1.BUS SEAT ALLOCATION SYSTEM

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>


typedef struct {

    char name[50];

    int age;

    long int contact;

    int seatNo;

} Passenger;


typedef struct {

    int busNo;

    char route[100];

    char departureTime[10];

    int totalSeats;

    Passenger passengers[100];

} Bus;


void allocateSeat(Bus *bus, Passenger *passenger) {

    for (int i = 0; i < bus->totalSeats; i++) {

        if (bus->passengers[i].seatNo == 0) {

            bus->passengers[i] = *passenger;

            bus->passengers[i].seatNo = i + 1;
            printf("Seat allocated: %d\n", bus->passengers[i].seatNo);
            return;
        }
    }
    printf("No available seats!\n");
}

void cancelSeat(Bus *bus, int seatNo) {
    if (seatNo > 0 && seatNo <= bus->totalSeats) {
        if (bus->passengers[seatNo - 1].seatNo != 0) {
            bus->passengers[seatNo - 1].seatNo = 0;
            strcpy(bus->passengers[seatNo - 1].name, "");
            bus->passengers[seatNo - 1].age = 0;
            bus->passengers[seatNo - 1].contact = 0;
            printf("Seat %d canceled successfully.\n", seatNo);
        } else {
            printf("Seat %d is already vacant.\n", seatNo);
        }
    } else {
        printf("Invalid seat number.\n");
    }
}

void displaySeats(Bus bus) {
    printf("Allocated Seats for Bus No %d:\n", bus.busNo);
    int allocatedSeats = 0;
    for (int i = 0; i < bus.totalSeats; i++) {
        if (bus.passengers[i].seatNo != 0) {
            printf("Seat No: %d, Name: %s, Age: %d, Contact: %lu\n",
                   bus.passengers[i].seatNo, bus.passengers[i].name,
                   bus.passengers[i].age, bus.passengers[i].contact);
            allocatedSeats++;
        }
    }
    if (allocatedSeats == 0) {
        printf("No seats allocated yet.\n");
    }
}

int main() {
    Bus bus1;
    int choice ;
    printf("Enter Bus Number: ");
    scanf("%d", &bus1.busNo);
    getchar();
    printf("Enter Route (e.g. Route A to B): ");
    fgets(bus1.route, sizeof(bus1.route), stdin);
    bus1.route[strcspn(bus1.route, "\n")] = '\0';
    printf("Enter Departure Time (e.g. 10:00 AM): ");
    fgets(bus1.departureTime, sizeof(bus1.departureTime), stdin);
```

```
bus1.departureTime[strcspn(bus1.departureTime, "\n")] = '\0';
printf("Enter Total Number of Seats: ");
scanf("%d", &bus1.totalSeats);
for (int i = 0; i < bus1.totalSeats; i++) {
    bus1.passengers[i].seatNo = 0;
}

do {
    printf("\nBus Seat Booking System\n");
    printf("1. Book a seat\n");
    printf("2. Cancel a seat\n");
    printf("3. Display all allocated seats\n");
    printf("4. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);
    getchar();
     if (choice == 1) {
        Passenger p;
        printf("\nEnter Passenger Name: ");
        fgets(p.name, sizeof(p.name), stdin);
        p.name[strcspn(p.name, "\n")] = '\0';
        printf("Enter Passenger Age: ");
        scanf("%d", &p.age);
        printf("Enter Passenger Contact Number: ");
        scanf("%lu", &p.contact);
```

```
            p.seatNo = 0;

            allocateSeat(&bus1, &p);

        } else if (choice == 2) {

            int seatNo;

            printf("\nEnter Seat Number to Cancel: ");

            scanf("%d", &seatNo);

            cancelSeat(&bus1, seatNo);

        } else if (choice == 3)

            displaySeats(bus1);

        } else if (choice == 4) {

            printf("Exiting the program...\n");

        } else {

            printf("Invalid choice! Please try again.\n");

        }

    } while (choice != 4);


    return 0;

}
```

2.ROUTE OPTIMIZATION SYSTEM

```
#include <stdio.h>

#include <limits.h>

#include <string.h> #define

MAX 10 int
```

```
graph[MAX][MAX]; int
dist[MAX], prev[MAX]; int
visited[MAX]; void
dijkstra(int start, int n) {
for (int i = 0; i < n; i++) {
dist[i] = INT_MAX;
prev[i] = -1;        visited[i] =
0;
  }
  dist[start] = 0;

  for (int i = 0; i < n - 1; i++) {
int minDist = INT_MAX, u;
    for (int j = 0; j < n; j++) {            if
(!visited[j] && dist[j] < minDist) {
minDist = dist[j];                u = j;            }

    }
    visited[u] = 1;        for (int v = 0; v < n; v++) {            if (!visited[v]
&& graph[u][v] && dist[u] + graph[u][v] < dist[v]) {                dist[v] =
dist[u] + graph[u][v];                prev[v] = u;
      }
    }
  }
}
```

