San Francisco State University

ENGR 476

Lab 1

Alberto Gonzalez Navarro

Date performed: 9/6/2018

Date submitted:9/20/18

```
// Alberto Gonzalez Navarro
// ENGR 476 SFSU
// LAB 1
//
// This program contains 4 different functions. It reads a text file and outputs its contents
// while allowing the user in the main menu to pick which information they would like displayed.
// It will sort in ascending order using a specific subject, calculate the average and standard
// deviation, which students received a diploma, and writeback to a new file.

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

typedef struct Student_struct{
   char studentName[50];
        int studentNum;
        float subA, subB;
} Student;

int readFile(Student[], char*);
int FileWrite(Student[], char*);
void gotDiploma(Student[]);
void SubASort(Student[]);
void AvgAndDevCal(Student[]);
```

```
int readFile(Student student[], char* File_name) {

	FILE* InFile;		// opening the input file
	InFile = fopen(File_name, "r");



	if (!InFile)		// if input file could not be opened
		return -1;



	char store[30];		// skipping header file of the file
	fscanf(InFile, "%s %s %s %s %s %s %s %s\n", store, store, store, store, store, store, store,
store);


	int totalStudents = 0;		// reading each line of input file till end of file
	printf("%s\n\n", File_name);
	printf("STUDENT NAME\t STUDENT NO.\t SUBJECT A\t SUBJECT B\n");
	while (fscanf(InFile, "%s %d %f %f\n", student[totalStudents].studentName,
&student[totalStudents].studentNum,
		&student[totalStudents].subA, &student[totalStudents].subB) != EOF) {
		// printing students details
		printf("%s\t\t%d\t\t%.1f\t\t%.1f\n", student[totalStudents].studentName,
student[totalStudents].studentNum,
			student[totalStudents].subA, student[totalStudents].subB);
		totalStudents++;
	}


	fclose(InFile);		//input file close
	return 0;
}
```

```
// Function that picks out the students that got a diploma
void gotDiploma(Student student[]) {

	printf("All the students that got diploma:\n"); // printing all the students who got diploma
	printf("STUDENT NAME\t SUBJECT A\t SUBJECT B\n");

	for (size_t i = 0; i < 7; i++) {		// Going through student records

				// if student has 50 or higher in each subject
		if (student[i].subA >= 50 && student[i].subB >= 50)
			printf("%s\t\t%.1f\t\t%.1f\n", student[i].studentName, student[i].subA, student[i].subB);
	}
}

// Sorting the students using subject A
void SubASort(Student student[]) {

	printf("Subject A in ascending order:\n");  // Output of subject A ascending order
	printf("STUDENT NAME\t SUBJECT A\t SUBJECT B\n");


	Student AsortAscend[7];		//store array to hold student records
	for (size_t i = 0; i < 7; i++)
		AsortAscend[i] = student[i];
	for (size_t i = 0; i < 6; i++) {
		for (size_t j = i + 1; j < 7; j++) {
```

```
                        if (AsortAscend[i].subA > AsortAscend[j].subA) {
                                Student store = AsortAscend[i];
                                AsortAscend[i] = AsortAscend[j];
                                AsortAscend[j] = store;
                        }
                }
        }
        for (size_t i = 0; i < 7; i++)
                printf("%s\t\t%.1f\t\t%.1f\n", AsortAscend[i].studentName,
AsortAscend[i].subA, AsortAscend[i].subB);
}

//Function calculating the average and standard deviation
void AvgAndDevCal(Student student[]) {

        float avgA = 0, avgB = 0;
        float devA = 0, devB = 0;



        for (size_t i = 0; i < 7; i++) {          // calculating sum of both subjects
                avgA += student[i].subA;
                avgB += student[i].subB;
        }

        avgA /= 7;                          //calculating averages
        avgB /= 7;

        printf("Average of Subject A: %.1f\n", avgA);   // printing averages of both subjects
        printf("Average of Subject B: %.1f\n", avgB);
```

```
	for (size_t i = 0; i < 7; ++i) {		// standard deviation of both subjects
		devA += pow(student[i].subA - avgA, 2);
		devB += pow(student[i].subB - avgB, 2);
	}
	devA = sqrt(devA / 7);
	devB = sqrt(devB / 7);

	printf("Standard deviation of Subject A: %.1f\n", devA);   // printing standard deviation of both subjects
	printf("Standard deviation of Subject B: %.1f\n", devB);
}

// Writing previous functions to a different output file
int FileWrite(Student student[], char* File_name) {

