**AIRLINES RESERVATION SYSTEM**

Team Members:

1149 – Srijan Dhak

1154 – Vishal Rajput

Problem Statement:

A new airlines have only one plane of capacity 40. Your system will assign seats on each flight. 5% seats are reserved for first class and rest for economic class. Passengers are allowed to book single and group tickets. Passenger may be a Prime category passenger (first class) or normal category (economic class). Generate a passenger list sorted on category if 60% tickets are sold. Senior citizens will have preference over normal category. Passengers may enquire about seat through interactive mode.

Steps To Develop: Airline Reservation System:

1. Initialization:

- Define necessary constants such as total seats, percentages for first class and economic class, and threshold for passenger list generation.

- Define structures and enums for booking system, priority queue, and passenger details.

2. Priority Queue Implementation:

- Implement priority queue operations like initialization, creation of nodes, checking if the queue is empty, dequeueing, and enqueueing based on priority.

3. File Operations:

- Implement functions to write passenger records to a CSV file, clear CSV file, generate seat numbers, count records in CSV file based on various criteria, and convert passenger records from CSV file to a priority queue.

4. Booking Functionality:

- Implement functions to book single and group tickets.

- For single booking, prompt user for name, age, sex, and preference for first class.

- For group booking, prompt user for the number of tickets and proceed with single booking accordingly.

5. Passenger List Viewing:

- Implement functionality to view the passenger list.

- Check if the number of bookings exceeds a certain threshold.

- If yes, read passenger records from the CSV file, convert them to a priority queue, and display the list sorted by category (first class or economic class).

6. Ticket Enquiry:

- Implement functionality for passengers to inquire about their tickets.

- Prompt the user to enter their ticket number and display the corresponding passenger details if found in the CSV file.

7. Main Function:

- Display a menu for options like booking tickets, checking passenger list, ticket enquiry, and exit.

- Based on the user's choice, execute the corresponding functionality.

8. Execution:

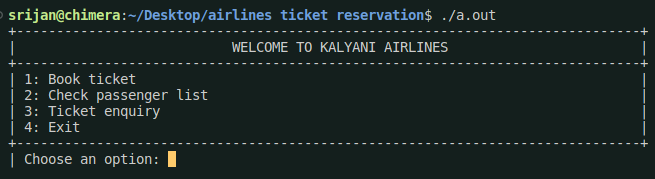
- Execute the main function to start the Airlines Reservation System.

- Prompt the user to choose an option from the menu and proceed accordingly until the user chooses to exit.

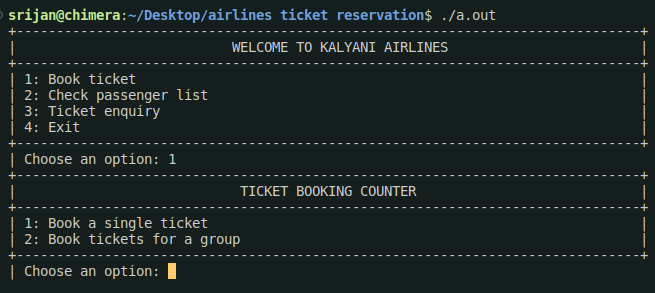
This algorithm outlines the flow and functionality of the provided code, describing how it interacts with users and manages bookings and passenger details.

Input/Output:

1. First we open the open the program and menu shows up.



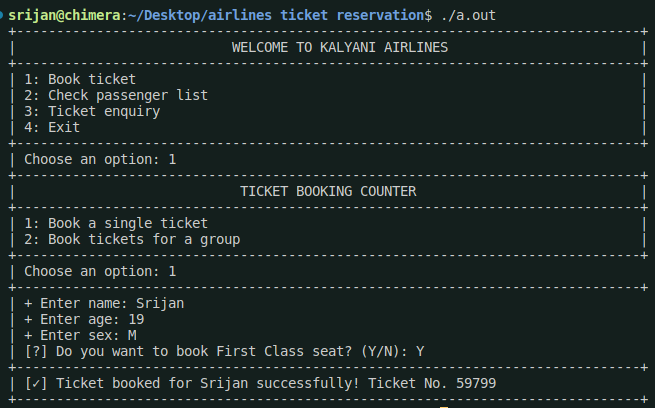
2. Then, we choose the first option (Book ticket). It prompts us with two choices: book single ticket and book group ticket.



