ASSIGNMENT 1 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

P1	P2	P3	M1	M2	D1

☼ **Summative Feedback:** ☼ **Resubmission**

Feedback:

Grade:	**Assessor Signature:**	**Date:**
Lecturer Signature:		

Contents

Task 1: Programming Language, Procedural Programming.

I/ Programming Language:

I.1/ Introduction:

A programming language is a set of instructions to a computer to execute or a computer language programmers use to develop software programs.

Currently, there are many programming languages that have been developed to meet the needs of programmers. This has encouraged many people to study the field of information technology because of its growing attraction and development.

Image 1: Some programming language (source: Internet)

I.2/ Type of programming language:

I.2.1/ Low-level language:

A low-level language is a programming language that does not provide abstraction from the hardware and and is more closely related to a computer system.(JavaTpoint,2021)

I.2.2/ High-level language:

High-level languages are close to human languages, making it easier for people to read and write.(Anthony Grant,2020)

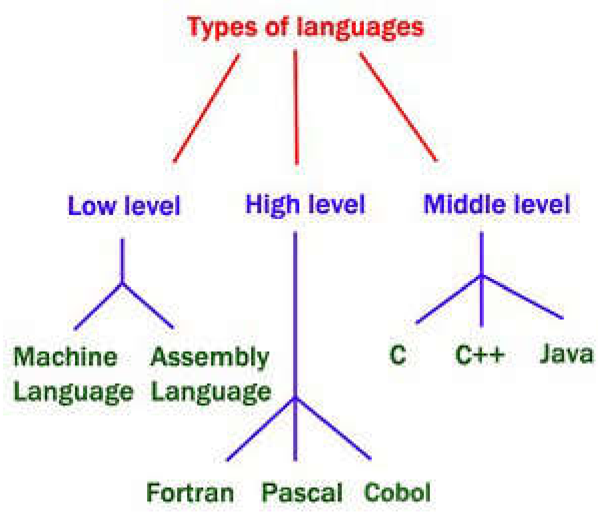


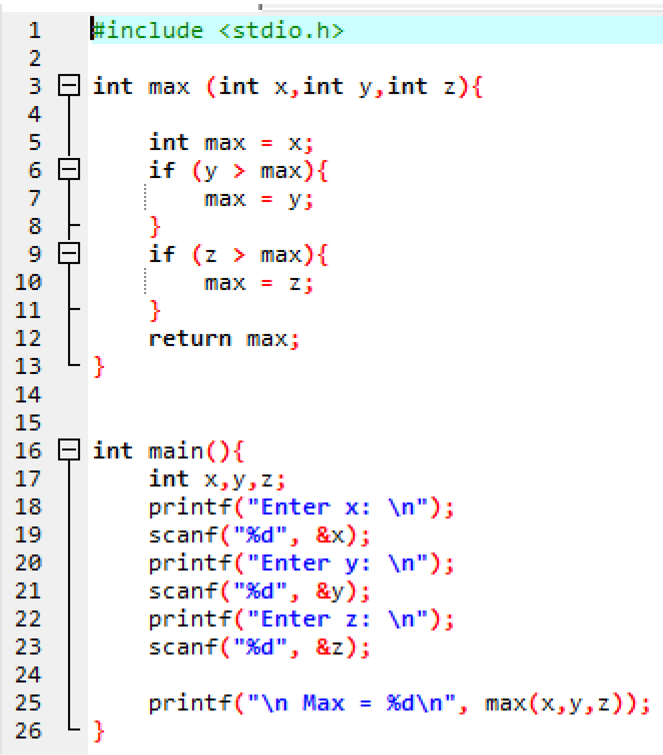
Image 2: Types of programming language(source: Internet)

II/ Procedural Programming:

II.1/ Introduction:

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)

Example: Using functions find max



```
#include <stdio.h>

int max (int x,int y,int z){

    int max = x;
    if (y > max){
        max = y;
    }
    if (z > max){
        max = z;
    }
    return max;
}


int main(){
    int x,y,z;
    printf("Enter x: \n");
    scanf("%d", &x);
    printf("Enter y: \n");
    scanf("%d", &y);
    printf("Enter z: \n");
    scanf("%d", &z);

    printf("\n Max = %d\n", max(x,y,z));
}
```

II.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)

III/ Problem statement:

III.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.

III.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.

Task 2: Analysis.

