|  |  |  |
| --- | --- | --- |
| Final Project Report | | |
| Name | : | Fadel Cahyo M |
| Student No. | : | 001201700040 |

1. **INTRODCUTION**

In this final project, I want to make an GPA Calculator program in C language by using all the material in Engineering Programming Subject, the material use in this project include, Condition, array, preprocessor, pointer, function, struct, and etc.

1. **PSEUDECODE**

Pseudocode for GPACalculator Program

**Begin**

**Define n, i, j, struct studentData std[]**

**Input the amount n of the student**

**While n != 1**

**If n <= 0 THEN**

**PRINT invalid input**

**Continue**

**Else**

**Break**

**ENDIF**

**ENDWHILE**

**Call The function inputData(n)**

**For inputData(n) 1 to n**

**INPUT Std.name, std.ID**

**INPUT std homeworks, std exercise**

**Call function Validation()**

**Return r**

**INPUT std.midexam, stdfinalexam**

**Call function Validation()**

**Return r**

**ENDFOR**

**Call The function showData(n)**

**For showData(n) 1 to n**

**Call Function avgHW()**

**Pass In: hw1, hw2, hw3**

**Do the calculation**

**Return Result**

**Endfunction**

**Call Function avgEx()**

**Pass In: ex1, ex2, ex3**

**Do the calculation**

**Return Result**

**Endfunction**

**Call function totalScore()**

**Pass In: exAvg, hwAvg, mid, final**

**Do the calculation**

**Return total**

**Endfunction**

**` Call function grades()**

**Pass In: totalScore(), char \*grades**

**IF totalScore >= 85 THEN \*grades = ‘A’**

**ELIF totalScore < 85 && totalScore >= 70 THEN \*grades = ‘B’**

**ELIF totalScore < 70 && totalScore >= 60 THEN \*grades = ‘C’**

**ELIF totalScore < 60 && totalScore >= 55 THEN \*grades = ‘D’**

**ELSE totalScore < 55 THEN \*grades = ‘E’**

**ENDIF**

**PRINT All data inside struct**

**WRITE all data in struct to ResultGPA.txt**

**ENDFOR**

**End**

1. **FULL CODE**

Program 2.0 Final Term Code

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "calculation.h"

#include "grades.h"

#include "ctype.h"

struct studentData

{

char name[20];

int ID;

float homework[3], exercise[3];

float mid, final;

};

struct studentData std[100];

int i, j, n;

char grade = 'F';

FILE \*fptr;

float value, r;

char input[64], checkss[64];

float validation(){

r = 0;

while(r == 0){

printf("\nInput : ");

gets(checkss);

if (sscanf(checkss, "%f", &r) != 1){

r = 0;

printf("Invalid Input, Try Again!\n");

continue;

}

if (r < 0 || r > 100){

r = 0;

printf("Invalid Range, Try Again!\n");

continue;

}

else{

return r;

break;

}

}

}

void inputData(int n){

int i;

for(i = 0; i < n; i++){

printf("=== Student GPA Calculator ===\n");

printf("\n=== Input Data of Student %d ===\n", i+1);

printf("\nStudent Name : ");

scanf("%[^\n]", &std[i].name);

printf("Student ID : ");

scanf("%d", &std[i].ID);

fflush(stdin);

printf("Homework Grades (3 Grades)\n");

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

r = 0;

printf("-- Grade %d --", j+1);

std[i].homework[j] = validation();

}

printf("Exercise Grades (3 Grades)\n");

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

r = 0;

printf("-- Grade %d--", j+1);

std[i].exercise[j] = validation();

}

printf("-- Mid Exam Grade --");

std[i].mid = validation();

printf("-- Final Exam Grade --");

std[i].final = validation();

system("CLS");

