ASSIGNMENT 2 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	PROG102: Procedural Programming		
Submission date		**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Bùi Hương Linh	**Student ID**	GBH200662
Class	GCH1002	**Assessor name**	Đỗ Tiến Thành
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	

Grading grid

P4	P5	M3	M4	D2

☼ **Summative Feedback:** ☼ **Resubmission** **Feedback:**		
Grade:	**Assessor Signature:**	**Date:**
Lecturer Signature:		

Contents

I/ Introduction:

I.1/ Problem :

A math teacher wants to manage grades of a class. The math teacher wants to make a student transcript and submit it to the school. Based on the transcript, the university can classify students according to merit, good, average,ect.

I.2/ Solution:

The school has expressed an opinion that it wants an application that can help teachers solve this problem easily and effectively. This application will contribute to solving the problems that users face.

I.3/ Procedural programming:

I.3.1/ What is the procedural programming?

Procedural programming (POP) is where the major focus on performing tasks in a sequential order. It divides a large program into small functional blocks or functions for ease of programming and testing easier.(Learn Computer Science,2021)

I.3.2/ Characteristies:

- Focus on the work to be done (algorithms).

+ Helps beginners can improve their mindset about solving problems.

- Large program is divided into subroutines, each of which can be called one or more times in any order.

+It makes it easier for programmers to address problems since faults in each sub-program may be readily fixed.

- Most functions use common data.

- Data in the system is moved from one function to another.

+ Programmers can manage data easily.

- Uses immutable data.(Leonila Cordrey,2021)

II/ Implementation:

+ #include<stdio.h> : Get various functions to perform input and output.

+ #include<stdlib.h> :

+ #include<conio.h> : Perform input and output operations from the screen.

+ Struct: Used to declare student objects with properties or program manager.

- int: Used to store student object with attributes such as id, name,gender,age,ect.
- float: Used to declare subjects, average scores, highest, lowest of student. Because when entering points, there may be commas, we have to use float.
- char name,gender: Use for entering name, gender of student, [50] is limit of characters.

```
struct Student{
    int IDs;
    char name[50];
    char gender[50];
    int age;
    int rate;
    float math,chemistry,physics;
    float avg;
    float highest, lowest;
    char nationality[10];
};
```

+ ST: Stands for student.

- &st: Declared to pass a parameter.
- a: array declares struct st.

```
typedef Student ST;
void InputStudent(ST &st);
void OutputStudent(ST st);
void InputStudentList(ST a[], int &n);
void OutputStudentList(ST a[], int n);
void SortStudent(ST a[], int n);
float FindHighestScore(ST a[], int n);
float FindLowestScore(ST a[], int n);
```

+ int main : Where the show starts.

- int n: Enter student integer.
- Do-while: Used for the user to enter the correct condition of ‘n’.
- ST a[n]: declare variable ‘ST’, variable ‘n’ array ‘a’.
- Create a menu:
  1. Enter student list
  2. Export student list
  3. Sort the student's score
  4. Highest score
  5. Lowest score
  0. Exit program

```
int main(){
    int n;

    do{
        printf("\n Enter the number of student: \n");
        scanf("%d", &n);
    }while (n<=0);

    ST a[n];


    while (true){

        printf("\n    ---STUDENT MANAGEMENT PROGRAM---   \n");
        printf("-----------------------------------------\n");
        printf("| 1.Input Student List                  |\n");
        printf("| 2.Output Sudent List                  |\n");
        printf("| 3.Sort of student by average scores   |\n");
        printf("| 4.Find highest score                  |\n");
        printf("| 5.Find lowest score                   |\n");
        printf("| 0.Exit program                        |\n");
        printf("-----------------------------------------\n");
        printf("\n      ---ENTER YOUR SELECTION---       \n");
```

+Switch case: Using switch case statement to create the menu.

- op: create a op variable.
- Input Student List (a,n)
- Output Student List(a,n)

→ call back the declared function.

