**Problem 1: Employee Management System**
**Objective: Create a program to manage employee details using structures.**
**Description:**
**Define a structure Employee with fields:**
**int emp_id: Employee ID**
**char name[50]: Employee name**
**float salary: Employee salary**
**Write a menu-driven program to:**
**Add an employee.**
**Update employee salary by ID.**
**Display all employee details.**
**Find and display details of the employee with the highest salary.**

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

struct Employee {
  int emp_id;
  char name[50];
  float salary;
};

void addEmployee(struct Employee employees[], int *count);
void updateSalary(struct Employee employees[], int count);
void displayEmployees(struct Employee employees[], int count);
void highestSalaryEmployee(struct Employee employees[], int count);

int main() {
  struct Employee employees[100];
  int count = 0, choice;

  do {
    printf("\n--- Employee Management System ---\n");
    printf("1. Add Employee\n");
    printf("2. Update Salary by ID\n");
    printf("3. Display All Employees\n");
    printf("4. Employee with Highest Salary\n");
    printf("5. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch(choice) {
      case 1:
        addEmployee(employees, &count);
        break;
      case 2:
        updateSalary(employees, count);
```

```
            break;
        case 3:
            displayEmployees(employees, count);
            break;
        case 4:
            highestSalaryEmployee(employees, count);
            break;
        case 5:
            printf("Exiting...\n");
            break;
        default:
            printf("Invalid choice, try again.\n");
    }
  } while (choice != 5);

  return 0;
}

void addEmployee(struct Employee employees[], int *count) {
  printf("Enter employee ID: ");
  scanf("%d", &employees[*count].emp_id);
  printf("Enter employee name: ");
  scanf(" %[^\n]", employees[*count].name);  // To read spaces
  printf("Enter employee salary: ");
  scanf("%f", &employees[*count].salary);
  (*count)++;
}

void updateSalary(struct Employee employees[], int count) {
  int id, found = 0;
  printf("Enter employee ID to update salary: ");
  scanf("%d", &id);
  for (int i = 0; i < count; i++) {
    if (employees[i].emp_id == id) {
      printf("Enter new salary: ");
      scanf("%f", &employees[i].salary);
      found = 1;
      break;
    }
  }
  if (!found) printf("Employee not found.\n");
}

void displayEmployees(struct Employee employees[], int count) {
  for (int i = 0; i < count; i++) {
    printf("ID: %d, Name: %s, Salary: %.2f\n", employees[i].emp_id, employees[i].name,
employees[i].salary);
```

```
    }
}

void highestSalaryEmployee(struct Employee employees[], int count) {
    if (count == 0) {
        printf("No employees to display.\n");
        return;
    }

    int maxIndex = 0;
    for (int i = 1; i < count; i++) {
        if (employees[i].salary > employees[maxIndex].salary) {
            maxIndex = i;
        }
    }

    printf("Employee with highest salary: %s, Salary: %.2f\n",
employees[maxIndex].name, employees[maxIndex].salary);
}
```

OUTPUT

```
--- Employee Management System ---
1. Add Employee
2. Update Salary by ID
3. Display All Employees
4. Employee with Highest Salary
5. Exit
Enter your choice: 1
Enter employee ID: 100
Enter employee name: MEGHA
Enter employee salary: 30000

--- Employee Management System ---
1. Add Employee
2. Update Salary by ID
3. Display All Employees
4. Employee with Highest Salary
5. Exit
Enter your choice: 1
Enter employee ID: 101
Enter employee name: AISWARYA
Enter employee salary: 25000