```
void printPath(int j) {
if (prev[j] == -1) {
return;
    }
    printPath(prev[j]);
printf(" -> %d", j);
}

int main() {     int n, e, start, u, v, weight;
printf("Enter the number of nodes: ");
scanf("%d", &n);     memset(graph, 0,
sizeof(graph));     printf("Enter the
number of edges: ");     scanf("%d", &e);
printf("Enter the edges (u, v) and their
weight:\n");
    for (int i = 0; i < e; i++) {
printf("Edge %d: ", i + 1);         scanf("%d
%d %d", &u, &v, &weight);
graph[u][v] = weight;         graph[v][u] =
weight;
    }
```

```
   printf("start node is from 0 ");     start=0;
dijkstra(start, n);     printf("\nOptimized Route
from %d:\n", start);
   for (int i = 0; i < n; i++) {        if (dist[i]
== INT_MAX) {            printf("No path to
node %d\n", i);
       } else {            printf("To %d, Distance: %d,
Path:", i, dist[i]);            printf("%d", start);
printPath(i);            printf("\n");
       }
   }

   return 0;
}
```

3.TICKET BOOKING AND PAYMENT SYSTEM

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
typedef struct {
char name[50];     int
age;     char
contact[15];     char
travelDate[11];
} Passenger;

typedef struct {
char busType[30];
char travelDate[11];
int ticketPrice;
} Ticket;

typedef struct {     char
paymentMethod[20];     int
amount;     char
transactionID[30];
} Payment;

void calculatePrice(Ticket *ticket) {     if (strcmp(ticket-
>busType, "Luxury") == 0) {         ticket->ticketPrice =
1000;
    } else {         ticket->ticketPrice = 500;
    }
}
```

```
void bookTicket(Passenger *passenger, Ticket *ticket) {
printf("\nBooking ticket for %s\n", passenger->name);
calculatePrice(ticket);     printf("Ticket Price: ₹%d\n",
ticket->ticketPrice);
}

void displayTicketDetails(Passenger *passenger, Ticket *ticket) {
printf("\nTicket Details:\n");     printf("Passenger Name: %s\n",
passenger->name);     printf("Age: %d\n", passenger->age);
printf("Contact: %s\n", passenger->contact);     printf("Travel
Date: %s\n", passenger->travelDate);     printf("Bus Type: %s\n",
ticket->busType);     printf("Ticket Price: ₹%d\n", ticket-
>ticketPrice);
}
void acceptPaymentDetails(Payment *payment, Ticket *ticket) {
printf("\nEnter Payment Method (e.g., Credit Card, Debit Card, UPI): ");
fgets(payment->paymentMethod, sizeof(payment->paymentMethod), stdin);
payment->paymentMethod[strcspn(payment->paymentMethod, "\n")] = '\0';

  printf("Enter  Transaction  ID:  ");        fgets(payment->transactionID,
sizeof(payment->transactionID),     stdin);                        payment-
>transactionID[strcspn(payment->transactionID, "\n")] = '\0';

  payment->amount = ticket->ticketPrice;
```

```
}


void processPayment(Payment *payment) {
printf("\nProcessing payment of ₹%d\n", payment->amount);
printf("Payment Method:");     printf(" %s\n", payment-
>paymentMethod);     printf("Transaction ID: %s\n", payment-
>transactionID);     printf("Payment successful!\n");
}

int main() {
    Passenger passenger;
    Ticket ticket;
    Payment payment;

    int choice;

    printf("Welcome to the Bus Ticket Booking System\n");
printf("\nEnter your name: ");     fgets(passenger.name,
sizeof(passenger.name), stdin);
passenger.name[strcspn(passenger.name, "\n")] = '\0';
printf("Enter your age: ");     scanf("%d",
&passenger.age);
```

```
    getchar();       printf("Enter your contact number: ");
fgets(passenger.contact, sizeof(passenger.contact), stdin);
passenger.contact[strcspn(passenger.contact, "\n")] = '\0';

    printf("\nSelect Bus Type:\n1. Luxury\n2. Regular\nEnter choice: ");
scanf("%d", &choice);     getchar();      if (choice == 1) {
strcpy(ticket.busType, "Luxury");
    } else {         strcpy(ticket.busType,
"Regular");
    }

    printf("Enter your travel date (DD-MM-YYYY): ");
fgets(ticket.travelDate, sizeof(ticket.travelDate), stdin);
ticket.travelDate[strcspn(ticket.travelDate, "\n")] = '\0';
bookTicket(&passenger, &ticket);     displayTicketDetails(&passenger,
&ticket);     acceptPaymentDetails(&payment, &ticket);
processPayment(&payment);

    return 0;
}
```

4.BUS TRACKING AND LOCATION SYSTEM

```
#include <stdio.h>
#include <stdlib.h> #include <time.h> typedef struct {     float
latitude;     float longitude;     time_t timestamp; } Location; void
updateLocation(Location *loc, float latitude, float longitude) {
loc->latitude = latitude;     loc->longitude = longitude;     loc-
>timestamp = time(NULL);
}

void displayLocation(Location loc) {     printf("Current Location: Latitude %.2f, Longitude
%.2f\n", loc.latitude, loc.longitude);     printf("Last Updated: %s",
ctime(&loc.timestamp));
}

int main() {
    Location bus1Location;     float latitude, longitude;
printf("Enter bus latitude: ");     scanf("%f", &latitude);
printf("Enter bus longitude: ");     scanf("%f",
&longitude);     updateLocation(&bus1Location,
latitude, longitude);     displayLocation(bus1Location);

    return 0;
}
```