**Intermediate Coding**

**Miniproject**

**Hu22csen0101658**

**S.Mrudula**

**Student Marks Calculator**

**1. Problem Statement**

This project aims to create a **Student Marks Calculator** that helps teachers manage and analyse student marks in a simple classroom setting. The application will allow a teacher to:

* Add a student’s marks for three subjects.
* Display a list of all students with their names, marks, and average scores.
* Calculate and display class-wide average scores for each subject.

This calculator provides an organized and easy way for teachers to manage basic student data, allowing them to track individual and overall class performance.

**2. Proposed Solution**

To solve this problem, we propose a C program that uses arrays and structures to store and manage student data. The program will include the following components:

1. **Data Structure**: A Student structure to store each student’s name, marks in three subjects, and their average score.
2. **Array of Structures**: An array to hold multiple Student structures, limited to a maximum of 10 students for simplicity.
3. **Functions for Key Operations**:
   * addStudent(): To add a student’s details, including their marks in each subject.
   * displayStudents(): To display all students, including their names, subject marks, and average scores.
   * calculateClassAverage(): To calculate and display the average score for each subject across the entire class.
4. **User Menu**: A menu-driven interface allowing the teacher to select an operation—adding student data, viewing all students, calculating class averages, or exiting the program.

This program will provide a user-friendly and efficient tool for teachers to input, view, and analyze student marks.

**3. Code:**

#include <stdio.h>

#include <string.h>

#define MAX\_STUDENTS 10

struct Student {

char name[50];

float marks[3];

float average;

};

void addStudent(struct Student students[], int \*count);

void displayStudents(const struct Student students[], int count);

void calculateAverages(struct Student students[], int count);

void calculateClassAverage(const struct Student students[], int count);

int main() {

struct Student students[MAX\_STUDENTS];

int count = 0;

int choice;

do {

printf("\nStudent Marks Calculator\n");

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

printf("2. Display All Students\n");

printf("3. Calculate and Display Class Average\n");

printf("4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

addStudent(students, &count);

break;

case 2:

displayStudents(students, count);

break;

case 3:

calculateClassAverage(students, count);

break;

case 4:

printf("Exiting program.\n");

break;

default:

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

}

} while (choice != 4);

return 0;

}

void addStudent(struct Student students[], int \*count) {

if (\*count >= MAX\_STUDENTS) {

printf("Cannot add more students.\n");

return;

}

printf("Enter name of student: ");

scanf("%s", students[\*count].name);

printf("Enter marks for three subjects: ");

for (int i = 0; i < 3; i++) {

scanf("%f", &students[\*count].marks[i]);

}

students[\*count].average = (students[\*count].marks[0] +

students[\*count].marks[1] +

students[\*count].marks[2]) / 3;

(\*count)++;

printf("Student added successfully.\n");

}

void displayStudents(const struct Student students[], int count) {

if (count == 0) {

printf("No students to display.\n");

return;

}

printf("\nList of Students:\n");

for (int i = 0; i < count; i++) {

printf("Name: %s\n", students[i].name);

printf("Marks: %.2f, %.2f, %.2f\n", students[i].marks[0], students[i].marks[1], students[i].marks[2]);

printf("Average: %.2f\n", students[i].average);

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

}

}

void calculateClassAverage(const struct Student students[], int count) {

if (count == 0) {

printf("No students available to calculate averages.\n");

return;

}

float subjectSum[3] = {0.0, 0.0, 0.0};

for (int i = 0; i < count; i++) {

for (int j = 0; j < 3; j++) {

subjectSum[j] += students[i].marks[j];

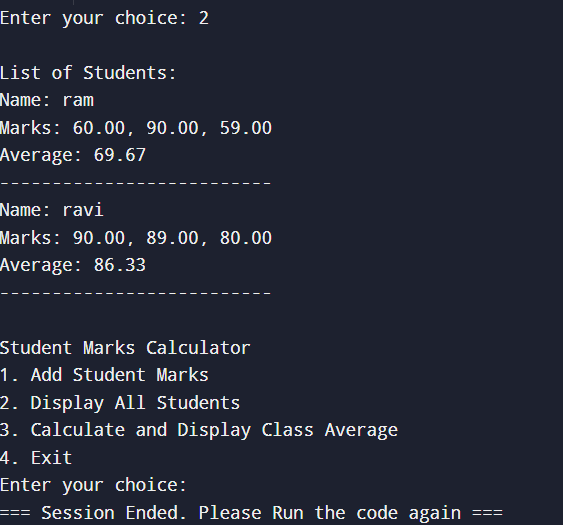
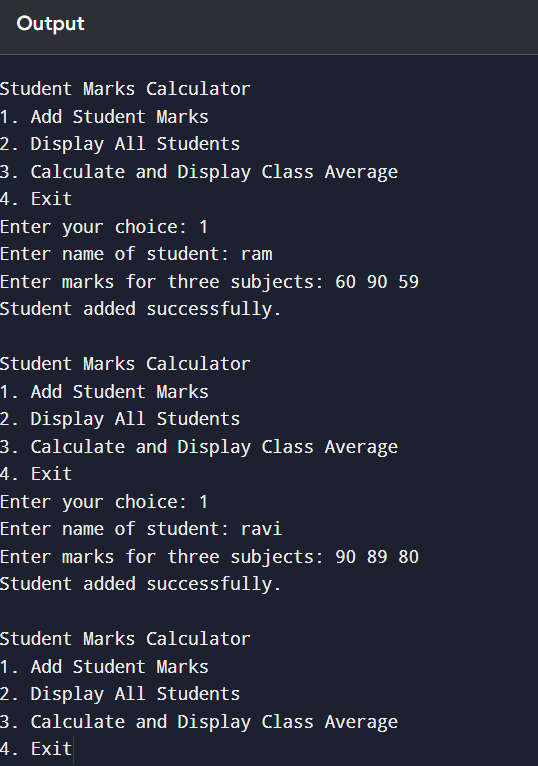
}

}

printf("\nClass Averages:\n");

for (int j = 0; j < 3; j++) {

printf("Subject %d Average: %.2f\n", j + 1, subjectSum[j] / count);

****

**4. Description of Code and Functions**

1. **Data Structure**:
   * The Student structure includes:
     + name: A string to store the student’s name.
     + marks: An array to store the student’s marks in three subjects.
     + average: A float to store the average of the student’s marks across the three subjects.
2. **Functions**:
   * addStudent(struct Student students[], int \*count):
     + This function takes an array of students and a pointer to the count of students as parameters. It allows the teacher to enter a new student's name and marks for each subject, calculates the student's average, and adds them to the list.
   * displayStudents(const struct Student students[], int count):
     + This function displays all students currently in the system. It shows each student’s name, their marks in each subject, and their calculated average.
   * calculateClassAverage(const struct Student students[], int count):
     + This function calculates the average marks for each subject across all students. It iterates through the array, summing marks for each subject and dividing by the number of students to find the class average for each subject.
3. **User Interface**:
   * The program provides a menu with options for each operation (adding students, displaying data, calculating averages, and exiting).
   * The user selects an option by entering a corresponding number. The menu ensures a straightforward interaction, looping until the user chooses to exit.

This solution offers a compact, efficient way to manage basic classroom data, making it easy for teachers to monitor both individual and class-wide academic performance. The program’s structure and simple functionality allow for possible future extensions, like adding more subjects or storing data in a file.