}

}

void showData(int n){

int i;

fprintf(fptr,"No\tStudent Name\tStudent ID\t HW1\t HW2\t HW3\tAvgHW\t EX1\t EX2\t EX3\tAvgEx\t Mid\tFinal\tTotal\tGrade\n");

printf("No\tStudent Name\tStudent ID\t HW1\t HW2\t HW3\tAvgHW\t EX1\t EX2\t EX3\tAvgEx\t Mid\tFinal\tTotal\tGrade\n");

for(i = 0; i < n; i++){

float avgHomework = avgHw(std[i].homework[0], std[i].homework[1], std[i].homework[2]);

float avgExercise = avgEx(std[i].exercise[0], std[i].exercise[1], std[i].exercise[2]);

float totScore = totalScore(avgHomework, avgExercise, std[i].mid, std[i].final);

grades(&totScore, &grade);

fprintf(fptr,"%d\t%s\t%d\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t %s\n", i+1,

std[i].name, std[i].ID,

std[i].homework[0], std[i].homework[1], std[i].homework[2], avgHomework,

std[i].exercise[0], std[i].exercise[1], std[i].exercise[2], avgExercise,

std[i].mid, std[i].final, totScore, &grade);

printf("%d\t%s\t%d\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t %s\n", i+1,

std[i].name, std[i].ID,

std[i].homework[0], std[i].homework[1], std[i].homework[2], avgHomework,

std[i].exercise[0], std[i].exercise[1], std[i].exercise[2], avgExercise,

std[i].mid, std[i].final, totScore, &grade);

}

}

int main(){

printf("=== Welcomoe to Student GPA Calculator ===\n");

while(n == 0){

printf("\nInput the number of student : ");

fgets(input, 63, stdin);

if (sscanf(input, "%d", &n) != 1){

n = 0;

printf("Invalid Input\n");

continue;

}

if (n <= 0){

n = 0;

printf("Invald num of Student\n");

continue;

}

}

system("CLS");

inputData(n);

fptr = fopen("ResultGPA.txt","w");

showData(n);

fclose(fptr);

printf("");

}

char name[20];

int ID;

float homework[3], exercise[3];

float mid, final;

};

struct studentData std[100];

int i, j, n;

char grade = 'F';

FILE \*fptr;

float value, r;

char input[64], checkss[64];

float validation(){

r = 0;

while(r == 0){

printf("\nInput : ");

gets(checkss);

if (sscanf(checkss, "%f", &r) != 1){

r = 0;

printf("Invalid Input, Try Again!\n");

continue;

}

if (r < 0 || r > 100){

r = 0;

printf("Invalid Range, Try Again!\n");

continue;

}

else{

return r;

break;

}

}

}

void inputData(int n){

int i;

for(i = 0; i < n; i++){

printf("=== Student GPA Calculator ===\n");

printf("\n=== Input Data of Student %d ===\n", i+1);

printf("\nStudent Name : ");

scanf("%[^\n]", &std[i].name);

printf("Student ID : ");

scanf("%d", &std[i].ID);

fflush(stdin);

printf("Homework Grades (3 Grades)\n");

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

r = 0;

printf("-- Grade %d --", j+1);

std[i].homework[j] = validation();

}

printf("Exercise Grades (3 Grades)\n");

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

r = 0;

printf("-- Grade %d--", j+1);

std[i].exercise[j] = validation();

}

printf("-- Mid Exam Grade --");

std[i].mid = validation();

printf("-- Final Exam Grade --");

std[i].final = validation();

system("CLS");

}

}

void showData(int n){

int i;

fprintf(fptr,"No\tStudent Name\tStudent ID\t HW1\t HW2\t HW3\tAvgHW\t EX1\t EX2\t EX3\tAvgEx\t Mid\tFinal\tTotal\tGrade\n");

printf("No\tStudent Name\tStudent ID\t HW1\t HW2\t HW3\tAvgHW\t EX1\t EX2\t EX3\tAvgEx\t Mid\tFinal\tTotal\tGrade\n");

for(i = 0; i < n; i++){

float avgHomework = avgHw(std[i].homework[0], std[i].homework[1], std[i].homework[2]);

float avgExercise = avgEx(std[i].exercise[0], std[i].exercise[1], std[i].exercise[2]);

