**Stack Tasks**

1. **Reverse a String Using a Stack:**  
   Write a Python program to reverse a given string using a stack. Define custom methods for push and pop, and process each character of the string using these operations. The program should take a string as input and return the reversed string as output, demonstrating how stacks work with string manipulation.
2. **Balanced Parentheses Checker Using a Stack:**  
   Create a Python program that verifies if an input string has balanced parentheses. Use a stack to push opening brackets ((, {, [) and pop them when encountering their corresponding closing brackets (), }, ]). The program should return True if the parentheses in the string are balanced and False otherwise.

**Queue Tasks**

1. **Simulate a Ticketing System Using a Queue:**  
   Implement a queue to simulate a ticketing system where customers join the queue to get served. Write a Python program that allows customers to enqueue when they arrive and dequeue when served. The program should print the order of service for customers and demonstrate the use of queue operations.
2. **Reverse the Order of Elements in a Queue:**  
   Write a Python program to reverse the elements of a queue using a stack. Implement both queue and stack operations (enqueue, dequeue, push, pop) and demonstrate the reversal process by displaying the queue before and after the reversal.