# Assesment-2(Tuesday)

1. 
```
#include <stdio.h>
#include <string.h>

#define MAX_SEATS 30
#define MAX_PASSENGERS 30
struct Bus {
    int busNumber;
    char route[100];
    char departureTime[100];
    int totalSeats;
    int availableSeats;
} bus;
struct Passenger {
    char name[50];
    int age;
    char contactNumber[15];
    int seatNumber;
} passengers[MAX_PASSENGERS];

int passengerCount = 0;



void inputBusDetails() {
    printf("Enter Bus Number:");
    scanf("%d", &bus.busNumber);
    printf("Enter Route:");
    scanf("%s", &bus.route);
    printf("Enter Departure Time: ");
    scanf("%s", bus.departureTime);
    printf("Enter Total Number of Seats: ");
    scanf("%d", &bus.totalSeats);
```

```
    bus.availableSeats = bus.totalSeats;
}



void allocateSeat() {
    if (bus.availableSeats > 0) {
        struct Passenger p;
        printf("Enter Passenger Name: ");
         scanf("%s", &p.name);
        printf("Enter Age: ");
        scanf("%d", &p.age);
        printf("Enter Contact Number: ");
        scanf("%s", &p.contactNumber);
       p.seatNumber = bus.totalSeats - bus.availableSeats + 1;
       passengers[passengerCount++] = p;
       bus.availableSeats--;


        printf("Seat Allocated Successfully! Seat Number: %d\n", p.seatNumber);
    } else {
        printf("No Seats Available!\n");
    }
}



void displayAllocatedSeats() {
    if (passengerCount == 0) {
        printf("No Seats Allocated Yet!\n");
    } else {
        printf("Allocated Seats and Passenger Details:\n");
        for (int i = 0; i < passengerCount; i++) {
            printf("Seat Number: %d\n", passengers[i].seatNumber);
```

```
            printf("Name: %s\n", passengers[i].name);
            printf("Age: %d\n", passengers[i].age);
            printf("Contact Number: %s\n", passengers[i].contactNumber);
        }
    }
}



void cancelSeat() {
    int seatNumber;
    printf("Enter Seat Number to Cancel: ");
    scanf("%d", &seatNumber);

    int found = 0;
    for (int i = 0; i < passengerCount; i++) {
        if (passengers[i].seatNumber == seatNumber) {
            found = 1;
            printf("Booking for %s (Seat Number %d) has been Cancelled.\n", passengers[i].name,
passengers[i].seatNumber);



            for (int j = i; j < passengerCount - 1; j++) {
                passengers[j] = passengers[j + 1];
                passengers[j].seatNumber--; // Update seat numbers
            }

            passengerCount--;
            bus.availableSeats++;
            break;
        }
    }
```

# Assesment-2(Tuesday)

```
    if (!found) {
        printf("Invalid Seat Number!\n");
    }
}


int main() {
    int choice;
    inputBusDetails();

    while (1) {
        printf("\n--- Bus Seat Allocation System ---\n");
        printf("1. Allocate Seat\n");
        printf("2. Cancel Seat\n");
        printf("3. Display Allocated Seats\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                allocateSeat();
                break;
            case 2:
                cancelSeat();
                break;
            case 3:
                displayAllocatedSeats();
                break;
            case 4:
```

# Assesment-2(Tuesday)

```
            printf("Exiting...\n");
            return 0;
        default:
            printf("Invalid Choice! Please try again.\n");
    }
  }
}
```

Output:

Enter Bus Number:9106

Enter Route: Vizag-Hyderabad

Enter Departure Time: 6:00pm

Enter Total Number of Seats: 32

--- Bus Seat Allocation System ---

1. Allocate Seat

2. Cancel Seat

3. Display Allocated Seats

4. Exit

Enter your choice: 1

Enter Passenger Name: Ananya

Enter Age: 18

Enter Contact Number: 9845730406

Seat Allocated Successfully! Seat Number: 1

--- Bus Seat Allocation System ---

1. Allocate Seat

2. Cancel Seat

3. Display Allocated Seats

4. Exit

Enter your choice: 3

# Assesment-2(Tuesday)

Allocated Seats and Passenger Details:

Seat Number: 1

Name: Ananya

Age: 18

Contact Number: 9845730406

3. #include <stdio.h>

#include <string.h>

#define MAX_TICKETS 50

struct Ticket {

    char passengerName[50];

    int age;

    char contactNumber[15];

    char travelDate[15];

    char busType[20];

    float ticketPrice;

    char paymentMethod[20];

    float paymentAmount;

    char transactionID[20];

    int isCancelled;