--- Employee Management System ---
1. Add Employee
2. Update Salary by ID
3. Display All Employees
```

4. Employee with Highest Salary
5. Exit
Enter your choice: 1
Enter employee ID: 102
Enter employee name: SREETHU
Enter employee salary: 20000

--- Employee Management System ---
1. Add Employee
2. Update Salary by ID
3. Display All Employees
4. Employee with Highest Salary
5. Exit
Enter your choice: 2
Enter employee ID to update salary: 103
Employee not found.

--- Employee Management System ---
1. Add Employee
2. Update Salary by ID
3. Display All Employees
4. Employee with Highest Salary
5. Exit
Enter your choice: 1
Enter employee ID: 102
Enter employee name: SREETHU
Enter employee salary: 28000

--- Employee Management System ---
1. Add Employee
2. Update Salary by ID
3. Display All Employees
4. Employee with Highest Salary
5. Exit
Enter your choice: 3
ID: 100, Name: MEGHA, Salary: 30000.00
ID: 101, Name: AISWARYA, Salary: 25000.00
ID: 102, Name: SREETHU, Salary: 20000.00
ID: 102, Name: SREETHU, Salary: 28000.00

--- Employee Management System ---
1. Add Employee
2. Update Salary by ID
3. Display All Employees
4. Employee with Highest Salary
5. Exit
Enter your choice: 4

Employee with highest salary: MEGHA, Salary: 30000.00

--- Employee Management System ---
1. Add Employee
2. Update Salary by ID
3. Display All Employees
4. Employee with Highest Salary
5. Exit
Enter your choice: 5
Exiting...

**Problem 2: Library Management System**
**Objective: Manage a library system with a structure to store book details.**
**Description:**
**Define a structure Book with fields:**
**int book_id: Book ID**
**char title[100]: Book title**
**char author[50]: Author name**
**int copies: Number of available copies**
**Write a program to:**
**Add books to the library.**
**Issue a book by reducing the number of copies.**
**Return a book by increasing the number of copies.**
**Search for a book by title or author name.**

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

struct Book {
   int book_id;
   char title[100];
   char author[50];
   int copies;
};

void addBook(struct Book library[], int *count);
void issueBook(struct Book library[], int count);
void returnBook(struct Book library[], int count);
void searchBook(struct Book library[], int count);

int main() {
   struct Book library[100];
   int count = 0, choice;

   do {
```

```
        printf("\n--- Library Management System ---\n");
        printf("1. Add Book\n");
        printf("2. Issue Book\n");
        printf("3. Return Book\n");
        printf("4. Search Book by Title/Author\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch(choice) {
            case 1:
                addBook(library, &count);
                break;
            case 2:
                issueBook(library, count);
                break;
            case 3:
                returnBook(library, count);
                break;
            case 4:
                searchBook(library, count);
                break;
            case 5:
                printf("Exiting...\n");
                break;
            default:
                printf("Invalid choice, try again.\n");
        }
    } while (choice != 5);

    return 0;
}

void addBook(struct Book library[], int *count) {
    printf("Enter book ID: ");
    scanf("%d", &library[*count].book_id);
    printf("Enter book title: ");
    scanf(" %[^\n]", library[*count].title);  // To read spaces in title
    printf("Enter author name: ");
    scanf(" %[^\n]", library[*count].author);  // To read spaces in author name
    printf("Enter number of copies: ");
    scanf("%d", &library[*count].copies);
    (*count)++;
    printf("Book added successfully!\n");
}

void issueBook(struct Book library[], int count) {
```

```
    int id, found = 0;
    printf("Enter book ID to issue: ");
    scanf("%d", &id);
    for (int i = 0; i < count; i++) {
        if (library[i].book_id == id) {
            if (library[i].copies > 0) {
                library[i].copies--;
                printf("Book issued successfully!\n");
            } else {
                printf("No copies available for this book.\n");
            }
            found = 1;
            break;
        }
    }
    if (!found) {
        printf("Book not found.\n");
    }
}

void returnBook(struct Book library[], int count) {
    int id, found = 0;
    printf("Enter book ID to return: ");
    scanf("%d", &id);
    for (int i = 0; i < count; i++) {
        if (library[i].book_id == id) {
            library[i].copies++;
            printf("Book returned successfully!\n");
            found = 1;
            break;
        }
    }
    if (!found) {
        printf("Book not found.\n");
    }
}

void searchBook(struct Book library[], int count) {
    char searchTerm[50];
    int found = 0;
    printf("Enter book title or author name to search: ");
    scanf(" %[^\n]", searchTerm);  // To read spaces in search term
    for (int i = 0; i < count; i++) {
        if (strstr(library[i].title, searchTerm) != NULL || strstr(library[i].author, searchTerm) !=
NULL) {
            printf("Book ID: %d, Title: %s, Author: %s, Copies: %d\n", library[i].book_id,
library[i].title, library[i].author, library[i].copies);
```

```
            found = 1;
        }
    }
    if (!found) {
        printf("No books found with the given title or author.\n");
    }
}
```

OUTPUT

```
--- Library Management System ---
1. Add Book
2. Issue Book
3. Return Book
4. Search Book by Title/Author
5. Exit
Enter your choice: 1
Enter book ID: 200
Enter book title: 12 NOV
Enter author name: CHETAN BHAGATH
Enter number of copies: 2
Book added successfully!

--- Library Management System ---
1. Add Book
2. Issue Book
3. Return Book
4. Search Book by Title/Author
5. Exit
Enter your choice: 1
Enter book ID: 201
```

Enter book title: AADUJEEVITHAM

Enter author name: BENYAMAN

Enter number of copies: 1

Book added successfully!