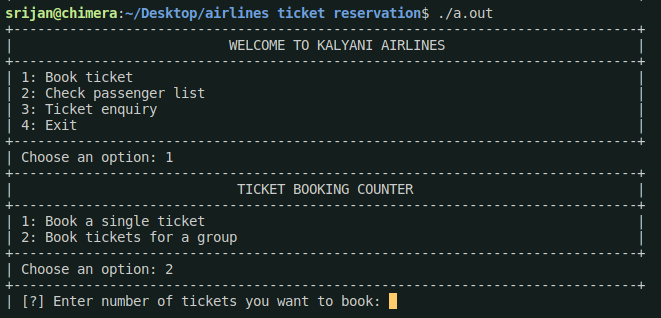
3. We choose the first one and enter our details needed for booking. Then we are asked if we want to book First Class seat for that person.



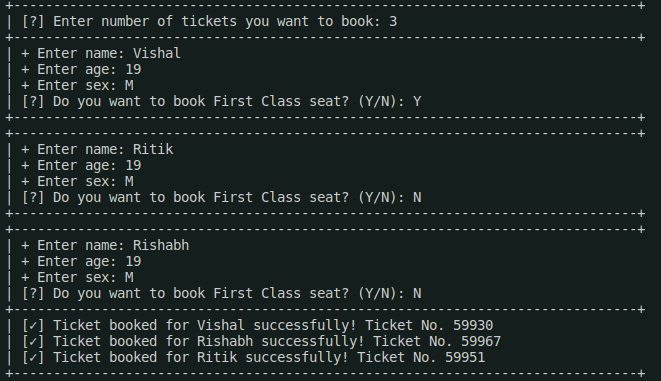
4. We choose ‘Y’ and the ticket was successfully booked and ticket number for that passenger was received.



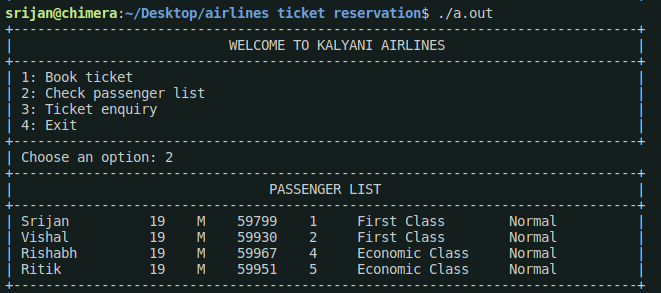
5. Now, we go back to menu and select Book Ticket option and then choose Group Booking. We are asked how many tickets we want to book.



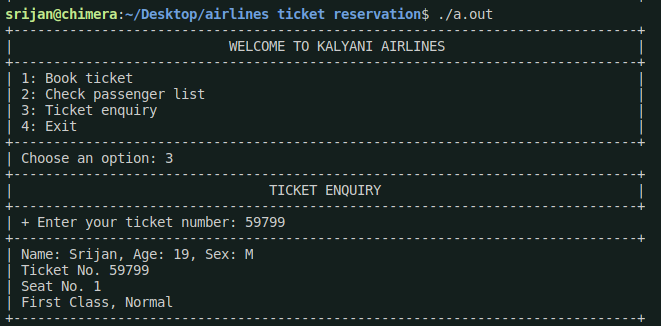
6. We enter the number of tickets we want to book and entered their details and tickets were successfully booked at the end.



