Experiment No.4:

Student Database Management

You are developing a student result management system. The database should support updating records, adding new entries, searching for specific students, and sorting based on performance.Using an array of structures, implement a student database with attributes: roll no,name, program, course, subject marks, total, and average. Support operations: display,search, and sort. (Students can additionally perform modify, append.)

```
#include <stdio.h>

#include <string.h>


#define MAX 100


struct Student {

   int roll_no;

   char name[50];

   char program[30];

   char course[30];

   int marks[3];

   int total;

   float average;

};


struct Student students[MAX];

int count = 0;


// Function to add new student

void addStudent() {

   if(count >= MAX) {
```

```
    printf("Database full!\n");
    return;
}

printf("\nEnter details for student %d:\n", count + 1);
printf("Roll No: ");
scanf("%d", &students[count].roll_no);

printf("Name: ");
scanf(" %[^\n]", students[count].name); // to read full name with spaces

printf("Program: ");
scanf(" %[^\n]", students[count].program);

printf("Course: ");
scanf(" %[^\n]", students[count].course);

printf("Enter marks for 3 subjects:\n");
students[count].total = 0;
for(int i = 0; i < 3; i++) {
    printf("Subject %d: ", i + 1);
    scanf("%d", &students[count].marks[i]);
    students[count].total += students[count].marks[i];
}

students[count].average = students[count].total / 3.0;
count++;
```

```
    printf("Student added successfully!\n");
}

// Function to display all students
void displayStudents() {
    if(count == 0) {
        printf("No student records to display.\n");
        return;
    }

    printf("\nStudent Records:\n");
    for(int i = 0; i < count; i++) {
        printf("\nRoll No: %d\n", students[i].roll_no);
        printf("Name: %s\n", students[i].name);
        printf("Program: %s\n", students[i].program);
        printf("Course: %s\n", students[i].course);
        printf("Marks: %d, %d, %d\n", students[i].marks[0], students[i].marks[1], students[i].marks[2]);
        printf("Total: %d\n", students[i].total);
        printf("Average: %.2f\n", students[i].average);
    }
}

// Function to search student by roll number
void searchStudent() {
    int roll, found = 0;
    printf("Enter roll number to search: ");
    scanf("%d", &roll);
```

```
    for(int i = 0; i < count; i++) {
        if(students[i].roll_no == roll) {
            printf("\nStudent Found:\n");
            printf("Name: %s\n", students[i].name);
            printf("Program: %s\n", students[i].program);
            printf("Course: %s\n", students[i].course);
            printf("Total: %d, Average: %.2f\n", students[i].total, students[i].average);
            found = 1;
            break;
        }
    }

    if(!found) {
        printf("Student with roll number %d not found.\n", roll);
    }
}

// Function to modify a student record
void modifyStudent() {
    int roll, found = 0;
    printf("Enter roll number to modify: ");
    scanf("%d", &roll);

    for(int i = 0; i < count; i++) {
        if(students[i].roll_no == roll) {
            printf("Enter new name: ");
```

```
            scanf(" %[^\n]", students[i].name);

            printf("Enter new program: ");
            scanf(" %[^\n]", students[i].program);

            printf("Enter new course: ");
            scanf(" %[^\n]", students[i].course);

            printf("Enter new marks for 3 subjects:\n");
            students[i].total = 0;
            for(int j = 0; j < 3; j++) {
                printf("Subject %d: ", j + 1);
                scanf("%d", &students[i].marks[j]);
                students[i].total += students[i].marks[j];
            }

            students[i].average = students[i].total / 3.0;
            printf("Record updated successfully.\n");
            found = 1;
            break;
        }
    }

    if(!found) {
        printf("Student with roll number %d not found.\n", roll);
    }
}
```

```
// Function to sort students by total marks
void sortStudentsByTotal() {
    struct Student temp;
    for(int i = 0; i < count - 1; i++) {
        for(int j = i + 1; j < count; j++) {
            if(students[i].total < students[j].total) {
                temp = students[i];
                students[i] = students[j];
                students[j] = temp;
            }
        }
    }
    printf("Students sorted by total marks (highest first).\n");
}

int main() {
    int choice;

    do {
        printf("\nStudent Database Menu:\n");
        printf("1. Add Student\n");
        printf("2. Display All Students\n");
        printf("3. Search Student by Roll No\n");
        printf("4. Modify Student Record\n");
        printf("5. Sort Students by Total Marks\n");
        printf("6. Exit\n");
```

```
        printf("Enter choice: ");
        scanf("%d", &choice);

        switch(choice) {
            case 1: addStudent(); break;
            case 2: displayStudents(); break;
            case 3: searchStudent(); break;
            case 4: modifyStudent(); break;
            case 5: sortStudentsByTotal(); break;
            case 6: printf("Exiting program.\n"); break;
            default: printf("Invalid choice. Try again.\n");
        }

    } while(choice != 6);

    return 0;
}
```