} tickets[MAX_TICKETS];

int ticketCount = 0;

void bookTicket();

float calculateTicketPrice(char busType[], char travelDate[]);

# Assesment-2(Tuesday)

```
void processPayment(int index);
void displayConfirmation(int index);
void displayReceipt(int index);
void cancelTicket();
void displayMenu();
void handleMenuChoice(int choice);



void bookTicket() {
    struct Ticket t;
    printf("Enter Passenger Name: ");
    scanf("%s", &t.passengerName);
    printf("Enter Age: ");
    scanf("%d", &t.age);

    printf("Enter Contact Number: ");
    scanf("%s", t.contactNumber);

    printf("Enter Travel Date (DD-MM-YYYY): ");
    scanf("%s", t.travelDate);

    printf("Enter Bus Type (AC/Non-AC/Sleeper): ");
    scanf("%s", t.busType);
    t.ticketPrice = calculateTicketPrice(t.busType, t.travelDate);
    printf("Ticket Price: %.2f\n", t.ticketPrice);
    processPayment(ticketCount);
    t.isCancelled = 0;
    tickets[ticketCount] = t;
    displayConfirmation(ticketCount);
    ticketCount++;
}
```

# Assesment-2(Tuesday)

```
void processPayment(int index) {
    printf("Enter Payment Method (Credit/Debit/UPI): ");
    scanf("%s", tickets[index].paymentMethod);

    printf("Enter Payment Amount: ");
    scanf("%f", &tickets[index].paymentAmount);

    printf("Enter Transaction ID: ");
    scanf("%s", tickets[index].transactionID);

    printf("Payment Successful!\n");
}

void displayConfirmation(int index) {
    printf("\n--- Booking Confirmation ---\n");
    printf("Passenger Name: %s\n", tickets[index].passengerName);
    printf("Age: %d\n", tickets[index].age);
    printf("Contact Number: %s\n", tickets[index].contactNumber);
    printf("Travel Date: %s\n", tickets[index].travelDate);
    printf("Bus Type: %s\n", tickets[index].busType);
    printf("Seat Number: %d\n", index + 1);
    printf("Booking Confirmed!\n");
}

void displayReceipt(int index) {
    if (index < ticketCount && !tickets[index].isCancelled) {
        printf("\n--- Payment Receipt ---\n");
        printf("Passenger Name: %s\n", tickets[index].passengerName);
        printf("Ticket Price: %.2f\n", tickets[index].ticketPrice);
        printf("Payment Method: %s\n", tickets[index].paymentMethod);
```

```
        printf("Transaction ID: %s\n", tickets[index].transactionID);
        printf("Booking Status: Confirmed\n");
    } else {
        printf("Invalid Ticket Number or Ticket is Cancelled!\n");
    }
}


void cancelTicket() {
    int ticketNumber;
    printf("Enter Ticket Number to Cancel: ");
    scanf("%d", &ticketNumber);

    if (ticketNumber > 0 && ticketNumber <= ticketCount && !tickets[ticketNumber - 1].isCancelled) {
        tickets[ticketNumber - 1].isCancelled = 1;
        printf("Ticket for %s has been Cancelled.\n", tickets[ticketNumber - 1].passengerName);
        printf("Refund Amount: %.2f\n", tickets[ticketNumber - 1].ticketPrice * 0.80);
    } else {
        printf("Invalid Ticket Number or Already Cancelled!\n");
    }
}


void displayMenu(){
    printf("\n--- Ticket Booking and Payment System ---\n");
    printf("1. Book Ticket\n");
    printf("2. Display Payment Receipt\n");
    printf("3. Cancel Ticket\n");
    printf("4. Exit\n");
    printf("Enter your choice: ");
```

```
}
int main() {
   int choice,ticketNumber;
   switch (choice) {
      case 1:
         bookTicket();
         break;
      case 2:
         printf("Enter Ticket Number to View Receipt: ");
         scanf("%d", &ticketNumber);
         displayReceipt(ticketNumber - 1);
         break;
      case 3:
         cancelTicket();
         break;
      case 4:
         printf("Exiting...\n");
         break;
      default:
         printf("Invalid Choice! Please try again.\n");
   }



   while (1) {
      displayMenu();
      scanf("%d", &choice);

      if (choice == 4) {
         break;
      }
```

```
    }


    return 0;
}
```