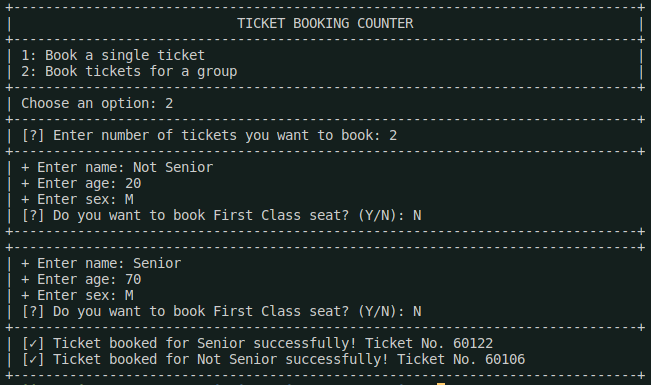
7. We go back to menu and checked the passenger list. The program returned all the tickets that were booked before in a sorted manner (by seat type).



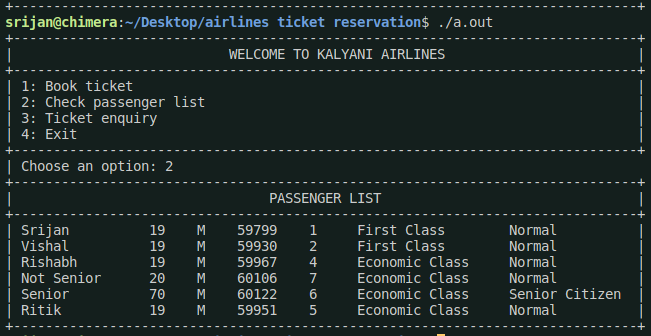
8. Now, we go back to menu and choose ‘Ticket enquiry’ option. It asks for the ticket number that was received during booking. We enter it and receive our details.



9. We again go to Ticket booking section and choose group booking. This is there is a senior in group. Their ticket is successful booked.



10. After booking, we checked the passenger list. The senior was given priority over the normal passenger.



Discussions:

Limitations:

1. Error Handling: The program lacks robust error handling. It assumes that the user will always input correct data. Adding more validation for user inputs can make the program more robust.
2. Security: The program doesn't handle sensitive information securely. For example, it reads and writes data to a file without any encryption or secure storage mechanisms.
3. Input Validation: Input validation is minimal. For instance, when taking user input, there is little validation to ensure that the entered values are within valid ranges or formats.
4. Incomplete Seat Allocation Information: The program only keeps track of the seat type, seat number, and ticket number. Additional details, such as flight information, destination, or departure time, could be included for a more comprehensive system.
5. User Interface: The user interface is text-based and may not be user-friendly. A graphical user interface (GUI) or a more interactive text-based interface could enhance the user experience.

Future Scope:

1. Graphical User Interface (GUI): Develop a graphical interface using a library like GTK, Qt, or any other suitable GUI framework. This can improve the user experience and make the system more user-friendly.
2. Database Integration: Replace the current file-based storage with a database system (e.g., SQLite, MySQL, or PostgreSQL). This would improve data management, efficiency, and allow for more complex queries.
3. User Authentication and Authorization: Implement user authentication to secure the system. Different users could have different roles (admin, staff, customer), each with specific permissions.
4. Flight Scheduling: Expand the system to include flight scheduling and availability. Users can check for available flights on specific dates and times.
5. Payment Integration: Integrate a payment system to handle booking payments securely. This can involve connecting to a payment gateway or implementing a mock payment system for educational purposes.
6. Email Notifications: Send confirmation emails or notifications to users upon successful booking or changes in their flight status. This enhances communication with passengers.
7. Seat Selection Interface: Provide a visual seat selection interface for users during the booking process. This is especially relevant for flights with assigned seating.
8. Cancellations and Refunds: Implement a system for users to cancel their bookings and receive refunds based on cancellation policies.
9. Flight Status and Updates: Incorporate a feature that allows users to check the real-time status of their flights, including delays or cancellations.
10. Mobile Application: Develop a mobile application for the airline reservation system, making it accessible on smartphones and tablets.

These future scopes can make the airline reservation system more comprehensive, feature-rich, and capable of handling various aspects of airline operations.