KJ's Educational Institutes
K J College Of Engineering & Management Research, Pune.
Department of E & TC

CLASS: S. E. (E &TC) **SUBJECT:-DSA**

Ex. No: 9 **Date:**

AIM

Stack or Queue using Linked List (Dynamic Implementation)
Design a service window system where customers arrive and are served in order (FIFO), or a browser history system where the last visited page is accessed first (LIFO). Use a linked list to implement a dynamic stack (push, pop, display) or queue (add, delete, display) based on the given use case.

OBJECTIVES

- To understand the dynamic memory implementation of **Stack** and **Queue** using **Linked Lists**.
- To design a **service window system** (Queue – First In First Out) and a **browser history system** (Stack – Last In First Out).

THEORY

1. **Linked List**
   A linked list is a collection of nodes where each node contains:

   - **Data** (customer ID, webpage ID, etc.)
   - **Pointer** to the next node

   Dynamic memory allocation (malloc, free) is used to create and delete nodes at runtime.

2. **Stack (LIFO – Last In, First Out)**
   - **Use case**: Browser history (the last visited page is accessed first).
   - **Operations**:
     - push(x) → Insert element at the **top**
     - pop() → Remove element from the **top**
     - display() → Show stack contents

3. **Queue (FIFO – First In, First Out)**
   - **Use case**: Service window system (customers are served in arrival order).
   - **Operations**:
     - enqueue(x) → Insert element at the **rear**
     - dequeue() → Remove element from the **front**
     - display() → Show queue contents

ALGORITHM

For Stack:

1. Create a node dynamically.
2. push: Insert at the beginning (top).
3. pop: Delete from the beginning (top).
4. display: Traverse and print stack nodes.

For Queue:

1. Create a node dynamically.
2. enqueue: Insert at the end (rear).
3. dequeue: Delete from the front.
4. display: Traverse and print queue nodes.

INPUT:

- Enter the data to push.
- Enter the data to enqueue.
- Choice.

OUTPUT:

Case 1: Stack (Browser History – LIFO)

```
Choose System: 1. Stack  2. Queue: 1

--- STACK MENU ---
1. Push
2. Pop
3. Display
4. Exit
Enter choice: 1
Enter data to push: 101
101 pushed to stack

Enter choice: 1
Enter data to push: 202
202 pushed to stack

Enter choice: 3
Stack contents: 202 -> 101 -> NULL

Enter choice: 2
202 popped from stack

Enter choice: 3
Stack contents: 101 -> NULL
```

```
Case 2: Queue (Service Window – FIFO)

Choose System: 1. Stack  2. Queue: 2

--- QUEUE MENU ---
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter choice: 1
Enter data to enqueue: 11
11 enqueued to queue

Enter choice: 1
Enter data to enqueue: 22
22 enqueued to queue

Enter choice: 3
Queue contents: 11 -> 22 -> NULL

Enter choice: 2
11 dequeued from queue

Enter choice: 3
Queue contents: 22 -> NULL
```

CONCLUSION:-