float totScore = totalScore(avgHomework, avgExercise, std[i].mid, std[i].final);

grades(&totScore, &grade);

fprintf(fptr,"%d\t%s\t%d\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t %s\n", i+1,

std[i].name, std[i].ID,

std[i].homework[0], std[i].homework[1], std[i].homework[2], avgHomework,

std[i].exercise[0], std[i].exercise[1], std[i].exercise[2], avgExercise,

std[i].mid, std[i].final, totScore, &grade);

printf("%d\t%s\t%d\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t %s\n", i+1,

std[i].name, std[i].ID,

std[i].homework[0], std[i].homework[1], std[i].homework[2], avgHomework,

std[i].exercise[0], std[i].exercise[1], std[i].exercise[2], avgExercise,

std[i].mid, std[i].final, totScore, &grade);

}

}

int main(){

printf("=== Welcomoe to Student GPA Calculator ===\n");

while(n == 0){

printf("\nInput the number of student : ");

fgets(input, 63, stdin);

if (sscanf(input, "%d", &n) != 1){

n = 0;

printf("Invalid Input\n");

continue;

}

if (n <= 0){

n = 0;

printf("Invald num of Student\n");

continue;

}

}

system("CLS");

inputData(n);

fptr = fopen("ResultGPA.txt","w");

showData(n);

fclose(fptr);

printf("");

}

for(i = 0; i < n; i++){

printf("=== Student GPA Calculator ===\n");

printf("\n=== Input Data of Student %d ===\n", i+1);

printf("\nStudent Name : ");

scanf("%[^\n]", &std[i].name);

printf("Student ID : ");

scanf("%d", &std[i].ID);

fflush(stdin);

printf("Homework Grades (3 Grades)\n");

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

r = 0;

printf("-- Grade %d --", j+1);

std[i].homework[j] = validation();

}

printf("Exercise Grades (3 Grades)\n");

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

r = 0;

printf("-- Grade %d--", j+1);

std[i].exercise[j] = validation();

}

printf("-- Mid Exam Grade --");

std[i].mid = validation();

printf("-- Final Exam Grade --");

std[i].final = validation();

system("CLS");

}

}

void showData(int n){

int i;

fprintf(fptr,"No\tStudent Name\tStudent ID\t HW1\t HW2\t HW3\tAvgHW\t EX1\t EX2\t EX3\tAvgEx\t Mid\tFinal\tTotal\tGrade\n");

printf("No\tStudent Name\tStudent ID\t HW1\t HW2\t HW3\tAvgHW\t EX1\t EX2\t EX3\tAvgEx\t Mid\tFinal\tTotal\tGrade\n");

for(i = 0; i < n; i++){

float avgHomework = avgHw(std[i].homework[0], std[i].homework[1], std[i].homework[2]);

float avgExercise = avgEx(std[i].exercise[0], std[i].exercise[1], std[i].exercise[2]);

float totScore = totalScore(avgHomework, avgExercise, std[i].mid, std[i].final);

grades(&totScore, &grade);

fprintf(fptr,"%d\t%s\t%d\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t %s\n", i+1,

std[i].name, std[i].ID,

std[i].homework[0], std[i].homework[1], std[i].homework[2], avgHomework,

std[i].exercise[0], std[i].exercise[1], std[i].exercise[2], avgExercise,

std[i].mid, std[i].final, totScore, &grade);

printf("%d\t%s\t%d\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t %s\n", i+1,

std[i].name, std[i].ID,

std[i].homework[0], std[i].homework[1], std[i].homework[2], avgHomework,

std[i].exercise[0], std[i].exercise[1], std[i].exercise[2], avgExercise,

std[i].mid, std[i].final, totScore, &grade);

}

}

int main(){

printf("=== Welcomoe to Student GPA Calculator ===\n");

while(n == 0){

printf("\nInput the number of student : ");

fgets(input, 63, stdin);

if (sscanf(input, "%d", &n) != 1){

n = 0;

printf("Invalid Input\n");

continue;

