Hotel Reservation System
Specifications:
Variables: Room number, guest name, check-in and check-out dates.
Static & Const: Static variable for total reservations; const for maximum rooms.
Switch Case: Menu for booking, canceling, and viewing reservations.
Looping Statements: Loop through reservation records.
Pointers: Pointer for dynamic allocation of guest details.
Functions: Functions for each reservation operation.
Arrays: Store reservation details.
Structures: Structure for reservation details.
Nested Structures: Nested structures for guest and room details.
Unions: Union for payment methods.
Nested Unions: Nested union for various payment details.
Output Expectations: Display reservation list and room availability.
Menu Example:
1. Book Room
2. Cancel Reservation
3. View Reservations
4. Exit

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

#define MAX_ROOM 10

static int totalReservation =0;
struct Guest{
    char name[100];
    char check_in[50];
    char check_out[50];
};
struct Reservation{
    int roomNumber;
    struct Guest guest;
};

union PaymentMethod{
    char card[50];
    char cash[50];
```

```
};
struct PayementDetails{
   int RoomNumber;
   union PaymentMethod payment;
};
void BookRoom();
void cancel();
void view_reservation();

struct Reservation reservations[MAX_ROOM];
int main(){

   int choice;
   while(1){
      printf("\n---------- Hotel Reservation System----------\n");
      printf("1. Book Room \n");
      printf("2. Cancel Reservation \n");
      printf("3 . View Reservation \n");
      printf("4. Exit \n");
      printf("Enter your choice \n");
      scanf("%d",&choice);

      switch(choice){
         case 1:
            BookRoom();
            break;
         case 2:
            cancel();
            break;
         case 3:
            view_reservation();
            break;
         case 4:
            printf("Exiting .....\n");
            return 0;
            break;
         default:
            printf("Invalid Choice \n");

      }
```

```
    }
    return 0;
}

void BookRoom(){
    if(totalReservation >=MAX_ROOM){
        printf("Rooms are full \n");
        return;
    }
    struct Reservation *res = &reservations[totalReservation];
    res->roomNumber = totalReservation +1;
    printf("\n Enter the Room number\n");
    scanf("%d",&res->roomNumber);
    printf("Enter guest Name ");
    scanf("%s",res->guest.name);
    printf("Enter check IN date (YYYY-MM-DD) ");
    scanf("%s",res->guest.check_in);
    printf("Enter check out  date (YYYY-MM-DD) ");
    scanf("%s",res->guest.check_out);

    totalReservation++;
    printf("Room booked Sucessfully! ");
}




void cancel(){

    int room_number;
    printf("\nENter room number to be cancel \n");
    scanf("%d",&room_number);
    for(int i=0;i<totalReservation;i++){
        if(reservations[i].roomNumber == room_number){
            for(int j=i;j<totalReservation-1;j++){
            reservations[j] =reservations[j+1];
        }
        totalReservation--;
        printf("\nReservation cancelled. \n");
        return;
        }
```

```
    }
    printf("Room not found \n");
}
void view_reservation(){
    if(totalReservation ==0){
        printf("\nNo reservation Available \n");
        return;
    }
    for(int i=0;i<totalReservation;i++){
        struct Reservation *res = &reservations[i];
        printf("\nRoom : %d \n",res->roomNumber);
        printf("\nGuest : %s \n",res->guest.name);
        printf("\nCheck In %s \n",res->guest.check_in);
        printf("\nCheckout : %d \n",res->guest.check_out);
    }
}
```