I/ List data types and data structures needed in the problem:

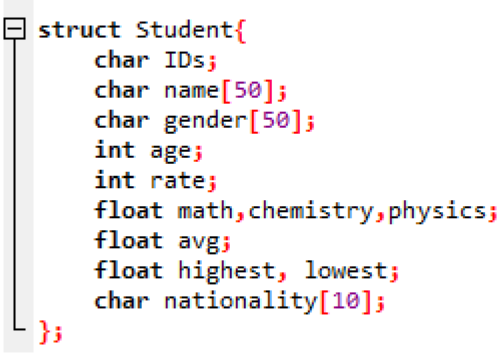
Variables	Type	Describe	Explain
Id	int	ID of student	Can include number and character. Eg: 200662,ect.
Name	char	Name of student	Used to declare name. Eg: Bui Huong Linh,ect.
Gender	char	Gender of student	Male and female.
Age	int	Age of student	Is always an integer. Eg: 18,19,ect.
n	int	Number of student	Is always an interger.
Math,chemistry,physics	float	Score of student	Used to declare subjects. Eg: math,english…

Rate	int	Rate of student	Characters Eg: Good, Excellent….
Avg	float	Avg of student	Used to declare average scores. Eg: 8.5,6,7,ect.
Highest, lowest	float	Highest, lowest of student	Find highest, lowest.
op	int	Option of user	Creat e a variable option.

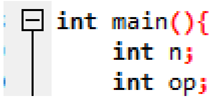
Code declaring a variable:

Int + Char: used to declare data types like IDs, name…

Float: used to declare subjects, average scores.



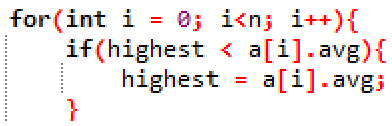
```
struct Student{
    char IDs;
    char name[50];
    char gender[50];
    int age;
    int rate;
    float math,chemistry,physics;
    float avg;
    float highest, lowest;
    char nationality[10];
};
```



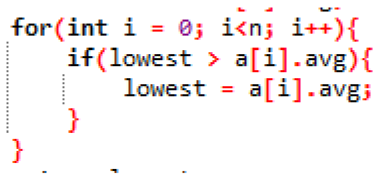
```
int main(){
    int n;
    int op;
```

Code: for loop

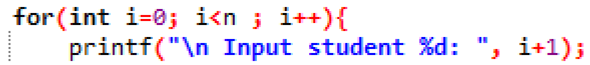
Use ‘int’ for ‘i’ to traverse data from first element to last element.



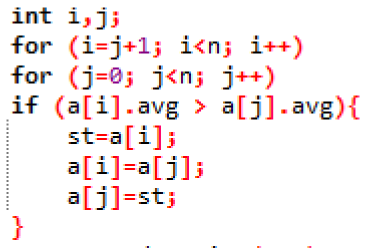
```
for(int i = 0; i<n; i++){
    if(highest < a[i].avg){
        highest = a[i].avg;
    }
```



```
for(int i = 0; i<n; i++){
    if(lowest > a[i].avg){
        lowest = a[i].avg;
    }
}
```



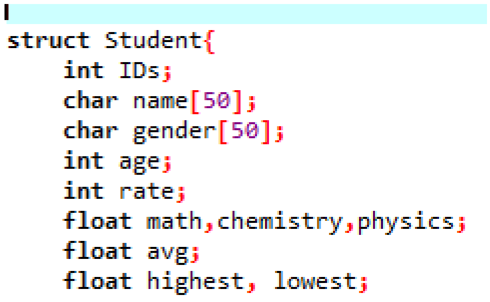
```
for(int i=0; i<n ; i++){
    printf("\n Input student %d: ", i+1);
```



```
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;
}
```

Code: Struct

Used to declare student objects with properties or program manager.