}

if (n <= 0){

n = 0;

printf("Invald num of Student\n");

continue;

}

}

system("CLS");

inputData(n);

fptr = fopen("ResultGPA.txt","w");

showData(n);

fclose(fptr);

printf("");

}

float avgExercise = avgEx(std[i].exercise[0], std[i].exercise[1], std[i].exercise[2]);

float totScore = totalScore(avgHomework, avgExercise, std[i].mid, std[i].final);

grades(&totScore, &grade);

fprintf(fptr,"%d\t%s\t%d\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t %s\n", i+1,

std[i].name, std[i].ID,

std[i].homework[0], std[i].homework[1], std[i].homework[2], avgHomework,

std[i].exercise[0], std[i].exercise[1], std[i].exercise[2], avgExercise,

std[i].mid, std[i].final, totScore, &grade);

printf("%d\t%s\t%d\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t %s\n", i+1,

std[i].name, std[i].ID,

std[i].homework[0], std[i].homework[1], std[i].homework[2], avgHomework,

std[i].exercise[0], std[i].exercise[1], std[i].exercise[2], avgExercise,

std[i].mid, std[i].final, totScore, &grade);

}

}

int main(){

printf("=== Welcomoe to Student GPA Calculator ===\n");

while(n == 0){

printf("\nInput the number of student : ");

fgets(input, 63, stdin);

if (sscanf(input, "%d", &n) != 1){

n = 0;

printf("Invalid Input\n");

continue;

}

if (n <= 0){

n = 0;

printf("Invald num of Student\n");

continue;

}

}

system("CLS");

inputData(n);

fptr = fopen("ResultGPA.txt","w");

showData(n);

fclose(fptr);

printf("");

}

}

}

system("CLS");

inputData(n);

fptr = fopen("ResultGPA.txt","w");

showData(n);

fclose(fptr);

}

Code for Calculation.h

float avgHw(float hw1, float hw2, float hw3){

float result = (hw1 + hw2 + hw3) / 3;

return result;

}

float avgEx(float ex1, float ex2, float ex3){

float result = (ex1 + ex2 + ex3) / 3;

return result;

}

float totalScore(float hw, float ex, float mid, float final){ //Function to calculate total score

float result = (0.15 \* hw) + (0.15 \* ex) + (0.3 \* mid) + (0.4 \* final);

return result;

}

Code for ctype.h

#define isUpper(c) ( ((c) >= 'A' && (c) <= 'Z' ) ? 1 : 0 )

#define isLower(c) ( ((c) >= 'a' && (c) <= 'z' ) ? 1 : 0 )

#define isAlpha(c) ( (isUpper((c)) || isLower((c)) ? 1 : 0) )

#define isDigit(c) ( ((c) >= '0' && (c) <= '9' ) ? 1 : 0)

#define isSpecial(c) ( (isAlpha((c)) || isDigit((c)) ) ? 0 : 1 )

Code for grades.h

char grades(float \*totScore, char \*grades)

{

if (\*totScore >= 85.0)

\*grades = 'A';

else if (\*totScore >= 70.0 && \*totScore < 85.0)

\*grades = 'B';

else if (\*totScore >= 60.0 && \*totScore < 70.0)

\*grades = 'C';

else if (\*totScore >= 55.0 && \*totScore < 60.0)

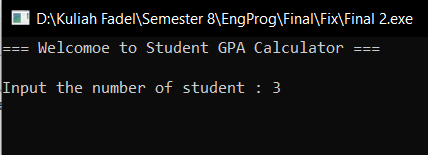
\*grades = 'D';

else if (\*totScore < 55.0)