--- Library Management System ---

1. Add Book

2. Issue Book

3. Return Book

4. Search Book by Title/Author

5. Exit

Enter your choice: 5

Exiting...

**Problem 3: Cricket Player Statistics**
**Objective: Store and analyze cricket player performance data.**
**Description:**
**Define a structure Player with fields:**
**char name[50]: Player name**
**int matches: Number of matches played**
**int runs: Total runs scored**
**float average: Batting average**
**Write a program to:**
**Input details for n players.**
**Calculate and display the batting average for each player.**
**Find and display the player with the highest batting average.**

```
#include <stdio.h>

struct Player {
   char name[50];
   int matches;
   int runs;
   float average;
};
```

```
void inputPlayerData(struct Player players[], int n);
void calculateBattingAverage(struct Player players[], int n);
void displayPlayerWithHighestAverage(struct Player players[], int n);

int main() {
   int n;

   printf("Enter number of players: ");
   scanf("%d", &n);

   struct Player players[n];

   inputPlayerData(players, n);
   calculateBattingAverage(players, n);
   displayPlayerWithHighestAverage(players, n);

   return 0;
}

void inputPlayerData(struct Player players[], int n) {
   for (int i = 0; i < n; i++) {
      printf("\nEnter details for Player %d:\n", i + 1);
      printf("Name: ");
      scanf(" %[^\n]", players[i].name);  // To read spaces in player name
      printf("Number of matches: ");
      scanf("%d", &players[i].matches);
      printf("Total runs scored: ");
      scanf("%d", &players[i].runs);
   }
}

void calculateBattingAverage(struct Player players[], int n) {
   for (int i = 0; i < n; i++) {
      if (players[i].matches != 0) {
         players[i].average = (float)players[i].runs / players[i].matches;
      } else {
         players[i].average = 0;  // If no matches, average is 0
      }
   }
}

void displayPlayerWithHighestAverage(struct Player players[], int n) {
   int highestIndex = 0;

   for (int i = 1; i < n; i++) {
      if (players[i].average > players[highestIndex].average) {
         highestIndex = i;
```

```
    }
  }

  printf("\nPlayer with the highest batting average:\n");
  printf("Name: %s\n", players[highestIndex].name);
  printf("Batting Average: %.2f\n", players[highestIndex].average);
}
```

OUTPUT

```
Enter number of players: 3

Enter details for Player 1:
Name: Virat Kohli
Number of matches: 200
Total runs scored: 8000

Enter details for Player 2:
Name: Steve Smith
Number of matches: 150
Total runs scored: 6500

Enter details for Player 3:
Name: Joe Root
Number of matches: 120
Total runs scored: 5400
Batting averages are calculated for each player:
Virat Kohli: 8000 / 200 = 40.00
Steve Smith: 6500 / 150 = 43.33
Joe Root: 5400 / 120 = 45.00

Player with the highest batting average:
Name: Joe Root
Batting Average: 45.00
```

**Problem 4: Flight Reservation System**
**Objective: Simulate a simple flight reservation system using structures.**
**Description:**
**Define a structure Flight with fields:**
**char flight_number[10]: Flight number**
**char destination[50]: Destination city**
**int available_seats: Number of available seats**

**Write a program to:**
**Add flights to the system.**
**Book tickets for a flight, reducing available seats accordingly.**
**Display the flight details based on destination.**
**Cancel tickets, increasing the number of available seats.**
**has context menu**

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

struct Flight {
  char flight_number[10];
  char destination[50];
  int available_seats;
};

void addFlight(struct Flight flights[], int *count);
void bookTicket(struct Flight flights[], int count);
void cancelTicket(struct Flight flights[], int count);
void displayFlights(struct Flight flights[], int count);

int main() {
  struct Flight flights[100];
  int count = 0, choice;

  do {
    printf("\n--- Flight Reservation System ---\n");
    printf("1. Add Flight\n");
    printf("2. Book Ticket\n");
    printf("3. Cancel Ticket\n");
    printf("4. Display Flights by Destination\n");
    printf("5. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch(choice) {
      case 1:
        addFlight(flights, &count);
        break;
      case 2:
        bookTicket(flights, count);
        break;
      case 3:
        cancelTicket(flights, count);
        break;
      case 4:
        displayFlights(flights, count);
```