	FILE* OutFile;			// opening output file
	OutFile = fopen(File_name, "w");

	if (!OutFile)			// if output file could not be opened
		return -1;

	// Writing students that received a diploma to writeback file
	fprintf(OutFile, "All the students that got diploma:\n");
	fprintf(OutFile, "STUDENT NAME\t SUBJECT A\t SUBJECT B\n");

	// going through student records in order to find scores >= 50
	for (size_t i = 0; i < 7; i++) {
```

```
            if (student[i].subA >= 50 && student[i].subB >= 50)
                fprintf(OutFile, "%s\t\t%.1f\t\t%.1f\n", student[i].studentName, student[i].subA, student[i].subB);
        }
        fprintf(OutFile, "\nSubject A in ascending order:\n");     // writing to file in ascending order using subject A
        fprintf(OutFile, "STUDENT NAME\t SUBJECT A\t SUBJECT B\n");

        Student AsortAscend[7];

        for (size_t i = 0; i < 7; i++)                    // copying student records
            AsortAscend[i] = student[i];

        for (size_t i = 0; i < 6; i++) {                  // sorting student records
            for (size_t j = i + 1; j < 7; j++) {
                if (AsortAscend[i].subA > AsortAscend[j].subA) {
                    Student store = AsortAscend[i];
                    AsortAscend[i] = AsortAscend[j];
                    AsortAscend[j] = store;
                }
            }
        }

        for (size_t i = 0; i < 7; i++)
            fprintf(OutFile, "%s\t\t%.1f\t\t%.1f\n", AsortAscend[i].studentName, AsortAscend[i].subA, AsortAscend[i].subB);

        float avgA = 0, avgB = 0;
        float devA = 0, devB = 0;
```

```
	for (size_t i = 0; i < 7; i++) {		// allocating sum of both sduents to average variable
		avgA += student[i].subA;
		avgB += student[i].subB;
	}
	avgA /= 7;				// calculating average of both subjects
	avgB /= 7;
	fprintf(OutFile, "\nAverage of Subject A: %.1f\n", avgA);   // printing student averages
	fprintf(OutFile, "Average of Subject B: %.1f\n", avgB);

	for (size_t i = 0; i < 7; ++i) {			// standard deviation calculation
		devA += pow(student[i].subA - avgA, 2);
		devB += pow(student[i].subB - avgB, 2);
	}
	devA = sqrt(devA / 7);
	devB = sqrt(devB / 7);

	fprintf(OutFile, "Standard deviation of Subject A: %.1f\n", devA);    // standard devition
output
	fprintf(OutFile, "Standard deviation of Subject B: %.1f\n", devB);

	fclose(OutFile);

	return 0;
}


// Main function with main manu and callback the previous functions
int main() {
```

```
	// Reading input file name
	char File_name[50];
	printf("Please enter input file name: ");
	scanf("%s", &File_name);

	// sorting students
	Student student[7];
	// reading data from file
	if (readFile(student, File_name) == -1) {
		printf("File %s not found!", File_name);
		return EXIT_FAILURE;
	}

	char uChoice[30];
	do {
		printf("\nPlease enter a command (enter h for help): ");	// User choice in the selection menu
		scanf("%s", &uChoice);

		if (strcmp(uChoice, "h") == 0) {		// output when user selects 'help' choice
			printf("1 to obtain all the students that got diploma\n");
			printf("2 to arrange subject A in ascending order\n");
			printf("3 to calculate the average and standard deviation\n");
			printf("4 to save all the above results in an output file\n");
			printf("e to exit\n");
		}
	// using if statements for each option available in the menu
		else if (strcmp(uChoice, "1") == 0)
```

```
                    // callback the Diploma function to see which students received one
                    gotDiploma(student);


            else if (strcmp(uChoice, "2") == 0)
                    // callback the sorting in ascending order using subject A
                    SubASort(student);
            else if (strcmp(uChoice, "3") == 0)
                    // callback the average and standard dev. calculation function
                    AvgAndDevCal(student);
            else if (strcmp(uChoice, "4") == 0) {
                    printf("Please specify the output file name: ");    // getting name of
output file to write data
                    scanf("%s", &File_name);

                    if (FileWrite(student, File_name) == -1)              // writing data to file
                            printf("File %s not opened!", File_name);
            }
    } while (strcmp(uChoice, "e") != 0);

    return EXIT_SUCCESS;
}
```