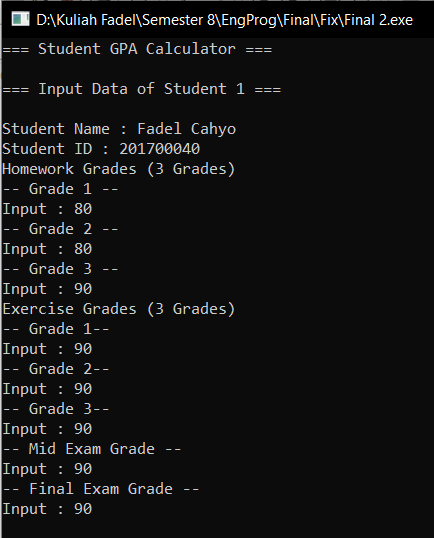
\*grades = 'E';

}

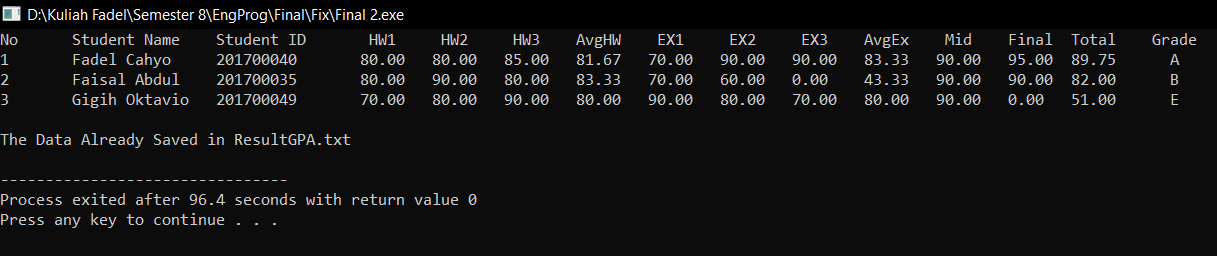
**Program Results**



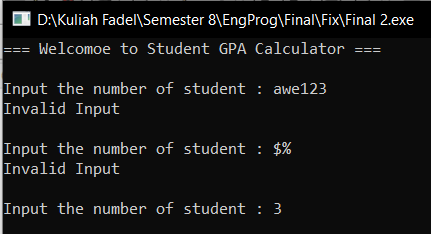
This is the first page when the program runs. We must input the amount of student. In this example is 3 student to calculate the gpa



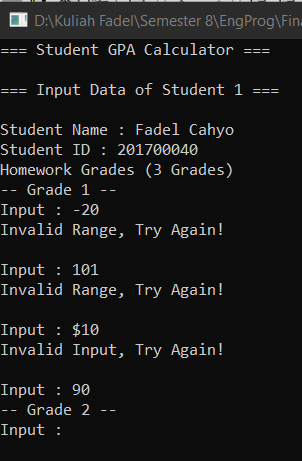
After we input the amount of student. We need to input all the data of that student. This is the function of input data, and this function will be loop until n.



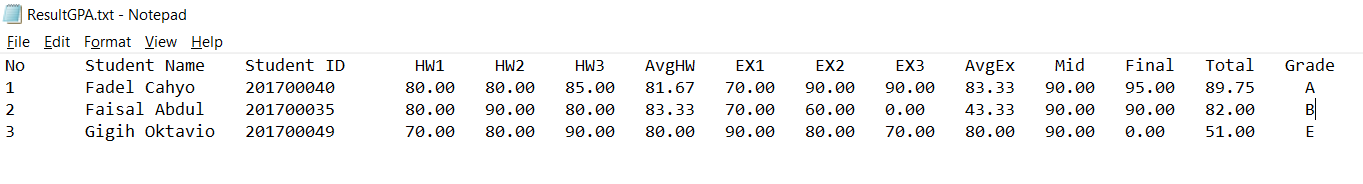
After all data already input, then we print the data, and saved it in ResultGPA.txt



This is for validation in the amount of the student, so we must input the integer value cannot be 0 or any other character beside number.



This is the validation on marks, so for all marks input is using the validation, the value cannot be negative, the value cannot more than 100, and the value cannot be alphabet or special character. After the validation passed then the next value can be inserted.



This is the ResultGPA.txt file.

1. **DESCRIPTION OF THE CODE**

**First Declaration of the code**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "calculation.h"

#include "grades.h"

#include "ctype.h"

So in my program, I use several preprocessor to make the code easier to understand, and also to manage the code to be more complex and easy to read.

