**1. Libraries and Definitions:**

c

CopyEdit

#include <stdio.h> // Standard input-output library (for printing, scanning, file operations)

#include <stdlib.h> // Standard library (for system functions like `system()` and `exit()`)

#include <string.h> // String manipulation functions (though not used directly in this code)

* **stdio.h**: Provides functionality for input and output operations (like printing and scanning).
* **stdlib.h**: Contains functions for memory allocation, process control, and other utilities. Here, it's used for system() to clear the console screen and exit() to terminate the program.
* **string.h**: Although it is included, it's not used in this particular code. It can be useful if you need to manipulate strings (like strcpy or strcmp), but it isn't necessary here.

c

CopyEdit

#ifdef \_WIN32

#define CLEAR\_SCREEN "cls"

#else

#define CLEAR\_SCREEN "clear"

#endif

* This part checks the operating system (OS) to set a proper command for clearing the screen. cls is used for Windows, while clear is used for Unix-based systems (Linux, macOS). This allows your program to work on both types of systems.

c

CopyEdit

#define FILE\_NAME "students.txt"

* **FILE\_NAME** is a macro that defines the file name where the student records will be stored.

**2. Structure Definition:**

c

CopyEdit

struct Student {

int roll\_no;

char first\_name[25];

char last\_name[25];

float marks;

};

* A **structure** (struct) is used to store student details. It includes:
  + roll\_no: Integer for the student's roll number.
  + first\_name: Array of characters to store the student's first name.
  + last\_name: Array of characters to store the student's last name.
  + marks: A floating-point number to store the student's marks.

The structure helps group all the related data of a student together.

**3. Function Prototypes:**

c

CopyEdit

void addStudent();

void viewStudents();

void searchStudent();

* These are function prototypes, indicating that the program will define three functions:
  + addStudent(): For adding student records to the file.
  + viewStudents(): For displaying all student records from the file.
  + searchStudent(): For searching and displaying a student record based on roll number.

**4. Main Function:**

c

CopyEdit

int main() {

int choice;

do {

system(CLEAR\_SCREEN);

printf("\nStudent Record Management System\n");

printf("1. Add Student\n");

printf("2. View Students\n");

printf("3. Search Student\n");

printf("4. Exit\n");

printf("\nEnter your choice: ");

scanf("%d", &choice);

* **main()** is the entry point of the program.
* The program starts with a loop that continually presents the user with a menu, allowing them to choose what action to take (add a student, view students, search for a student, or exit).
* **system(CLEAR\_SCREEN)** is used to clear the console screen for a fresh look.
* The choice is captured using scanf(), which takes input from the user and stores it in the choice variable.

c

CopyEdit

switch (choice) {

case 1:

addStudent();

break;

case 2:

viewStudents();

break;

case 3:

searchStudent();

break;

case 4:

printf("Exiting the program.\n");

exit(0);

default:

printf("Invalid choice! Please try again.\n");

}

* The switch statement checks the user's choice and calls the corresponding function.
  + **addStudent()** is called if the user chooses to add a student.
  + **viewStudents()** is called to view all students.
  + **searchStudent()** is called to search a student by roll number.
  + If the user enters an invalid choice, an error message is shown.

**5. addStudent() Function:**

c

CopyEdit

void addStudent() {

FILE \*file = fopen(FILE\_NAME, "a");

if (file == NULL) {

printf("Error opening file!\n");

return;

}

* **fopen(FILE\_NAME, "a")**: Opens the file in append mode ("a"), meaning new data will be added at the end of the file. If the file doesn't exist, it will be created.
* If the file can't be opened (perhaps due to permissions or path issues), it prints an error message and returns.

c

CopyEdit

struct Student s;

int num\_students;

char confirm;

printf("How many students do you want to add? ");

scanf("%d", &num\_students);

* A struct Student s is defined to store the student data for the current entry.
* The program asks how many students the user wants to add and stores the number in num\_students.

c

CopyEdit

for (int i = 0; i < num\_students; i++) {

printf("\nEnter details for student %d:\n", i + 1);

printf("Enter Roll No: ");

scanf("%d", &s.roll\_no);

printf("Enter First Name: ");

scanf("%s", s.first\_name);

printf("Enter Last Name: ");

scanf("%s", s.last\_name);

printf("Enter Marks: ");

scanf("%f", &s.marks);

* A loop runs for each student. It asks the user for the student's roll number, first name, last name, and marks.
* **scanf("%s", s.first\_name)** and **scanf("%s", s.last\_name)** store the input names in the corresponding fields.

