



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:**
Feedback:

⚙ **Resubmission**

Grade:

Assessor Signature:

Date:

Lecturer Signature:

Contents

I/ Introduction:	4
I.1/ Problem :	4
I.2/ Solution:	4
I.3/ Procedural programming:	4
I.3.1/ What is the procedural programming?	4
I.3.2/ Characteristics:	4
II/ Implementation:	5
III/ Program results:	10
IV/ Testing:	14
IV.1/ Test plan and perform tests:.....	14
IV.2/ Show test:	14
V/ Evaluation:	16
Bibliography	16

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/ Characteristics:

- 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:

-Explain program:

- + #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.
- + int main : Where the show starts.

```
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");
```

- + printf: Printf messages in the app you do and display it on the screen.

```
printf("\n    ---STUDENT MANAGEMENT PROGRAM---  \n");
printf("-----\n");
printf("1. Input Student List\n");
printf("2. Output Student 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");
```

+ Struct: Used to store student object with attributes such as id, name, gender, age, etc.

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

+ getch: Use to stop the screen.

```
printf("\n You have selected: Input Student List \n");
InputStudentList(a, n);
printf("\n Press any key to continue \n");
getch();
break;
```

+ Do-while:

- Used for the user to enter the correct condition of 'n'.
- Used to perform the previous job and check the following condition.

```
int main(){
    int n;

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

    ST a[n];

    while (true){
```

```
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.

```
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");
}
}
```

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

```
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];
};
```

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

```

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;
}

```

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

```

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 SortStudent(ST a[], int n){
    student st;
    int i,j;
    for (i=j+1; i<n; i++)
        for (j=0; j<n; j++)
            if (a[i].avg > a[j].avg){
                st=a[i];
                a[i]=a[j];
                a[j]=st;
            }
    OutputStudentList(a,n);
}

```

+ void: Return null function.

```

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/ Program results:

```
Enter the number of student:
3

---STUDENT MANAGEMENT PROGRAM---
-----
| 1.Input Student List          |
| 2.Output Sudent List        |
| 3.Sort of student by average |
| 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: Female

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
**ENTER YOUR SELECTION**

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 |
| 4.Find highest score         |
| 5.Find lowest score          |
| 0.Exit program               |
|                               |
---|
---ENTER YOUR SELECTION---
5
Lowest score = 6.00
```

IV/ Testing:

IV.1/ Test plan and perform tests:

-First, when I run the program, I will enter the student number as negative to see if it works. If it doesn't work, the program will re-enter .

-Next, the program will display a menu for the user to choose from 0-5.

-Select 1 to enter id, name, age,

-I will press any key to continue the program. the following options are similar.

-Select 2 to display the part entered in 1.

-Select 3 to display the average score.

-Select 4 to display the highest grade point average.

-Select 5 to display the lowest grade point average.

-Select 0 to exit the program.

IV.2/ Show test:

```
Enter the number of student:
-10

Enter the number of student:
3

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

```
3
You have selected: Sort of student by average scores
-----

Student information 1
IDs: 112
Name: Linh
Gender: Female
Age: 19
Nationality: Viet nam
Math scores = 8.00
Chemistry scores = 9.00
Physics scores = 8.00
Avg = 8.33
Rate student: (null)
Excellent
-----

Student information 2
IDs: 0
Name: Anh
Gender: Male
Age: 20
Nationality: Viet nam
Math scores = 7.00
Chemistry scores = 4.00
Physics scores = 5.00
Avg = 5.33
Rate student: `!!L
Average
-----

Student information 3
IDs: -66
Name: Hoa
Gender: Male
Age: 19
Nationality: Viet nam
Math scores = 7.00
Chemistry scores = 8.00
Physics scores = 6.00
```

```
ENTER YOUR SELECTION
4
Highest score = 8.33
Press any key to continue

---STUDENT MANAGEMENT PROGRAM---
| 1.Input Student List |
| 2.Output Student 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 = 5.33
Press any key to continue
```

V/ Evaluation:

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

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