**Struct Code**

struct studentData

{

char name[20];

int ID;

float homework[3], exercise[3];

float mid, final;

};

struct studentData std[100];

This structure is use to show the database of the student data, so in here I make the struct studentData to store the student data from input such as student name, student ID, student homeworks and exercise marks, student mid & final marks, and declare it as an array, the array name std[].

**Global Variables**

int i, j, n;

char grade = 'F';

FILE \*fptr;

float r;

char input[64], checkss[64];

This is the global variable for the program, int I, j is for the looping. Int n is for the total amount of the student, char grade is to declare the grade of each student, FILE \*fptr is for declaring file operations, float r is variable for validation in marks, char input and checks is for validation, so my validation is using the character first then convert it to the int or float.

**Int Main Code**

int main(){

printf("=== Welcomoe to Student GPA Calculator ===\n");

while(n == 0){ //Validation of input

printf("\nInput the number of student : ");

fgets(input, 63, stdin);

if (sscanf(input, "%d", &n) != 1){

n = 0;

printf("Invalid Input\n");

continue;

}

if (n <= 0){

n = 0;

printf("Invald num of Student\n");

continue;

}

}

system("CLS"); //clear screen

inputData(n); //function to input data

fptr = fopen("ResultGPA.txt","w");

showData(n);

fclose(fptr);

printf("\nThe Data Already Saved in ResultGPA.txt\n");

}

This is the main function of the program, First of all we need to input the amount of the student data, then there’s a validation if the input data isn’t an number it will be looping until it gets the number. Then it call system(“cls”) is for clearing the screen of the program. inputData(n) is a function to input the student data. Fptr is to write the data to ResultGPA.txt. showData(n) function is to show the data that already inputted.

**Integer Validation for student amount Code**

while(n == 0){ //Validation of input

printf("\nInput the number of student : ");

fgets(input, 63, stdin);

if (sscanf(input, "%d", &n) != 1){

n = 0;

printf("Invalid Input\n");

continue;

}

if (n <= 0){

n = 0;

printf("Invald num of Student\n");

continue;

}

}

This is to check whether the input is integer or not, so n is the amount of student, the loop will always run while n == 0, so when the value of input is 0 it will, it will looping until he got the number value, also if user input alphabet or special character, also if user input is negative number, because this validation is for the amount of the student.

**inputData(n) Function Code**

void inputData(int n){

int i;

for(i = 0; i < n; i++){

printf("=== Student GPA Calculator ===\n");

printf("\n=== Input Data of Student %d ===\n", i+1);

printf("\nStudent Name : ");

scanf("%[^\n]", &std[i].name);

printf("Student ID : ");

scanf("%d", &std[i].ID);

fflush(stdin);

printf("Homework Grades (3 Grades)\n");

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

r = 0;

printf("-- Grade %d --", j+1);

std[i].homework[j] = validation();

}

printf("Exercise Grades (3 Grades)\n");

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

r = 0;

printf("-- Grade %d--", j+1);

std[i].exercise[j] = validation();

}

printf("-- Mid Exam Grade --");

std[i].mid = validation();

printf("-- Final Exam Grade --");

std[i].final = validation();

system("CLS");

}

}

So this code is for input the student data, the parameter n is got from the amount of the student, so that the inputData function will be looping until the amount of the student data. So this input function is to store the student name, ID, homeworks marks, exercise marks, mid, and final marks into the struct. Also every input of the marks there is a function of validation. This validation is to check whether the input from user is a number or not.

