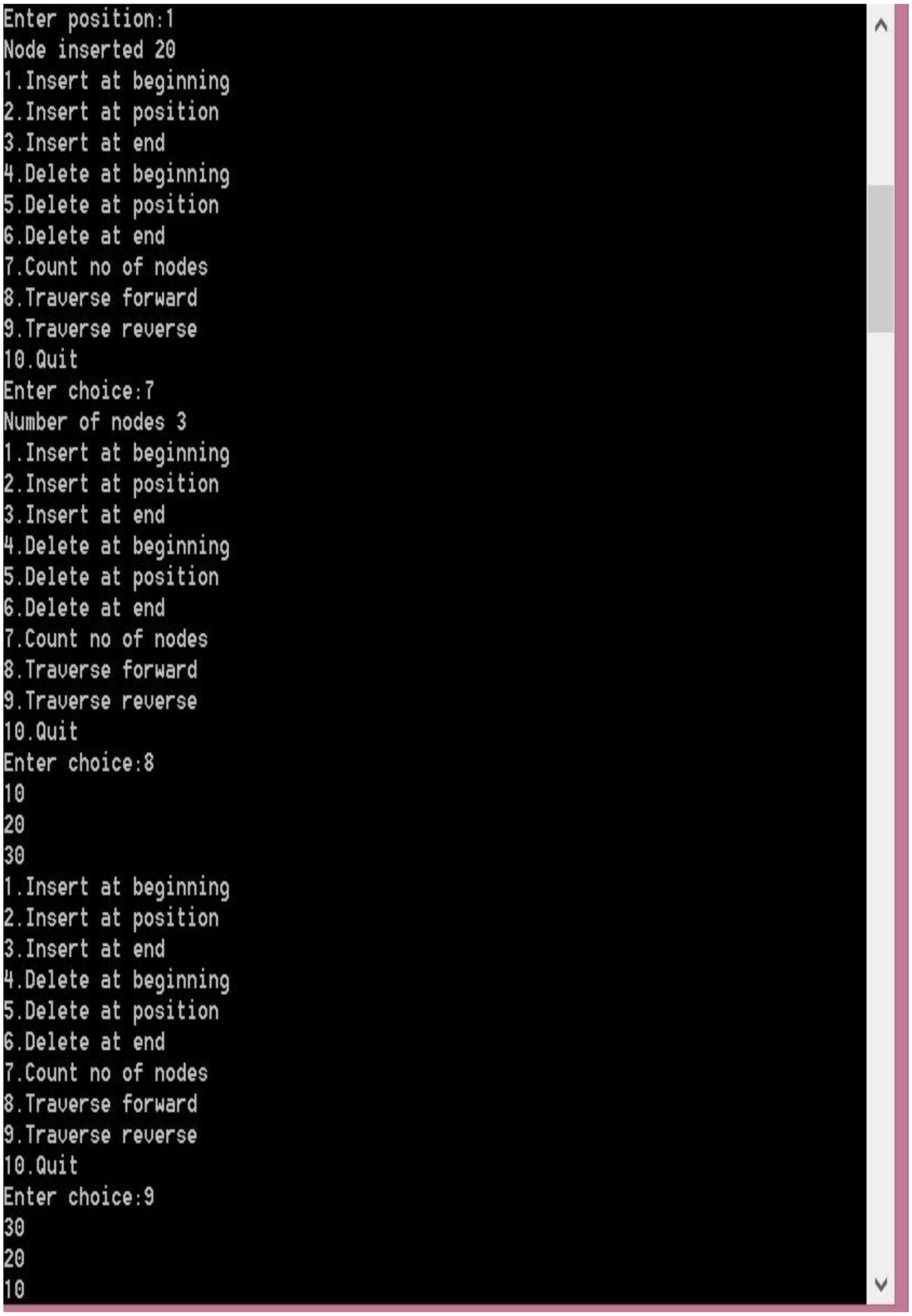

STACK
100
start
The start pointer holds the address of the first node of the list.
Stores the previous node address.
HEAP
X
10
200
100
20
300
200
30
X
100
200
300
Stores the data.
Stores the next node address.
The right field of the last node is NULL.

