

```

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

#define SIZE_G 4
#define SIZE_A 30

// function prototypes
float studentAverage(float theGrades[], size_t size_g);
void courseAverage(float averages[], size_t size_a, int numOfStudents);
void classMedian(float averages[], int numOfStudents);
void highestLowest(float averages[], int numOfStudents);
void scholarship(float *averagesPtr);

int main(void) {

    // variables to control looping
    char _continue[2] = { "y" };

    unsigned int numOfStudents;

    int counter;
    counter = 0;

    // course and student information
    char day[5];
    char time[7];
    char courseName[30];
    char courseNumber[20];
    char firstName[30];
    char lastName[30];
    char studentID[12];

    float grades[SIZE_G]; // array to hold student's grades
    float averages[SIZE_A]; // array to hold all student's averages

    float grade;

    printf("%s\n", "Welcome to Grade Book!\n");

    // loop to enter and display course information
    while (_continue[0] == (tolower('y'))) {

        printf("%s", "Please enter the Course Name: ");
        scanf("%29s", courseName);

        printf("%s", "Please enter the Course Number: ");
        scanf("%19s", courseNumber);

        printf("%s", "Please enter the days class will be held: ");
        scanf("%4s", day);

        printf("%s", "Please enter the time class will meet: ");
        scanf("%6s", time);

        printf("%s", "Please enter the number of students in this course: ");
        scanf("%d", &numOfStudents);

        // loop to enter and display student information
        while (counter < numOfStudents) {

            printf("%s", "\nFirst Name: ");
            scanf("%29s", firstName);

            printf("%s", "Last Name : ");
            scanf("%29s", lastName);

            printf("%s", "ID Number : ");
            scanf("%11s", studentID);

            printf("%s", "\n    Grades: ");

            for (size_t i = 0; i < SIZE_G; i++) {
                scanf("%f", &grade);

                // disregard invalid grades
                if (grade < 0 || grade > 100) {
                    printf("%s", "\nInvalid Entry. Please enter a new grade: ");
                    scanf("%f", &grade);

                    grades[i] = grade;
                    if (i < SIZE_G - 1) {
                        printf("%s", "\t : ");
                    }
                }
                else { // enter grades into the grades array
                    grades[i] = grade;
                    if (i < SIZE_G - 1) {
                        printf("%s", "\t : ");
                    }
                }
            } // end for loop

            // calculate this student's average, store this student's average in
            // the averages array, and print the letter grade
            printf("\nAverage: %.2f\n", averages[counter] = studentAverage(grades, SIZE_G));

            if (averages[counter] >= 90) {
                printf("%s", "Letter Grade: A");
            }
            else if (averages[counter] >= 80) {
                printf("%s", "Letter Grade: B");
            }

            else if (averages[counter] >= 70) {
                printf("%s", "Letter Grade: C");
            }
            else if (averages[counter] >= 60) {
                printf("%s", "Letter Grade: D");
            }
            else {
                printf("%s", "Letter Grade: F");
            }

            // calculate scholarship
            float *averagesPtr = &averages[counter];

            if (averages[counter] >= 80) {
                scholarship(averagesPtr);
            }

            puts("");
            counter++;
        } // end nested while loop

        // display course information
        puts("\n Day    Time    Course    Number");
        puts(" ---    ---    -----    -----");
        printf(" %s    %s    %s    %s\n\n", day, time, courseName, courseNumber);

        // display the course average, class median, highest and lowest grades,
        // number of students who scored above 70 and above course average
        courseAverage(averages, SIZE_A, numOfStudents);

        classMedian(averages, numOfStudents);

        counter = 0;

        // prompt the user to enter another class or quit
        printf("\n\n%s", "Enter another course? Y for yes or N for no: ");
        scanf("%1s", _continue);
        puts("");

    } // end outer while loop

    system("pause"); // necessary for visual studios

} // end main

// function to calculate the student average
float studentAverage(float theGrades[], size_t size_g) {

    float average;

    float total;
    total = 0;

    for (size_t i = 0; i < size_g; i++) {
        total += theGrades[i];
    }
    average = total / size_g;

    return average;
}

// calculate the course average and determine how many students have an
// average >= 70 and how many have an average >= the course average
void courseAverage(float averages[], size_t size_a, int numOfStudents) {

    // calculate and print the course average
    float courseAverage;
    float total;
    total = 0;

    for (size_t i = 0; i < numOfStudents; i++) {
        total += averages[i];
    }

    printf("Course Average: %.2f\n", courseAverage = total / numOfStudents);

    // print number of students with averages above 70 and course average
    int aboveAverage;
    aboveAverage = 0;

    int aboveSeventy;
    aboveSeventy = 0;

    for (size_t i = 0; i < numOfStudents; i++) {

        if (averages[i] >= courseAverage) {
            aboveAverage++;
        }
        if (averages[i] >= 70) {
            aboveSeventy++;
        }
    }

    printf("\nStudents who scored above 70: %d", aboveSeventy);
    printf("\nStudents who scored above class average: %d", aboveAverage);
}

// function to determine the class median
void classMedian(float averages[], int numOfStudents) {

    float holder; // place holder to sort array
    int median;

    // loop to control number of passes
    for (unsigned int pass = 1; pass < numOfStudents; pass++) {

        // sort the array of averages
        for (size_t i = 0; i < numOfStudents; i++) {

            if (averages[i] > averages[i + 1]) {
                holder = averages[i];
                averages[i] = averages[i + 1];
                averages[i + 1] = holder;
            }
        }
    }
    // numOfStudents needs to keep it original value
    int factor;
    factor = numOfStudents;

    // prevent non integer values
    if (factor % 2 != 0) {
        factor++;
    }

    // prevent division by zero
    if (factor != 0 && factor > 4) {

        // determine and print class median
        median = factor / 2;
        printf("\nClass Median: %.2f", averages[median]);
    }
    else if (factor == 4) {
        median = 2;
        printf("\nClass Median: %.2f", averages[median]);
    }
    else if (factor == 3) {
        median = 1;
        printf("\nClass Median: %.2f", averages[median]);
    }
    else {
        median = 0;
        printf("\nClass Median: %.2f", averages[median]);
    }
    // call function highestLowest and pass it the sorted array
    highestLowest(averages, numOfStudents);
}

// function to determine the highest and the lowest averages
void highestLowest(float averages[], int numOfStudents) {

    // print the highest and lowest scores

    if (numOfStudents == 1) {
        printf("\nHighest average: %.2f", averages[0]);
    }
    else {
        printf("\nHighest average: %.2f", averages[numOfStudents]);
    }
    printf("\nLowest average: %.2f", averages[0]);
}

// function to determine the amount of scholarship given to students
void scholarship(float *averagesPtr) {

    float scholarship;

    // calculate and print the amount of scholarship
    if (*averagesPtr >= 90) {

        scholarship = 4 * 425;
        printf("\nScholarship Awarded: $%.2f", scholarship);
    }
    else if (*averagesPtr >= 80) {

        scholarship = 3 * 425;
        printf("\nScholarship Awarded: $%.2f", scholarship);
    }
}

```