# Student Record Management System

## Algorithm

Step 1: Start  
Step 2: Display Main Menu with options:  
 1. Add Student  
 2. View Students  
 3. Search Student  
 4. Update Student  
 5. Delete Student  
 6. Exit  
Step 3: Read user's choice  
Step 4: Use switch-case to perform selected operation:  
 Case 1: Call addStudent()  
 - Take roll, name, marks  
 - Save to file  
 Case 2: Call viewStudents()  
 - Read and display all records from file  
 Case 3: Call searchStudent()  
 - Search by roll number  
 Case 4: Call updateStudent()  
 - Search and update record by roll  
 Case 5: Call deleteStudent()  
 - Remove record from file using temp file  
 Case 6: Exit program  
Step 5: Repeat until user chooses Exit  
Step 6: End

## C Code

#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
  
struct Student {  
 int roll;  
 char name[50];  
 float marks;  
};  
  
void addStudent();  
void viewStudents();  
void searchStudent();  
void deleteStudent();  
void updateStudent();  
void saveToFile(struct Student s);  
void header();  
void clearInputBuffer();  
  
FILE \*fp;  
  
int main() {  
 int choice;  
 do {  
 header();  
 printf("\n1. Add Student\n");  
 printf("2. View All Students\n");  
 printf("3. Search Student by Roll No\n");  
 printf("4. Update Student Record\n");  
 printf("5. Delete Student Record\n");  
 printf("6. Exit\n");  
 printf("Enter your choice: ");  
 scanf("%d", &choice);  
 clearInputBuffer();  
 switch (choice) {  
 case 1: addStudent(); break;  
 case 2: viewStudents(); break;  
 case 3: searchStudent(); break;  
 case 4: updateStudent(); break;  
 case 5: deleteStudent(); break;  
 case 6: printf("\nExiting the system. Goodbye!\n"); break;  
 default: printf("Invalid choice! Please try again.\n");  
 }  
 printf("\nPress Enter to continue...");  
 getchar();  
 } while (choice != 6);  
 return 0;  
}  
  
void header() {  
 system("cls");  
 printf("====================================\n");  
 printf(" STUDENT RECORD MANAGEMENT SYSTEM \n");  
 printf("====================================\n");  
}  
  
void clearInputBuffer() {  
 int c;  
 while ((c = getchar()) != '\n' && c != EOF);  
}  
  
void saveToFile(struct Student s) {  
 fp = fopen("students.dat", "ab");  
 if (fp == NULL) {  
 printf("Error opening file!\n");  
 return;  
 }  
 fwrite(&s, sizeof(struct Student), 1, fp);  
 fclose(fp);  
}  
  
void addStudent() {  
 struct Student s;  
 printf("Enter Roll Number: ");  
 scanf("%d", &s.roll);  
 clearInputBuffer();  
 printf("Enter Name: ");  
 fgets(s.name, sizeof(s.name), stdin);  
 s.name[strcspn(s.name, "\n")] = '\0';  
 printf("Enter Marks: ");  
 scanf("%f", &s.marks);  
 saveToFile(s);  
 printf("Student added successfully.\n");  
}  
  
void viewStudents() {  
 struct Student s;  
 fp = fopen("students.dat", "rb");  
 if (fp == NULL) {  
 printf("No records found.\n");  
 return;  
 }  
 printf("\n%-10s %-30s %-10s\n", "Roll No", "Name", "Marks");  
 printf("--------------------------------------------------------\n");  
 while (fread(&s, sizeof(struct Student), 1, fp)) {  
 printf("%-10d %-30s %-10.2f\n", s.roll, s.name, s.marks);  
 }  
 fclose(fp);  
}  
  
void searchStudent() {  
 int roll, found = 0;  
 struct Student s;  
 printf("Enter roll number to search: ");  
 scanf("%d", &roll);  
 fp = fopen("students.dat", "rb");  
 if (fp == NULL) {  
 printf("File not found.\n");  
 return;  
 }  
 while (fread(&s, sizeof(struct Student), 1, fp)) {  
 if (s.roll == roll) {  
 printf("Record Found:\n");  
 printf("Roll No: %d\n", s.roll);  
 printf("Name : %s\n", s.name);  
 printf("Marks : %.2f\n", s.marks);  
 found = 1;  
 break;  
 }  
 }  
 if (!found) {  
 printf("Record not found.\n");  
 }  
 fclose(fp);  
}  
  
void updateStudent() {  
 int roll, found = 0;  
 struct Student s;  
 FILE \*temp;  
 printf("Enter roll number to update: ");  
 scanf("%d", &roll);  
 clearInputBuffer();  
 fp = fopen("students.dat", "rb");  
 temp = fopen("temp.dat", "wb");  
 if (fp == NULL || temp == NULL) {  
 printf("File error!\n");  
 return;  
 }  
 while (fread(&s, sizeof(struct Student), 1, fp)) {  
 if (s.roll == roll) {  
 found = 1;  
 printf("Enter new name: ");  
 fgets(s.name, sizeof(s.name), stdin);  
 s.name[strcspn(s.name, "\n")] = '\0';  
 printf("Enter new marks: ");  
 scanf("%f", &s.marks);  
 }  
 fwrite(&s, sizeof(struct Student), 1, temp);  
 }  
 fclose(fp);  
 fclose(temp);  
 remove("students.dat");  
 rename("temp.dat", "students.dat");  
 if (found)  
 printf("Record updated successfully.\n");  
 else  
 printf("Record not found.\n");  
}  
  
void deleteStudent() {  
 int roll, found = 0;  
 struct Student s;  
 FILE \*temp;  
 printf("Enter roll number to delete: ");  
 scanf("%d", &roll);  
 fp = fopen("students.dat", "rb");  
 temp = fopen("temp.dat", "wb");  
 if (fp == NULL || temp == NULL) {  
 printf("File error!\n");  
 return;  
 }  
 while (fread(&s, sizeof(struct Student), 1, fp)) {  
 if (s.roll != roll) {  
 fwrite(&s, sizeof(struct Student), 1, temp);  
 } else {  
 found = 1;  
 }  
 }  
 fclose(fp);  
 fclose(temp);  
 remove("students.dat");  
 rename("temp.dat", "students.dat");  
 if (found)  
 printf("Record deleted successfully.\n");  
 else  
 printf("Record not found.\n");  
}