```
            break;
        case 5:
            printf("Exiting...\n");
            break;
        default:
            printf("Invalid choice, try again.\n");
    }
  } while (choice != 5);

  return 0;
}

void addFlight(struct Flight flights[], int *count) {
  printf("Enter flight number: ");
  scanf("%s", flights[*count].flight_number);
  printf("Enter destination city: ");
  scanf(" %[^\n]", flights[*count].destination);  // To read spaces in destination
  printf("Enter available seats: ");
  scanf("%d", &flights[*count].available_seats);
  (*count)++;
  printf("Flight added successfully!\n");
}

void bookTicket(struct Flight flights[], int count) {
  char flight_num[10];
  int found = 0;
  printf("Enter flight number to book: ");
  scanf("%s", flight_num);
  for (int i = 0; i < count; i++) {
    if (strcmp(flights[i].flight_number, flight_num) == 0) {
      if (flights[i].available_seats > 0) {
        flights[i].available_seats--;
        printf("Ticket booked successfully! Remaining seats: %d\n",
flights[i].available_seats);
      } else {
        printf("No available seats for this flight.\n");
      }
      found = 1;
      break;
    }
  }
  if (!found) {
    printf("Flight not found.\n");
  }
}

void cancelTicket(struct Flight flights[], int count) {
```

```
    char flight_num[10];
    int found = 0;
    printf("Enter flight number to cancel: ");
    scanf("%s", flight_num);
    for (int i = 0; i < count; i++) {
        if (strcmp(flights[i].flight_number, flight_num) == 0) {
            flights[i].available_seats++;
            printf("Ticket canceled successfully! Available seats: %d\n",
flights[i].available_seats);
            found = 1;
            break;
        }
    }
    if (!found) {
        printf("Flight not found.\n");
    }
}

void displayFlights(struct Flight flights[], int count) {
    char dest[50];
    int found = 0;
    printf("Enter destination city to search: ");
    scanf(" %[^\n]", dest);  // To read spaces in destination city
    for (int i = 0; i < count; i++) {
        if (strstr(flights[i].destination, dest) != NULL) {
            printf("Flight Number: %s, Destination: %s, Available Seats: %d\n",
flights[i].flight_number, flights[i].destination, flights[i].available_seats);
            found = 1;
        }
    }
    if (!found) {
        printf("No flights found to the destination %s.\n", dest);
    }
}
```

OUTPUT

```
--- Flight Reservation System ---
1. Add Flight
2. Book Ticket
3. Cancel Ticket
4. Display Flights by Destination
5. Exit
Enter your choice: 1

Enter flight number: AI202
Enter destination city: New York
Enter available seats: 100
```

Flight added successfully!
--- Flight Reservation System ---
1. Add Flight
2. Book Ticket
3. Cancel Ticket
4. Display Flights by Destination
5. Exit
Enter your choice: 2

Enter flight number to book: AI202
Ticket booked successfully! Remaining seats: 99

--- Flight Reservation System ---
1. Add Flight
2. Book Ticket
3. Cancel Ticket
4. Display Flights by Destination
5. Exit
Enter your choice: 3

Enter flight number to cancel: AI202
Ticket canceled successfully! Available seats: 100

--- Flight Reservation System ---
1. Add Flight
2. Book Ticket
3. Cancel Ticket
4. Display Flights by Destination
5. Exit
Enter your choice: 4

Enter destination city to search: New York
Flight Number: AI202, Destination: New York, Available Seats: 100

**5. Problem 1: Student Record Management System Objective Create a program to manage student records using structures. Requirements 1. Define a Student structure with the following fields: char name[50] int rollNumber float marks 2. Implement functions to: Add a new student record. * Display all student records. Find and display a student record by roll number. Calculate and display the average marks of all students. 3. Implement a menu-driven interface to perform the above operations.**

