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

struct Student {
    int id;
    char name[50];
    float marks[5];
    float total;
    float average;
    char grade;
    struct Student *next;
};

typedef struct Student Student;

// Function prototypes
void addStudent(Student **head);
void displayStudents(Student *head);
void calculateGrade(Student *s);
void sortStudents(Student **head);
void searchStudent(Student *head, int id);

int main() {
    Student *head = NULL;
    int choice, id;

    while (1) {
        printf("\n=== Transparent Evaluation System (SDG 4) ===\n");
        printf("1. Add Student Record\n");
        printf("2. Display All Students\n");
        printf("3. Sort by Total Marks (Fair Ranking)\n");
        printf("4. Search Student by ID\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                addStudent(&head);
                break;
            case 2:
                displayStudents(head);
                break;
```

```
            case 3:
                sortStudents(&head);
                printf("Students sorted by total marks (high to low).\n");
                break;
            case 4:
                printf("Enter student ID to search: ");
                scanf("%d", &id);
                searchStudent(head, id);
                break;
            case 5:
                printf("Exiting... ensuring fair evaluation is our priority!\n");
                return 0;
            default:
                printf("Invalid choice. Try again.\n");
        }
    }
}

void addStudent(Student **head) {
    Student *newNode = (Student *)malloc(sizeof(Student));
    printf("Enter Student ID: ");
    scanf("%d", &newNode->id);
    printf("Enter Name: ");
    scanf(" %[^\n]", newNode->name);

    newNode->total = 0;
    for (int i = 0; i < 5; i++) {
        printf("Enter marks in subject %d: ", i + 1);
        scanf("%f", &newNode->marks[i]);
        newNode->total += newNode->marks[i];
    }

    newNode->average = newNode->total / 5.0;
    calculateGrade(newNode);

    newNode->next = *head;
    *head = newNode;

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

void displayStudents(Student *head) {
    if (head == NULL) {
        printf("No student records found.\n");
```

```
        return;
    }

    printf("\n%-5s %-20s %-10s %-10s %-6s\n", "ID", "Name", "Total", "Average", "Grade");
    printf("-------------------------------------------------------\n");

    Student *temp = head;
    while (temp != NULL) {
        printf("%-5d %-20s %-10.2f %-10.2f %-6c\n",
               temp->id, temp->name, temp->total, temp->average, temp->grade);
        temp = temp->next;
    }
}

void calculateGrade(Student *s) {
    if (s->average >= 90)
        s->grade = 'A';
    else if (s->average >= 75)
        s->grade = 'B';
    else if (s->average >= 60)
        s->grade = 'C';
    else if (s->average >= 50)
        s->grade = 'D';
    else
        s->grade = 'F';
}

void sortStudents(Student **head) {
    if (*head == NULL || (*head)->next == NULL)
        return;

    Student *i, *j;
    for (i = *head; i != NULL; i = i->next) {
        for (j = i->next; j != NULL; j = j->next) {
            if (i->total < j->total) {
                // Swap student data
                Student temp = *i;
                *i = *j;
                *j = temp;

                // Fix linked list pointers
                Student *t = i->next;
                i->next = j->next;
                j->next = t;
```

```
            }
        }
    }
}

void searchStudent(Student *head, int id) {
    Student *temp = head;
    while (temp != NULL) {
        if (temp->id == id) {
            printf("\nRecord Found:\n");
            printf("ID: %d\nName: %s\nTotal: %.2f\nAverage: %.2f\nGrade: %c\n",
                  temp->id, temp->name, temp->total, temp->average, temp->grade);
            return;
        }
        temp = temp->next;
    }
    printf("? Student with ID %d not found.\n", id);
}
```