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

// --- 1. Data Structures ---
// Define a structure for a single student
struct Student {
    int id;
    char name[50];
};

// Define a structure for an attendance record
struct AttendanceRecord {
    int student_id;
    char date[12]; // Format: YYYY-MM-DD
    int status;    // 1 for Present, 0 for Absent
};

// Global file name for persistence
const char *STUDENT_FILE = "students.dat";
const char *ATTENDANCE_FILE = "attendance.dat";

// --- 2. Function Prototypes ---
void add_student();
void mark_attendance();
void view_attendance();
void display_menu();

// --- 3. Main Function ---
int main() {
    int choice;
    printf("Attendance Management System (C Program)\n");

    do {
        display_menu();
        printf("Enter your choice: ");
        scanf("%d", &choice);
        // Consume the newline character left by scanf
        while (getchar() != '\n');

        switch (choice) {
            case 1:
                add_student();
                break;
            case 2:
                mark_attendance();
                break;
            case 3:
                view_attendance();
                break;
            case 4:
                printf("Exiting the system. Goodbye!\n");
                break;
            default:
                printf("Invalid choice. Please try again.\n");
```

```
        }
    } while (choice != 4);

    return 0;
}

// --- 4. Menu and Utility Functions ---
void display_menu() {
    printf("\n--- Main Menu ---\n");
    printf("1. Add New Student\n");
    printf("2. Mark Attendance\n");
    printf("3. View Student Attendance Report\n");
    printf("4. Exit\n");
}

// --- 5. Add Student Function ---
void add_student() {
    struct Student new_student;
    FILE *fp;

    printf("\n--- Add New Student ---\n");
    printf("Enter Student ID (e.g., 101): ");
    scanf("%d", &new_student.id);

    // Consume newline before reading string
    while (getchar() != '\n');

    printf("Enter Student Name: ");
    // Safely read a string up to 49 characters
    if (fgets(new_student.name, 50, stdin) == NULL) {
        printf("Error reading name.\n");
        return;
    }
    // Remove the trailing newline character from fgets
    new_student.name[strcspn(new_student.name, "\n")] = 0;

    // Open the file in append binary mode ("ab")
    fp = fopen(STUDENT_FILE, "ab");
    if (fp == NULL) {
        perror("Error opening student file");
        return;
    }

    // Write the entire structure to the file
    fwrite(&new_student, sizeof(struct Student), 1, fp);
    fclose(fp);

    printf("Student **%s (ID: %d)** added successfully.\n", new_student.name,
new_student.id);
}

// --- 6. Mark Attendance Function ---
void mark_attendance() {
    struct AttendanceRecord record;
    int status_input;
```

```
    FILE *fp;

    printf("\n--- Mark Attendance ---\n");
    printf("Enter Date (YYYY-MM-DD): ");
    // Read date string
    if (fgets(record.date, 12, stdin) == NULL) {
        printf("Error reading date.\n");
        return;
    }
    record.date[strcspn(record.date, "\n")] = 0; // Remove newline

    printf("Enter Student ID to mark: ");
    scanf("%d", &record.student_id);
    while (getchar() != '\n'); // Consume newline

    printf("Enter Status (1 for Present / 0 for Absent): ");
    scanf("%d", &status_input);
    while (getchar() != '\n'); // Consume newline

    if (status_input != 0 && status_input != 1) {
        printf("Invalid status input. Attendance not recorded.\n");
        return;
    }
    record.status = status_input;

    // Open the file in append binary mode ("ab")
    fp = fopen(ATTENDANCE_FILE, "ab");
    if (fp == NULL) {
        perror("Error opening attendance file");
        return;
    }

    // Write the attendance record to the file
    fwrite(&record, sizeof(struct AttendanceRecord), 1, fp);
    fclose(fp);

    printf("Attendance marked for ID %d on %s: %s\n",
           record.student_id, record.date, (record.status == 1 ? "PRESENT" :
"ABSENT"));
}

// --- 7. View Attendance Report Function ---
void view_attendance() {
    int search_id;
    struct AttendanceRecord record;
    struct Student current_student;
    int total_classes = 0;
    int present_count = 0;
    FILE *fp_att, *fp_stud;
    char student_name[50] = "Unknown Student";
    int found_student = 0;

    printf("\n--- View Attendance Report ---\n");
    printf("Enter Student ID to view report: ");
    scanf("%d", &search_id);
```

```
    while (getchar() != '\n'); // Consume newline

    // First, find the student's name
    fp_stud = fopen(STUDENT_FILE, "rb");
    if (fp_stud != NULL) {
        while (fread(¤t_student, sizeof(struct Student), 1, fp_stud)) {
            if (current_student.id == search_id) {
                strcpy(student_name, current_student.name);
                found_student = 1;
                break;
            }
        }
        fclose(fp_stud);
    }

    if (!found_student) {
        printf("Student ID **%d** not found in the student registry.\n", search_id);
        return;
    }

    printf("\n*** Report for: %s (ID: %d) ***\n", student_name, search_id);

    // Now, read and count attendance records
    fp_att = fopen(ATTENDANCE_FILE, "rb");
    if (fp_att == NULL) {
        printf("No attendance records found yet.\n");
        return;
    }

    printf("Date\t\tStatus\n");
    printf("------------------------------\n");

    while (fread(&record, sizeof(struct AttendanceRecord), 1, fp_att)) {
        if (record.student_id == search_id) {
            total_classes++;
            if (record.status == 1) {
                present_count++;
            }
            printf("%s\t%s\n", record.date, (record.status == 1 ? "Present" :
"Absent"));
        }
    }

    fclose(fp_att);

    // Calculate and display the summary
    if (total_classes > 0) {
        float percentage = ((float)present_count / total_classes) * 100.0;
        printf("------------------------------\n");
        printf("Summary:\n");
        printf("Total Classes Recorded: %d\n", total_classes);
        printf("Days Present: %d\n", present_count);
        printf("Attendance Percentage: **%.2f%%**\n", percentage);
    } else {
        printf("No attendance records found for this student.\n");
```

```
        }
    }
```