**Introduction:**

This program appears to be a simple implementation of a queue system for managing a ticket queue. It allows users to join the queue, serve the next person, display the current queue, and exit the program. Here are some thoughts on the program:

1. **Functionality:** The program seems to provide the basic functionality of a queue system, allowing users to join the queue, serve the next person, and view the current queue status.
2. **Code Structure:** The code is well-structured with separate functions for different operations, which makes it readable and modular. The use of functions like **joinQueue**, **serveNext**, and **displayQueue** enhances code clarity and maintainability.
3. **User Interaction:** The program uses a simple menu-driven approach for user interaction, making it user-friendly. The loop in the **main** function ensures that the program continues to interact with the user until the user decides to quit.
4. **Error Handling:** The program includes basic error handling, such as checking if the queue is full before allowing a person to join and checking if the queue is empty before serving the next person. However, there is no validation for user input, so entering non-numeric values when prompted for a choice could cause issues.
5. **Variable Naming:** Variable names are descriptive and contribute to the readability of the code. However, it's good practice to use more descriptive names for variables like **choice** and **person** to improve code readability.

Overall, the program seems well-written for its intended purpose, but small improvements could be made for better maintainability and user input validation.

**Time complexity and space complexity:**

The time complexity of the provided program is O(n), where n is the number of elements in the queue.

The space complexity of the provided program is O(maxQueueSize), where maxQueueSize is the maximum size of the queue.

**Top of Form**