**showData(n) function**

void showData(int n){

int i;

fprintf(fptr,"No\tStudent Name\tStudent ID\t HW1\t HW2\t HW3\tAvgHW\t EX1\t EX2\t EX3\tAvgEx\t Mid\tFinal\tTotal\tGrade\n");

printf("No\tStudent Name\tStudent ID\t HW1\t HW2\t HW3\tAvgHW\t EX1\t EX2\t EX3\tAvgEx\t Mid\tFinal\tTotal\tGrade\n");

for(i = 0; i < n; i++){

float avgHomework = avgHw(std[i].homework[0], std[i].homework[1], std[i].homework[2]);

float avgExercise = avgEx(std[i].exercise[0], std[i].exercise[1], std[i].exercise[2]);

float totScore = totalScore(avgHomework, avgExercise, std[i].mid, std[i].final);

grades(&totScore, &grade);

fprintf(fptr,"%d\t%s\t%d\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t %s\n", i+1,

std[i].name, std[i].ID,

std[i].homework[0], std[i].homework[1], std[i].homework[2], avgHomework,

std[i].exercise[0], std[i].exercise[1], std[i].exercise[2], avgExercise,

std[i].mid, std[i].final, totScore, &grade);

printf("%d\t%s\t%d\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t%.2f\t %s\n", i+1,

std[i].name, std[i].ID,

std[i].homework[0], std[i].homework[1], std[i].homework[2], avgHomework,

std[i].exercise[0], std[i].exercise[1], std[i].exercise[2], avgExercise,

std[i].mid, std[i].final, totScore, &grade);

}

}

This function is to show the data that already store in array of struct and also to print the data to ResultGPA.txt, in this function there’s also a function to calculate the total, average of homework, exercise, and the grade of each students. This function is in calculation.h in header file

**Function for validation on marks Code**

float validation(){

r = 0;

while(r == 0){

printf("\nInput : ");

gets(checkss);

if (sscanf(checkss, "%f", &r) != 1){

r = 0;

printf("Invalid Input, Try Again!\n");

continue;

}

if (r < 0 || r > 100){

r = 0;

printf("Invalid Range, Try Again!\n");

continue;

}

else{

return r;

break;

}

}

}

So, this function is to validate the user input for the marks in the studentData input, so in here I use the while loop, while loop here to check just like a flag, so I declare the r = 0, then check the condition if the input is not a number, declare r = 0 and print invalid input, try again!. The second condition is if the marks value is < 0 or r > 100 is will return invalid range, because for marks the value is only from 0 to 100 and else is the condition where the user put the correct input, then breaks the loop.

**Calculation.h code**

float avgHw(float hw1, float hw2, float hw3){

float result = (hw1 + hw2 + hw3) / 3;

return result;

}

float avgEx(float ex1, float ex2, float ex3){

float result = (ex1 + ex2 + ex3) / 3;

return result;

}

float totalScore(float hw, float ex, float mid, float final){ //Function to calculate total score

float result = (0.15 \* hw) + (0.15 \* ex) + (0.3 \* mid) + (0.4 \* final);

return result;

}

So, in calculation.h there are 3 function in here, there is to calculate the average of homework, calculate the average of exercise, and calculate the total score of the student. Actually avgHW() and avgEx() function is the same function, it just to calculate the average, but I separate it so that it would not get confuse, the parameter is from the student data, it got from user input the homework and the exercise marks, then all the parameter divided by 3 and return the result. For the totalScore() function is to get the total score value of the students, the parameter is from the student marks from avgEx, avgHw, mid, and final exam value then return as result also.

**Grades.h Code**

char grades(float \*totScore, char \*grades)

{

if (\*totScore >= 85.0)

\*grades = 'A';

else if (\*totScore >= 70.0 && \*totScore < 85.0)

\*grades = 'B';

else if (\*totScore >= 60.0 && \*totScore < 70.0)

\*grades = 'C';

else if (\*totScore >= 55.0 && \*totScore < 60.0)

\*grades = 'D';

else if (\*totScore < 55.0)

\*grades = 'E';

}

So, in grades.h is to determine the the grades of the student, by using pointer the value can easily change for each student. So the condition is to determine the grades of each student whether the student gets A, B, C, D or, E.

1. **References**
2. Jack Ma,” Code in C,” Fourth Edition, What Publishing, 2015
3. Nelson Dee,” C Programming for Students,” First Edition, No Publishing, 2001