

TEAM MEMBERS

24KB1AO5DW: Pathan Arif

24KB1A05BB: Muram Madhu Srikar

24KB1A05K6: Gutlapalli PurnaChandra Rao

24KB1A05M1: Jorepalli Ashok

Introduction

 This presentation will explore the essential components of a flight reservation system built using C programming. We'll discuss its implementation, focusing on how arrays and linked lists are utilized to handle flight and passenger records effectively.



Introduction to Flight Reservation System

 A flight reservation system is software that allows users to search for flights, create reservations, and manage bookings. In this system, we will use arrays to store flight information such as flight numbers, departure and arrival times, and passenger details will be handled using linked lists to enable dynamic management.

Importance of Flight Reservation Systems

 Flight reservation systems streamline the booking process for airlines and customers. They enhance efficiency by automating transactions and reducing manual errors. Additionally, they help maintain an organized database for easy access to flight schedules and passenger information, ultimately improving the travel experience. C programming is a widely-used language known for its efficiency and control over system resources. Key concepts to grasp include variables, data types, functions, loops, and conditionals. Understanding how to utilize these concepts will be essential for building our flight reservation system.

Basic Concepts of C Programming

```
Anthen MI & 1 Seinell went furpomprounchet (
                                                     Phier & CitritionLanes: Interester street, 160mm=1, 46)
                                                                        Ol ther latement in
                                                                    I to good beer alles no 9 ore?
                                                             figget } tilagile Cancer Closer (1)
                                                           Milital & freesonness son while
                                                        Panceseries vegs to inditesion benja
                                              Describing : [log 1 seer see vome that the tig
                                          consiste: Unicelacterite ast Thigh sow name ]);
                    Assett (Leavest (Philips ades 3% from legificater) Than gold Amegod) ),
                                                One 1 thronoveness and lengest Michiga (1);
                                         Extended to a contrar Cortin + Fittendistrement)
                                             (1991 I Smitwell (1994) 1998); 1998);
                                                          Last Consustant Communical
CONS. SOMETHING & PRIN SONG! THAT BRITISHEN LIFES COOP TO LEAD (REVISED PRINCE) THE PRINCE FIRST
                                                        ) resonance a e (Canada /hada/an-n
                                                            Phiese (HESTERMANN SES)
                                                                      Lin you of atiout?
                                   TO SOLE CONTRIBUTION PROPERTIES CONTRIBUTED TO THE BOTH
                                    Engineering about mornel design manufactor a second cond on provided.
```





In our system, arrays will hold static information about flights like flight IDs, departure locations, and arrival times. This makes it easy to access and manipulate the data since arrays allow quick index-based access. For instance, we can know the next available flight instantly by referencing the array.

Using Arrays for Flight Management

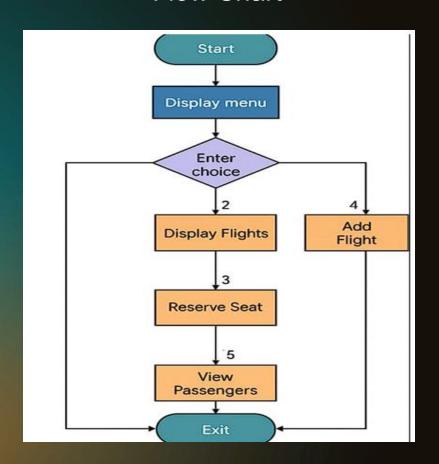
Implementing Passenger Records with Linked Lists

 We'll use linked lists to store passenger information dynamically. Unlike arrays, linked lists allow us to efficiently add or remove passengers without reallocating memory. Each node in the linked list will contain the passenger's details and a pointer to the next passenger, enabling easy traversal.

Code Structure and Flow

 The code for our flight reservation system will typically follow a modular structure. We'll define functions for each key operation, like adding a flight, making a reservation, or displaying available flights. This not only makes the code organized, but also enhances readability and allows for easier debugging.

Flow Chart



Output Images

```
-- Flight Reservation System ---
1. Add Flight
2. Display Flights
3. Reserve Seat
4. View Passengers
Exit
Enter choice: 1
Enter Flight ID: 123456
Enter Destination: delhi
Flight added successfully.
--- Flight Reservation System ---
1. Add Flight
2. Display Flights

    Reserve Seat

4. View Passengers
5. Exit
Enter choice: 3
Enter Flight ID: 123456
Enter Passenger Name: kohli
Enter Seat Number: 18
Reservation successful.
--- Flight Reservation System ---
1. Add Flight
2. Display Flights

    Reserve Seat

4. View Passengers
5. Exit
```

```
Enter choice: 2
Available Flights:
Flight ID: 123456, Destination: delhi
--- Flight Reservation System ---
1. Add Flight
2. Display Flights
Reserve Seat
4. View Passengers
5. Exit
Enter choice: 4
Enter Flight ID to view passengers: 123456
Passengers on Flight 123456:
Name: kohli, Seat No: 18
--- Flight Reservation System ---
1. Add Flight
2. Display Flights
3. Reserve Seat
4. View Passengers
Exit
Enter choice: 5
...Program finished with exit code 0
Press ENTER to exit console.
```



In summary, we've introduced a simple flight reservation system using C programming. By combining arrays for flight management and linked lists for passenger records, we've created a flexible system that demonstrates essential programming concepts. This foundational knowledge can be expanded upon for more complex applications in the future.

Conclusions

Thank You!