Student Management System

Objectives

- To develop a student management system for storing, searching, and sorting student data.
- To practice working with text files for data storage and retrieval.
- To implement structured programming with user-defined functions for modularity.
- To apply sorting algorithms and searching techniques in a real-world scenario.

Problem Description

Create a C program to manage student information in a text file. The program should:

- Allow users to add new students to the file.
- Allow users to edit a student by ID.
- Allow users to delete a student by ID.
- Enable users to search for students by their ID or last name.
- Display the list of students sorted in ascending order by last name.

Specific Requirements

- Data format: Each student record should include
  - ID: Integer (e.g., 101)
  - First name: String (e.g., John)
  - Last name: String (e.g., Doe)
  - GPA: Float (e.g., 3.75)
- File format: The student data should be stored in a plain text file (student.txt) with the following format for each record: `ID, First name, Last name, GPA`

Menu Options

- Add a new student
- Edit a student by ID
- Delete a student by ID
- Search for a student by ID
- Search for a student by last name
- Display all students sorted by last name

Output Requirements

- Properly formatted table of student data when displaying.
- Meaningful messages for successful or unsuccessful search operations.

`/* the end */`

student_management_system_template.c

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

void print_menu();
int make_choice(int min, int max);
void add();
void edit();
void delete();
void search_by_id();
void search_by_name();
void display();

int main(void)
{
    void (*funs[7])(void) = {NULL, &add, &edit, &delete, &search_by_id, &search_by_name, &display};
    int choice;

    do
    {
        choice = make_choice(0, 6);
        if (choice)
        {
            funs[choice]();
            printf("\nPress any key to return to the menu.");
            getchar();
        }
    } while (choice);

    return 0;
}

void print_menu()
{
    system("clear"); // for macos, linux
    // system("cls"); // for windows
    printf("STUDENT MANAGEMENT SYSTEM\n");
    printf("-------------------------\n");
    printf("1. Add new student\n");
    printf("2. Edit a student\n");
    printf("3. Delete a student\n");
    printf("4. Search for a student by ID\n");
    printf("5. Search for a student by last name\n");
    printf("6. Display all students sorted by last name\n");
    printf("0. Exit\n");
}

int make_choice(int min, int max)
{
    int not_valid = 1, num, scanf_ret;
    char c;

    print_menu();
    fflush(stdin);
    do
    {
        printf("\nEnter your choice: ");
        scanf_ret = scanf("%d%c", &num, &c);
        if (scanf_ret < 2 || c != '\n')
        {
            print_menu();
            printf("\nYour choice is not valid. Please try again!\n");
            fflush(stdin);
        }
        else if (num < min || num > max)
        {
            print_menu();
            printf("\nYour choice is not valid. Please try again!\n");
        }
        else
            not_valid = 0;
    } while (not_valid);

    system("clear");

    return num;
}

void add() { printf("TODO: Implement the add() function.\n"); }

void edit() { printf("TODO: Implement the edit() function.\n"); }

void delete() { printf("TODO: Implement the delete() function.\n"); }

void search_by_id() { printf("TODO: Implement the search_by_id() function.\n"); }

void search_by_name() { printf("TODO: Implement the search_by_name() function.\n"); }

void display() { printf("TODO: Implement the display() function.\n"); }
```