- getch(): to stop the screen.
- break: end

```
int op;
scanf("%d", &op);
switch(op){
    case 1:
        printf("\n You have selected: Input Student List \n");
        InputStudentList(a, n);
        printf("\n Press any key to continue \n");
        getch();
        break;

    case 2:
        printf("\n You have selected: Output Student List \n");
        OutputStudentList(a, n);
        printf("\n Press any key to continue \n");
        getch();
        break;

    case 3:
        printf("\n You have selected: Sort of student by average scores \n");
        SortStudent(a,n);
        printf("\n Press any key to continue \n");
        getch();
        break;

    case 4:
        printf("\n Highest score = %.2f", FindHighestScore (a,n));
        printf("\n Press any key to continue \n");
        getch();
        break;

    case 5:
        printf("\n Lowest score = %.2f", FindLowestScore (a,n));
        printf("\n Press any key to continue \n");
        getch();
        break;

    case 0:
        printf("\n You have selected: Exit program \n");
        exit(0);
        break;
    default:
        break;


    }

    }
    return 0;
}
```

+ Do-while: Used to perform the previous job and check the following condition.

First, enter 3 points of math, physics, chemistry. if you enter math, physics, chemistry scores between 0 and10, the program will continue to run. Conversely, if you enter math < 0 or physics < 0 or chemistry < 0 or math > 10 or physics > 10 or chemistry > 10, the program forces the user to re-enter the score.

```
do{
printf("\n Math scores = \n");
scanf("%f", &st.math);

printf("\n Chemistry scores = \n");
scanf("%f", &st.chemistry);

printf("\n Physics scores = \n");
scanf("%f", &st.physics);
}while(st.math < 0 || st.chemistry < 0 || st.physics < 0 || st.math > 10 || st.chemistry >10 || st.physics >10);
```

+If-else: Use used to make right and wrong decisions when implementing algorithms.

If the average score is less than 6.5, the student is rankes as average. else if the average score is < 8 and the average score is >= 6.5, the student is ranked as good. Conversely, the student is ranked as excellent.

```
    printf("\n Rate: %s", st.rate);
    if (st.avg < 6.5){
        printf("\n Average \n");
    }else if (st.avg < 8 && st.avg >= 6.5){
        printf("\n Good \n");
    }else{
        printf("\n Excellent \n");
    }
}
```

+ for loop: Use ‘int’ for ‘i’ to traverse data from first element to last element.

- Initials value highest=a[1].avg. Browse the elements of the array in turn. If an element has the highest value, we proceed to assign that value to the highest. For each element, compare it with 'highest'. If 'highest' is lower than the element of the array, assign it temporarily to that particular element which is the array.
- Initials value highest=a[1].avg. Browse the elements of the array in turn. If an element has the lowest value, we proceed to assign that value to the lowest. For each element, compare it with 'lowest'. If 'lowest' is raiser than the element of the array, assign it temporarily to that particular element which is the array.

```
float FindHighestScore(ST a[], int n){
    float highest = a[1].avg;
    for(int i = 0; i<n; i++){
        if(highest < a[i].avg){
            highest = a[i].avg;
        }
    }
    return highest;
}
float FindLowestScore(ST a[], int n){
    float lowest = a[1].avg;
    for(int i = 0; i<n; i++){
        if(lowest > a[i].avg){
            lowest = a[i].avg;
        }
    }
    return lowest;
}
```

+ void: Return null function.

- Use the for loop structure to enter students in order. Here, I use 'i+1' because 'student' is never 0 students. It always starts with the first student.
- Use the for loop structure to export students in order.

```
void InputStudentList(ST a[], int &n){
    printf("\n ---------- \n");
    for(int i=0; i<n ; i++){
        printf("\n Input student %d: ", i+1);
        InputStudent(a[i]);
        printf("\n ---------- \n");
    }
}

void OutputStudentList(ST a[], int n){
    printf("\n ---------- \n");
    for(int i=0; i<n ; i++){
        printf("\n Student information %d ", i+1);
        OutputStudent(a[i]);
        printf("\n ---------- \n");
        }
}
```

