

## **Lab 1: Implementation of Stack ADT and Queue ADT**

**Q1). A)** Use the stack abstract data type to solve the Parenthesis Matching Problem, where the input is the string containing parentheses ((), {}, [], []), and the goal is to determine if the parentheses are balanced.

**B)** Name the data structure used to solve the Parenthesis Matching Problem.

**Q2). A)** Design a ticket booking system where customers arrive to book tickets and join a queue. The system should process customers in a first-come, first-served (FIFO) manner. The queue must provide the following operations (options) to the user:

1. *Add Customer* -> Add a new customer to the booking queue.
2. *Process Booking* -> Process the ticket booking for the current customer in line and remove them from the queue after processing.
3. *View Queue* -> Display the current list of customers waiting in the queue.

The system should simulate real-life ticket booking, where customers are served in the order of arrival.

**B)** Name the data structure used for the ticket booking system.

### Example (Expected Output):

--- Welcome to ABC Ticket Booking System ---

1. Add Customer
2. Process Booking
3. View Queue

Enter Your Choice (1 - 3): **1**

Enter Customer Name: **Virat**

Customer Virat has joined the queue.

Enter Your Choice (1 - 3): **1**

Enter Customer Name: **Rohit**

Customer Rohit has joined the queue.

Enter Your Choice (1 - 3): **3**

Current Queue: Virat -> Rohit

Enter Your Choice (1 - 3): **2**

Booking processed for customer: Virat

Enter Your Choice (1 - 3): **3**

Current Queue: Rohit