7.3 IMPLEMENTATION:

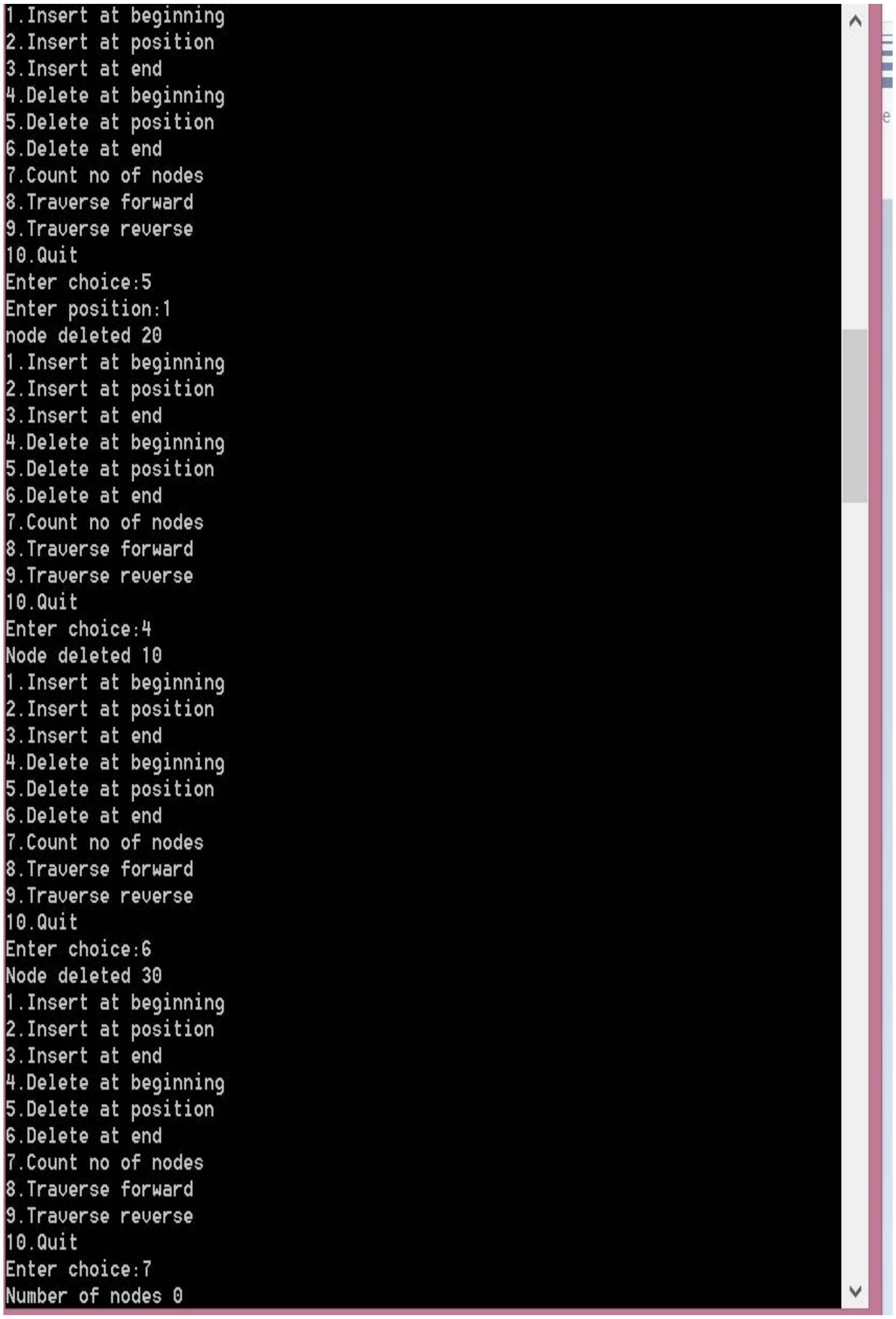
Output:



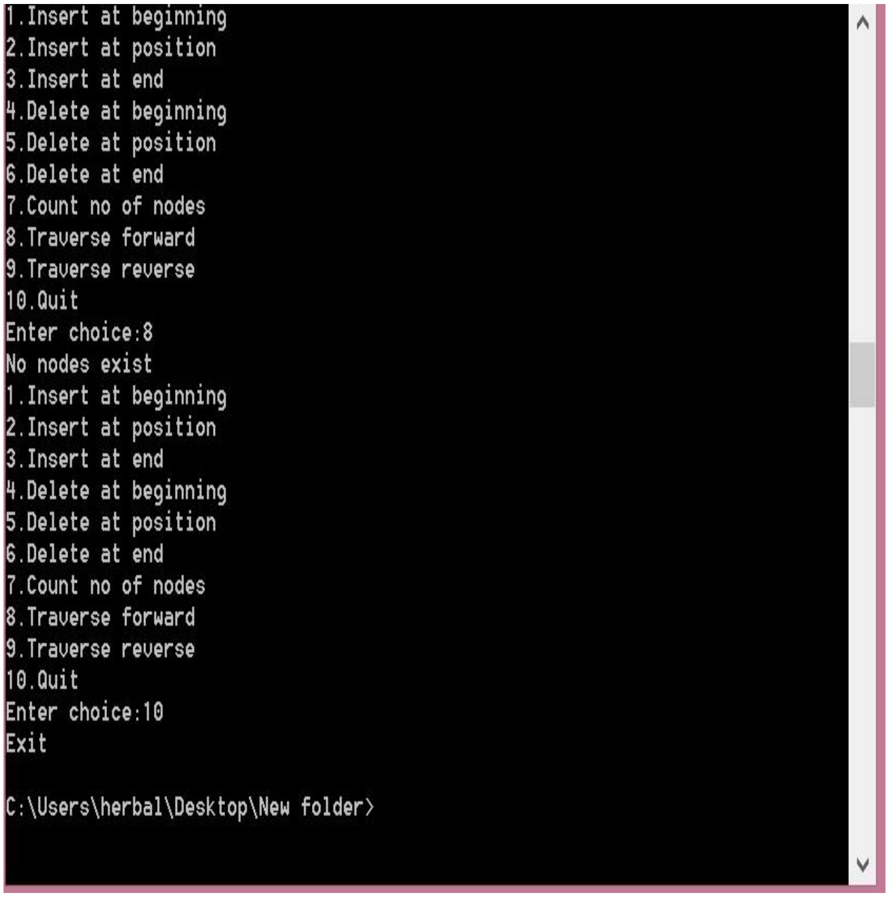
```
******************Double linked list**************
1.Insert at beginning
2.Insert at position
3.Insert at end
4.Delete at beginning
5.Delete at position
6.Delete at end
7.Count no of nodes
8.Traverse forward
9.Traverse reverse
10.Quit
Enter choice:1
Enter data:10
Nodes inserted 10
1.Insert at beginning
2.Insert at position
3.Insert at end
4.Delete at beginning
5.Delete at position
6.Delete at end
7.Count no of nodes
8.Traverse forward
9.Traverse reverse
10.Quit
Enter choice:3
Enter data:30
Node inserted 30
1.Insert at beginning
2.Insert at position
3.Insert at end
4.Delete at beginning
5.Delete at position
6.Delete at end
7.Count no of nodes
8.Traverse forward
9.Traverse reverse
10.Quit
Enter choice:2
Enter data:20
Enter position:1
Node inserted 20
```



```
Enter position:1
Node inserted 20
1.Insert at beginning
2.Insert at position
3.Insert at end
4.Delete at beginning
5.Delete at position
6.Delete at end
7.Count no of nodes
8.Traverse forward
9.Traverse reverse
10.Quit
Enter choice:7
Number of nodes 3
1.Insert at beginning
2.Insert at position
3.Insert at end
4.Delete at beginning
5.Delete at position
6.Delete at end
7.Count no of nodes
8.Traverse forward
9.Traverse reverse
10.Quit
Enter choice:8
10
20
30
1.Insert at beginning
2.Insert at position
3.Insert at end
4.Delete at beginning
5.Delete at position
6.Delete at end
7.Count no of nodes
8.Traverse forward
9.Traverse reverse
10.Quit
Enter choice:9
30
20
10
```



```
1.Insert at beginning
2.Insert at position
3.Insert at end
4.Delete at beginning
5.Delete at position
6.Delete at end
7.Count no of nodes
8.Traverse forward
9.Traverse reverse
10.Quit
Enter choice:5
Enter position:1
node deleted 20
1.Insert at beginning
2.Insert at position
3.Insert at end
4.Delete at beginning
5.Delete at position
6.Delete at end
7.Count no of nodes
8.Traverse forward
9.Traverse reverse
10.Quit
Enter choice:4
Node deleted 10
1.Insert at beginning
2.Insert at position
3.Insert at end
4.Delete at beginning
5.Delete at position
6.Delete at end
7.Count no of nodes
8.Traverse forward
9.Traverse reverse
10.Quit
Enter choice:6
Node deleted 30
1.Insert at beginning
2.Insert at position
3.Insert at end
4.Delete at beginning
5.Delete at position
6.Delete at end
7.Count no of nodes
8.Traverse forward
9.Traverse reverse
10.Quit
Enter choice:7
Number of nodes 0
```



```
1.Insert at beginning
2.Insert at position
3.Insert at end
4.Delete at beginning
5.Delete at position
6.Delete at end
7.Count no of nodes
8.Traverse forward
9.Traverse reverse
10.Quit
Enter choice:8
No nodes exist
1.Insert at beginning
2.Insert at position
3.Insert at end
4.Delete at beginning
5.Delete at position
6.Delete at end
7.Count no of nodes
8.Traverse forward
9.Traverse reverse
10.Quit
Enter choice:10
Exit

C:\Users\herbal\Desktop\New folder>
```

7.4 LAB ASSIGNMENT:

1. Write a program to insert a node at first , last and at specified position of double linked list?
2. Write a program to eliminate duplicates from double linked list?
3. Write a program to delete a node from first, last and at specified position of double linked list?

7.5 POST-LAB VIVA QUESTIONS:

1. How to represent double linked list?
2. How will you traverse double linked list?
3. List the advantages of double linked list over single list?