III/ Progrem results:

```
 Enter the number of student:
3

    ---STUDENT MANAGEMENT PROGRAM---
-----------------------------------------
| 1.Input Student List                  |
| 2.Output Sudent List                  |
| 3.Sort of student by average scores   |
| 4.Find highest score                  |
| 5.Find lowest score                   |
| 0.Exit program                        |
-----------------------------------------

      ---ENTER YOUR SELECTION---
1

 You have selected: Input Student List

 ----------

 Input student 1:
 IDs: 200662

 Name: Linh

 Gender: Female

 Age: 19

 Nationality: Viet nam

 Math scores =
8

 Chemistry scores =
8

 Physics scores =
9

 Avg = 8.33
 Rate: (null)
 Excellent
```

```
Input student 2:
IDs: 200562

Name: Anh

Gender: Famle

Age: 20

Nationality: Viet nam

Math scores =
7

Chemistry scores =
6

Physics scores =
5

Avg = 6.00
Rate: `!!
Average

----------

Input student 3:
IDs: 200719

Name: Hoa

Gender: Male

Age: 19

Nationality: Viet nam

Math scores =
7

Chemistry scores =
6

Physics scores =
7

Avg = 6.67
Rate: (null)
Good

----------
```

```
3

You have selected: Sort of student by average scores

----------

Student information 1
IDs: 200662
Name: Linh
Gender: Female
Age: 19
Nationality: Viet nam
Math scores = 8.00
Chemistry scores = 8.00
Physics scores = 9.00
Avg = 8.33
Rate student: (null)
Excellent

----------

Student information 2
IDs: 200562
Name: Anh
Gender: Famle
Age: 20
Nationality: Viet nam
Math scores = 7.00
Chemistry scores = 6.00
Physics scores = 5.00
Avg = 6.00
Rate student: `‼╡
Average

----------

Student information 3
IDs: 200719
Name: Hoa
Gender: Male
Age: 19
Nationality: Viet nam
Math scores = 7.00
Chemistry scores = 6.00
Physics scores = 7.00
Avg = 6.67
Rate student: (null)
Good
```

```
4

 Highest score = 8.33
 Press any key to continue

    ---STUDENT MANAGEMENT PROGRAM---
----------------------------------------
| 1.Input Student List                 |
| 2.Output Sudent List                 |
| 3.Sort of student by average scores  |
| 4.Find highest score                 |
| 5.Find lowest score                  |
| 0.Exit program                       |
----------------------------------------

      ---ENTER YOUR SELECTION---
5

 Lowest score = 6.00
```

```
0

 You have selected: Exit program

--------------------------------
Process exited after 33.88 seconds with return value 0
Press any key to continue . . .
```

IV/ Testing:

Test	Test Description	Test Steps	Test Data	Expected Results	Actural Results	Pass/Fall
1	Used to import student list and calculate the average of three subjects.	1. Enter 2. Input data of student	id: 200662 Name: Linh Gender:Female Age:19 Math:7 Chemistry:8 Physics:9 Avg=8.00 Rate:Excellent	Input Student List	Successfull	Pass
2	Export the list entered case 1	1.Enter 2 2.Output Student List	Id:200662 Name: Linh Gender:Female Age:19 Math:7 Chemistry:8 Physics:9 Avg=8.00 Rate:Excellent	Show id,name, age, etc. enter in case 1	Successfull	Pass
3	Sort of student by average score	1.Enter 3 2.Sort average score	Id:200662 Name: Linh Gender:Female Age:19 Math:7 Chemistry:8 Physics:9 Avg=8.00 Rate:Excellent	Show average score	Successfull	Pass
4	Find highest score	1.Enter 4 2.Highest score	Highest score: 8.00	Show highest score	Successful	Pass
5	Find lowest score	1.Enter 5 2.Lowest score	Lowest score:6.5	Show lowest score	Successfull	Pass
0	Exit	1.Enter 2.Exit(0)	0	Exit program	Successful	Pass

V/ Evaluation & Conclution:

I think my program is not very good, need to change some places to be more suitable.

Through creating this student management program I have learned a lot about coding. it helped me improve my background knowledge, understand more about how functions, variables, statements, and loops are used.

In terms of advantages, the functions and algorithms are quite simple and the program is not too complicated. The program runs very simple and fast.

In terms of challenges, several interlocking structures and processes make program development complicated and the app has some bugs that I can't fix.

In the future, I will write a more complete student management software with many functions based on the designed algorithms.

I hope that the program I created can help teachers and schools manage students more effectively.In the future, I hope, I can write a perfect student management program.

Bibliography

(2021, 10 28). Retrieved from learncomputerscienceonline.com: https://www.learncomputerscienceonline.com/procedural-programming/

Cordrey, L. (2021, 10 29). Retrieved from popularask.net: https://popularask.net: https://popularask.net/which-of-the-following-is-a-main-characteristic-of-procedural-programming-language/?fbclid=IwAR1EOdIZqEnRmfwDA15-V1zZHECGGma92dS13gFxfN8_yiPUP5unXvtaX8M