Department of Computer Science and Engineering,
Anna University, Chennai- 600025
CS23302 – Data Structures and Algorithms
Practical
July – November 2026
Year/Sem/Batch : II/III/ N

Exercise: 04	Stack Applications and Queue using Linked List	24 – Jul - 2026

Observation (5 Marks)

1. Why is a linked list preferred over an array for implementing a queue?
2. What is the purpose of the front and rear pointers?
3. What is the advantage of a linked-list queue over a static array queue?
4. What is the time complexity of the infix-to-postfix conversion algorithm?
5. Why do we implement two stacks in a single array?
6. What is the condition for stack overflow?
5. What is the condition for stack underflow?
6. How do you perform push and pop operations on each stack?
7. What is the advantage of using two stacks in one array?
8. How is memory utilization improved compared to using two separate arrays?
9. What is the time complexity of push and pop operations?
10. What happens if both stacks meet in the middle of the array?

Execution (15 Marks)

1. A customer service center receives customer requests throughout the day. Since requests must be handled in the order they are received (First-In-First-Out (FIFO)), a queue is the ideal data structure. However, due to a software design constraint, the system only supports stack operations (LIFO).

   As a software developer, your task is to simulate queue operations using two stacks so that customers are still served in the correct order.

2. A web browser maintains two histories:

Back History – Stores previously visited web pages.

Forward History – Stores pages that can be revisited after clicking the Back button.

To save memory, both histories are stored in a single array using two stacks.

Implement two stacks in a single array to simulate the browser's Back and Forward history.

- Implement the following operations:
- push1(int pageID) – Add a page to the Back History.
- push2(int pageID) – Add a page to the Forward History.
- pop1() – Remove the latest page from the Back History.
- pop2() – Remove the latest page from the Forward History.
- Display – Show the contents of both histories.

The program should use a single array efficiently to store both stacks.

3. A ticket booking system follows the First-In-First-Out (FIFO) principle, where the customer who arrives first is served first. Implement a queue using a linked list to simulate this process.

Write a C program to perform the following operations:

Enqueue – Insert a customer into the queue.

Dequeue – Remove the customer at the front of the queue.

Display – Display all customers currently in the queue.

Peek (Front) – Display the customer at the front of the queue.

Check if Queue is Empty – Display an appropriate message if the queue is empty.