Project Title:

Student Record Management System

Objective:

To create a simple C program to manage student records, including storing, viewing, editing, and deleting information.

Features:

1. Add student records (name, roll number, marks, etc.).
2. View all student records.
3. Edit existing student records.
4. Delete a specific student record.

Program Logic:

1. Use a **structure** to define a student's attributes, such as name, roll number, and marks.
2. Store student records in an **array of structures**.
3. Use file handling to make the data persistent.
4. Implement menu-driven functionality:
    - **Add Record**: Add a new record to the array.
    - **View Records**: Display all records.
    - **Edit Record**: Search for a record by roll number and modify it.
    - **Delete Record**: Remove a record by roll number.
5. Use functions for modularity.

Code:

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

#define MAX_STUDENTS 100
```

```
// Define the structure for a student
typedef struct {
    int rollNumber;
    char name[50];
    float marks;
} Student;

Student students[MAX_STUDENTS];
int studentCount = 0;

// Function prototypes
void addStudent();
void viewStudents();
void editStudent();
void deleteStudent();
void saveToFile();
void loadFromFile();

int main() {
    int choice;

    // Load records from file
    loadFromFile();

    while (1) {
        printf("\nStudent Record Management System\n");
        printf("1. Add Student\n");
        printf("2. View Students\n");
        printf("3. Edit Student\n");
        printf("4. Delete Student\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                addStudent();
                break;
            case 2:
                viewStudents();
                break;
            case 3:
                editStudent();
                break;
```

```
            case 4:
                deleteStudent();
                break;
            case 5:
                saveToFile();
                printf("Exiting... Records saved.\n");
                exit(0);
            default:
                printf("Invalid choice! Please try again.\n");
        }
    }
    return 0;
}

// Function to add a student
void addStudent() {
    if (studentCount >= MAX_STUDENTS) {
        printf("Error: Cannot add more students.\n");
        return;
    }

    Student newStudent;
    printf("Enter roll number: ");
    scanf("%d", &newStudent.rollNumber);
    printf("Enter name: ");
    scanf(" %[^\n]", newStudent.name);  // To read string with spaces
    printf("Enter marks: ");
    scanf("%f", &newStudent.marks);

    students[studentCount++] = newStudent;
    printf("Student added successfully.\n");
}

// Function to view all students
void viewStudents() {
    if (studentCount == 0) {
        printf("No student records available.\n");
        return;
    }

    printf("\n%-10s %-20s %-10s\n", "Roll No", "Name", "Marks");
    for (int i = 0; i < studentCount; i++) {
        printf("%-10d %-20s %-10.2f\n", students[i].rollNumber, students[i].name,
students[i].marks);
```

```
    }
}

// Function to edit a student's record
void editStudent() {
    int rollNumber, found = 0;
    printf("Enter roll number of the student to edit: ");
    scanf("%d", &rollNumber);

    for (int i = 0; i < studentCount; i++) {
        if (students[i].rollNumber == rollNumber) {
            printf("Editing record for %s.\n", students[i].name);
            printf("Enter new name: ");
            scanf(" %[^\n]", students[i].name);
            printf("Enter new marks: ");
            scanf("%f", &students[i].marks);
            printf("Record updated successfully.\n");
            found = 1;
            break;
        }
    }

    if (!found) {
        printf("Error: No record found with roll number %d.\n", rollNumber);
    }
}

// Function to delete a student's record
void deleteStudent() {
    int rollNumber, found = 0;
    printf("Enter roll number of the student to delete: ");
    scanf("%d", &rollNumber);

    for (int i = 0; i < studentCount; i++) {
        if (students[i].rollNumber == rollNumber) {
            for (int j = i; j < studentCount - 1; j++) {
                students[j] = students[j + 1];
            }
            studentCount--;
            printf("Record deleted successfully.\n");
            found = 1;
            break;
        }
    }
```

```
    if (!found) {
        printf("Error: No record found with roll number %d.\n", rollNumber);
    }
}

// Function to save student records to a file
void saveToFile() {
    FILE *file = fopen("students.dat", "wb");
    if (file == NULL) {
        printf("Error: Could not save records to file.\n");
        return;
    }
    fwrite(&studentCount, sizeof(int), 1, file);
    fwrite(students, sizeof(Student), studentCount, file);
    fclose(file);
}

// Function to load student records from a file
void loadFromFile() {
    FILE *file = fopen("students.dat", "rb");
    if (file == NULL) {
        return;  // File doesn't exist yet
    }
    fread(&studentCount, sizeof(int), 1, file);
    fread(students, sizeof(Student), studentCount, file);
    fclose(file);
}
```

Explanation:

1. **Structure Definition**:
    - `Student` structure stores the roll number, name, and marks.
2. **Data Persistence**:
    - Records are saved to and loaded from a binary file (`students.dat`).
3. **Menu-driven Program**:
    - Users can choose operations from a menu.
4. **Functions**:
    - Modularize the code for adding, viewing, editing, and deleting records.

How to Run:

1. Compile the program using a C compiler, e.g., `gcc student_management.c -o student_management`.
2. Run the executable: `./student_management`.
3. Follow the menu options to manage student records.

Output Sample:

```
Student Record Management System
1. Add Student
2. View Students
3. Edit Student
4. Delete Student
5. Exit
Enter your choice: 1
Enter roll number: 101
Enter name: John Doe
Enter marks: 85.5
Student added successfully.

Enter your choice: 2
Roll No    Name            Marks
101        John Doe        85.50

Enter your choice: 5
Exiting... Records saved.
```

This project demonstrates the fundamentals of structured programming, file handling, and array manipulation in C.