c

CopyEdit

printf("\nAre you sure you want to add this student? (y/n): ");

scanf(" %c", &confirm);

* The program asks for confirmation before adding each student.
* The " %c" format specifier with a leading space helps to clear any leftover newline characters in the input buffer from previous scanf() calls.

c

CopyEdit

if (confirm == 'y' || confirm == 'Y') {

fprintf(file, "%d %s %s %.2f\n", s.roll\_no, s.first\_name, s.last\_name, s.marks);

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

} else {

printf("Student record not added.\n");

i--; // Decrement counter to re-ask for this student

}

}

fclose(file);

* If the user confirms (presses y or Y), the student's details are written to the file using fprintf().
* If the user cancels (presses any other key), the student is not added, and the loop counter i is decremented, so it asks again for this student.

c

CopyEdit

// Ask user to press 'q' to clear the screen

printf("\nPress 'q' to clear the screen and return to the menu: ");

char ch;

scanf(" %c", &ch);

if (ch == 'q' || ch == 'Q') {

system(CLEAR\_SCREEN);

}

}

* After the student data is added, the user is asked to press 'q' to clear the screen and return to the main menu.

**6. viewStudents() Function:**

c

CopyEdit

void viewStudents() {

FILE \*file = fopen(FILE\_NAME, "r");

if (file == NULL) {

printf("No records found!\n");

return;

}

* **fopen(FILE\_NAME, "r")** opens the file in read mode ("r"). If the file doesn't exist or is empty, an error message is shown.

c

CopyEdit

struct Student s;

printf("\n%-10s %-15s %-15s %-6s\n", "Roll No", "First Name", "Last Name", "Marks");

printf("-----------------------------------------------\n");

* The program prints a header to label the columns for student details.
* **%-10s**, **%-15s**, etc., are used to format the output with specific field widths, making the output look neat.

c

CopyEdit

while (fscanf(file, "%d %24s %24s %f", &s.roll\_no, s.first\_name, s.last\_name, &s.marks) == 4) {

printf("%-10d %-15s %-15s %-6.2f\n", s.roll\_no, s.first\_name, s.last\_name, s.marks);

}

* The fscanf() function is used to read data from the file. It reads the student's roll number, first name, last name, and marks, and prints them out.
* The loop continues until all records are read.

c

CopyEdit

fclose(file);

* After reading all records, the file is closed.

c

CopyEdit

printf("\nPress 'q' to clear the screen and return to the menu: ");

char ch;

scanf(" %c", &ch);

if (ch == 'q' || ch == 'Q') {

system(CLEAR\_SCREEN);

}

}

* Similar to addStudent(), the user is asked to press 'q' to clear the screen and return to the menu.

**7. searchStudent() Function:**

c

CopyEdit

void searchStudent() {

FILE \*file = fopen(FILE\_NAME, "r");

if (file == NULL) {

printf("No records found!\n");

return;

}

* Opens the file in read mode, just like in viewStudents(), but this time for searching.

c

CopyEdit

int roll\_no;

printf("Enter Roll No to search: ");

scanf("%d", &roll\_no);

* The user is asked to enter the roll number of the student they want to search for.

c

CopyEdit

struct Student s;

int found = 0;

while (fscanf(file, "%d %24s %24s %f", &s.roll\_no, s.first\_name, s.last\_name, &s.marks) == 4) {

if (s.roll\_no == roll\_no) {

printf("Record Found:\n");

printf("%-10s %-15s %-15s %-6s\n", "Roll No", "First Name", "Last Name", "Marks");

printf("-----------------------------------------------\n");

printf("%-10d %-15s %-15s %-6.2f\n", s.roll\_no, s.first\_name, s.last\_name, s.marks);

found = 1;

break;

}

}

* The fscanf() reads through the records and compares each student's roll number with the one the user entered.
* If a match is found, the student’s details are printed, and the found flag is set to 1, exiting the loop.

c

CopyEdit

if (!found) {

printf("Student with Roll No %d not found!\n", roll\_no);

}

fclose(file);

* If no record was found, it prints a message indicating that the student was not found.

c

CopyEdit

printf("\nPress 'q' to clear the screen and return to the menu: ");

char ch;

scanf(" %c", &ch);

if (ch == 'q' || ch == 'Q') {

system(CLEAR\_SCREEN);

}

}

* Again, prompts the user to press 'q' to clear the screen and return to the main menu.

**Final Thoughts:**

This code creates a simple text-based student record management system where users can add, view, and search student records. The system stores records in a text file and uses a simple structure to handle the student data.

Let me know if you need further clarifications or improvements!

4o mini