Q1

```
// 1. Create a structure Book with data members as bname, id, author, price.
// Accept the values of all these members from user and display them.

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

typedef struct Book{
   char bname[10];
   int id;
   char auther[20];
   int price;
}Book;

Book Detail(){
   Book b;
   printf("Enter book name: ");
   scanf("%s", b.bname);

   printf("Enter book id: ");
   scanf("%d", &b.id);

   printf("Enter author name: ");
   scanf("%s", b.auther);

   printf("Enter price: ");
   scanf("%d", &b.price);

   return b;
}
void display( Book b){
```

```
  printf("Name = %s\nId = %d \nAuther = %s\nPrice = %d",b.bname,b.id,b.auther,b.price);
}

int main(){
   Book b1;
   b1=Detail();

   display(b1);

   return 0;
}
```

Q2

```
// Create a structure Time with data members as hrs, min, sec. Accept the values
// of all these members from user and display them. Also perform addition of two
//  time variables and display the result. If sec goes beyond 60, carry it to
//  min etc. Add a method to convert the given time into sec.
#include<stdio.h>
#include<string.h>

typedef struct Time {
   int hr;
   int min;
   int sec;
} Time;

Time Input() {
   Time t;
   printf("Enter time in hr min sec format: ");
   scanf("%d %d %d", &t.hr, &t.min, &t.sec);
   return t;
```

```
}

void display(Time t) {
    printf("%02d:%02d:%02d\n", t.hr, t.min, t.sec);
}

Time Add(Time t1, Time t2) {
    Time result;
    result.sec = t1.sec + t2.sec;
    result.min = t1.min + t2.min + (result.sec / 60);
    result.sec %= 60;
    result.hr = t1.hr + t2.hr + (result.min / 60);
    result.min %= 60;
    return result;
}

int Convert(Time t) {
    return t.hr * 3600 + t.min * 60 + t.sec;
}

int main() {
    Time t1, t2, sum;
    int totalSecond;

    printf("Enter first time:\n");
    t1 = Input();

    printf("Enter second time:\n");
    t2 = Input();

    printf("\nFirst Time: ");
    display(t1);
```

```
    printf("Second Time: ");
    display(t2);

    sum = Add(t1, t2);

    printf("\nSum of Times: ");
    display(sum);

    totalSecond = Convert(sum);

    printf("\nTotal seconds of the resulting time: %d seconds\n", totalSecond);

    return 0;
}
```

Q3

```
// 3. Write  a program to create an array for 10 players. For each player store name, no. of
// matches played, runs, wickets takes.
// a. Create function to Accept the information of each player.
// b. Create function to display the information of all the players
// c. Display the information of player who made maximum runs and the one who took
// maximum number of wickets.

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

typedef struct Player {
    char name[20];
```

```
    int matches;
    int runs;
    int wickets;
} Player;

void inputPlayers(Player players[], int size) {
    for (int i = 0; i < size; i++) {
        printf("\nEnter details for player %d:\n", i + 1);
        printf("Name: ");
        scanf("%s", players[i].name);
        printf("Matches played: ");
        scanf("%d", &players[i].matches);
        printf("Runs scored: ");
        scanf("%d", &players[i].runs);
        printf("Wickets taken: ");
        scanf("%d", &players[i].wickets);
    }
}

void displayPlayers(Player players[], int size) {
    printf("\n%-15s %-10s %-10s %-10s\n", "Name", "Matches", "Runs", "Wickets");
    for (int i = 0; i < size; i++) {
        printf("%-15s %-10d %-10d %-10d\n", players[i].name, players[i].matches, players[i].runs,
players[i].wickets);
    }
}

void findMaxRunsWickets(Player players[], int size) {
    int maxRuns = 0, maxWickets = 0;
    int maxRunsIndex = 0, maxWicketsIndex = 0;

    for (int i = 0; i < size; i++) {
        if (players[i].runs > maxRuns) {
```

```
            maxRuns = players[i].runs;
            maxRunsIndex = i;
        }
        if (players[i].wickets > maxWickets) {
            maxWickets = players[i].wickets;
            maxWicketsIndex = i;
        }
    }

    printf("\nPlayer with maximum runs: %s (%d runs)\n", players[maxRunsIndex].name, maxRuns);
    printf("Player with maximum wickets: %s (%d wickets)\n", players[maxWicketsIndex].name,
maxWickets);
}

int main() {
    Player players[10];

    inputPlayers(players, 10);

    printf("\n=== Player Information ===\n");
    displayPlayers(players, 10);

    findMaxRunsWickets(players, 10);

    return 0;
}
```

Q4

```
// Point of Sale System: Build a simple point of sale system using structures to
// represent products with attributes like name, price, and quantity. Allow users
// to add items to a cart and calculate the total cost.
```

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

#define MAX_PRODUCTS 5
#define MAX_CART 10

typedef struct Product {
    char name[20];
    float price;
    int quantity;
} Product;

typedef struct CartItem {
    char name[20];
    float price;
    int quantity;
} CartItem;

void displayProducts(Product products[], int size) {
    printf("\nAvailable Products:\n");
    printf("%-5s %-15s %-8s %-10s\n", "ID", "Name", "Price", "Quantity");
    for (int i = 0; i < size; i++) {
        printf("%-5d %-15s $%-7.2f %-10d\n", i + 1, products[i].name, products[i].price,
products[i].quantity);
    }
}

void addToCart(Product products[], CartItem cart[], int *cartSize, int size) {
    int productId, quantity;

    printf("\nEnter product ID to add to cart (0 to finish): ");
    scanf("%d", &productId);
```