```
#include <stdio.h>

struct Student {
   char name[50];
   int rollNumber;
```

```
    float marks;
};

void addStudent(struct Student students[], int *count);
void displayStudents(struct Student students[], int count);
void findStudentByRollNumber(struct Student students[], int count);
void calculateAverageMarks(struct Student students[], int count);

int main() {
    struct Student students[100];
    int count = 0;
    int choice;

    do {
        printf("\n--- Student Record Management System ---\n");
        printf("1. Add Student\n");
        printf("2. Display All Students\n");
        printf("3. Find Student by Roll Number\n");
        printf("4. Calculate Average Marks\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                addStudent(students, &count);
                break;
            case 2:
                displayStudents(students, count);
                break;
            case 3:
                findStudentByRollNumber(students, count);
                break;
            case 4:
                calculateAverageMarks(students, count);
                break;
            case 5:
                printf("Exiting the program.\n");
                break;
            default:
                printf("Invalid choice, please try again.\n");
        }
    } while (choice != 5);

    return 0;
}
```

```
void addStudent(struct Student students[], int *count) {
   printf("\nEnter student name: ");
   scanf(" %[^\n]", students[*count].name);
   printf("Enter roll number: ");
   scanf("%d", &students[*count].rollNumber);
   printf("Enter marks: ");
   scanf("%f", &students[*count].marks);
   (*count)++;
   printf("Student added successfully!\n");
}

void displayStudents(struct Student students[], int count) {
   if (count == 0) {
      printf("\nNo student records available.\n");
      return;
   }
   printf("\n--- All Students ---\n");
   for (int i = 0; i < count; i++) {
      printf("Name: %s, Roll Number: %d, Marks: %.2f\n", students[i].name,
students[i].rollNumber, students[i].marks);
   }
}

void findStudentByRollNumber(struct Student students[], int count) {
   int rollNumber, found = 0;
   printf("\nEnter roll number to search: ");
   scanf("%d", &rollNumber);
   for (int i = 0; i < count; i++) {
      if (students[i].rollNumber == rollNumber) {
         printf("Name: %s, Roll Number: %d, Marks: %.2f\n", students[i].name,
students[i].rollNumber, students[i].marks);
         found = 1;
         break;
      }
   }
   if (!found) {
      printf("Student with roll number %d not found.\n", rollNumber);
   }
}

void calculateAverageMarks(struct Student students[], int count) {
   if (count == 0) {
      printf("\nNo student records available to calculate average.\n");
      return;
   }
   float totalMarks = 0;
   for (int i = 0; i < count; i++) {
```

```
        totalMarks += students[i].marks;
    }
    printf("\nAverage Marks: %.2f\n", totalMarks / count);
}
```

OUTPUT

```
--- Student Record Management System ---
1. Add Student
2. Display All Students
3. Find Student by Roll Number
4. Calculate Average Marks
5. Exit
Enter your choice: 1

Enter student name: MEGHA
Enter roll number: 10
Enter marks: 40
Student added successfully!

--- Student Record Management System ---
1. Add Student
2. Display All Students
3. Find Student by Roll Number
4. Calculate Average Marks
5. Exit
Enter your choice: 1

Enter student name: ARSHA
Enter roll number: 11
Enter marks: 41
Student added successfully!

--- Student Record Management System ---
1. Add Student
2. Display All Students
3. Find Student by Roll Number
4. Calculate Average Marks
5. Exit
Enter your choice: 1

Enter student name: AISWARYA
Enter roll number: 13
Enter marks: 38
Student added successfully!
```

--- Student Record Management System ---
1. Add Student
2. Display All Students
3. Find Student by Roll Number
4. Calculate Average Marks
5. Exit
Enter your choice: 1

Enter student name: SREETHU
Enter roll number: 18
Enter marks: 40
Student added successfully!

--- Student Record Management System ---
1. Add Student
2. Display All Students
3. Find Student by Roll Number
4. Calculate Average Marks
5. Exit
Enter your choice: 4

Average Marks: 39.75

--- Student Record Management System ---
1. Add Student
2. Display All Students
3. Find Student by Roll Number
4. Calculate Average Marks
5. Exit
Enter your choice: 2

--- All Students ---
Name: MEGHA, Roll Number: 10, Marks: 40.00
Name: ARSHA, Roll Number: 11, Marks: 41.00
Name: AISWARYA, Roll Number: 13, Marks: 38.00
Name: SREETHU, Roll Number: 18, Marks: 40.00

--- Student Record Management System ---
1. Add Student
2. Display All Students
3. Find Student by Roll Number
4. Calculate Average Marks
5. Exit
Enter your choice: 5
Exiting the program.