```
    while (productId != 0) {
        if (productId < 1 || productId > size) {
            printf("Invalid product ID. Try again: ");
        } else {
            printf("Enter quantity: ");
            scanf("%d", &quantity);

            if (quantity <= 0 || quantity > products[productId - 1].quantity) {
                printf("Invalid quantity. Only %d available.\n", products[productId - 1].quantity);
            } else {
                strcpy(cart[*cartSize].name, products[productId - 1].name);
                cart[*cartSize].price = products[productId - 1].price;
                cart[*cartSize].quantity = quantity;
                (*cartSize)++;

                products[productId - 1].quantity -= quantity;

                printf("%s added to cart.\n", products[productId - 1].name);
            }
        }
        printf("\nEnter product ID to add to cart (0 to finish): ");
        scanf("%d", &productId);
    }
}

void displayCart(CartItem cart[], int cartSize) {
    if (cartSize == 0) {
        printf("\nYour cart is empty!\n");
        return;
    }
```

```
    float total = 0;
    printf("\nYour Cart:\n");
    printf("%-15s %-8s %-10s\n", "Product", "Price", "Quantity");

    for (int i = 0; i < cartSize; i++) {
        printf("%-15s $%-7.2f %-10d\n", cart[i].name, cart[i].price, cart[i].quantity);
        total += cart[i].price * cart[i].quantity;
    }

    printf("\nTotal Amount: $%.2f\n", total);
}

int main() {
    Product products[MAX_PRODUCTS] = {
        {"Apples", 1.50, 20},
        {"Bananas", 0.99, 30},
        {"Milk", 2.50, 15},
        {"Bread", 1.99, 10},
        {"Eggs", 3.00, 12}
    };

    CartItem cart[MAX_CART];
    int cartSize = 0;

    printf("=== Welcome to the Point of Sale System ===\n");
    displayProducts(products, MAX_PRODUCTS);

    addToCart(products, cart, &cartSize, MAX_PRODUCTS);

    displayCart(cart, cartSize);

    printf("\nThank you for shopping with us!\n");
```

```
    return 0;
}
```

Q5

```
// Movie Database: Create a program that uses structures to manage a movie
// database with details like title, director, release year, and genre. Allow users
// to add, search for, and update movie records.

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

#define MAX_MOVIES 20

typedef struct Movie {
    char title[50];
    char director[30];
    int year;
    char genre[20];
} Movie;

void addMovie(Movie movies[], int *count) {
    if (*count >= MAX_MOVIES) {
        printf("Movie database is full!\n");
        return;
    }

    printf("Enter movie title: ");
    scanf(" %[^\n]", movies[*count].title);
```

```
    printf("Enter director: ");
    scanf(" %[^\n]", movies[*count].director);

    printf("Enter release year: ");
    scanf("%d", &movies[*count].year);

    printf("Enter genre: ");
    scanf(" %[^\n]", movies[*count].genre);

    (*count)++;
    printf("Movie added successfully!\n");
}

void displayMovies(Movie movies[], int count) {
    if (count == 0) {
        printf("No movies in the database yet.\n");
        return;
    }

    printf("\n%-30s %-20s %-5s %-15s\n", "Title", "Director", "Year", "Genre");
    for (int i = 0; i < count; i++) {
        printf("%-30s %-20s %-5d %-15s\n", movies[i].title, movies[i].director, movies[i].year,
movies[i].genre);
    }
}

void searchMovie(Movie movies[], int count) {
    if (count == 0) {
        printf("No movies available to search.\n");
        return;
    }

    char searchTitle[50];
```

```
    printf("Enter the movie title to search: ");
    scanf(" %[^\n]", searchTitle);

    for (int i = 0; i < count; i++) {
        if (strcmp(movies[i].title, searchTitle) == 0) {
            printf("\nMovie found!\n");
            printf("Title: %s\nDirector: %s\nYear: %d\nGenre: %s\n",
                   movies[i].title, movies[i].director, movies[i].year, movies[i].genre);
            return;
        }
    }

    printf("Movie not found!\n");
}

void updateMovie(Movie movies[], int count) {
    if (count == 0) {
        printf("No movies available to update.\n");
        return;
    }

    char updateTitle[50];
    printf("Enter the movie title to update: ");
    scanf(" %[^\n]", updateTitle);

    for (int i = 0; i < count; i++) {
        if (strcmp(movies[i].title, updateTitle) == 0) {
            printf("Enter new director: ");
            scanf(" %[^\n]", movies[i].director);

            printf("Enter new release year: ");
            scanf("%d", &movies[i].year);
```

```
            printf("Enter new genre: ");
            scanf(" %[^\n]", movies[i].genre);

            printf("Movie updated successfully!\n");
            return;
        }
    }

    printf("Movie not found!\n");
}

int main() {
    Movie movies[MAX_MOVIES];
    int count = 0;
    int choice;

    while (1) {
        printf("\n=== Movie Database Menu ===\n");
        printf("1. Add Movie\n");
        printf("2. Display All Movies\n");
        printf("3. Search for a Movie\n");
        printf("4. Update Movie Details\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                addMovie(movies, &count);
                break;
            case 2:
```

```
            displayMovies(movies, count);
            break;
        case 3:
            searchMovie(movies, count);
            break;
        case 4:
            updateMovie(movies, count);
            break;
        case 5:
            printf("Exiting the Movie Database. Goodbye!\n");
            return 0;
        default:
            printf("Invalid choice. Please try again.\n");
      }
   }
}
```