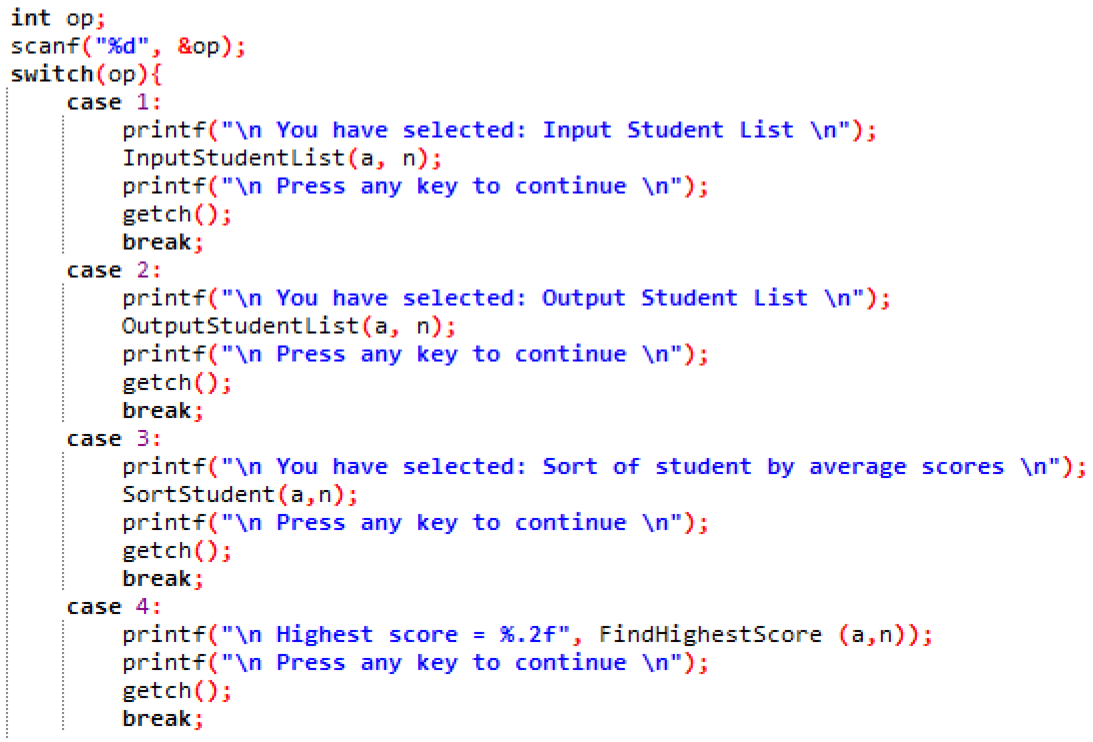


```
struct Student{
    int IDs;
    char name[50];
    char gender[50];
    int age;
    int rate;
    float math,chemistry,physics;
    float avg;
    float highest, lowest;
```

II/ Switch case statement:

Using switch case statement to create the menu.

Code:

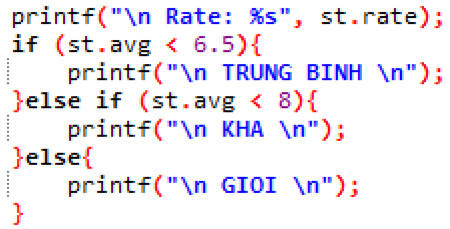


```
int op;
scanf("%d", &op);
switch(op){
    case 1:
        printf("\n You have selected: Input Student List \n");
        InputStudentList(a, n);
        printf("\n Press any key to continue \n");
        getch();
        break;
    case 2:
        printf("\n You have selected: Output Student List \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;
```

III/ If-else statement:

Use used to make right and wrong decisions when implementing algorithms.

Code:

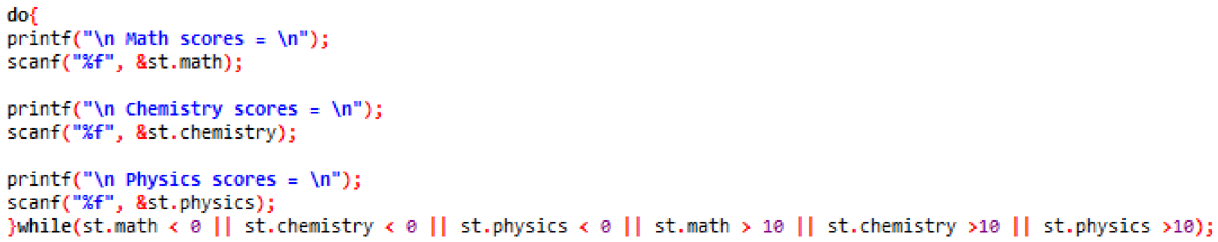


```
printf("\n Rate: %s", st.rate);
if (st.avg < 6.5){
    printf("\n TRUNG BINH \n");
}else if (st.avg < 8){
    printf("\n KHA \n");
}else{
    printf("\n GIOI \n");
}
```

IV/ Do-while statement:

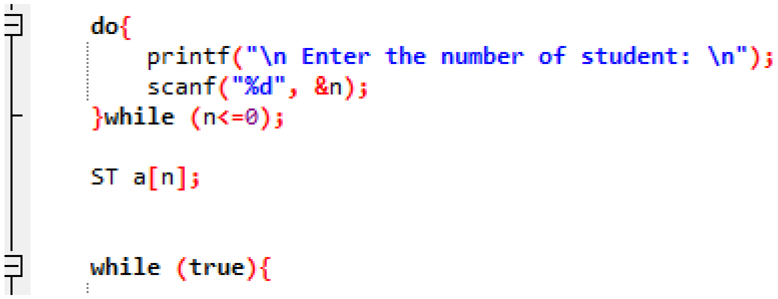
-Used to perform the previous job and check the following condition.

Code:



```
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);
```

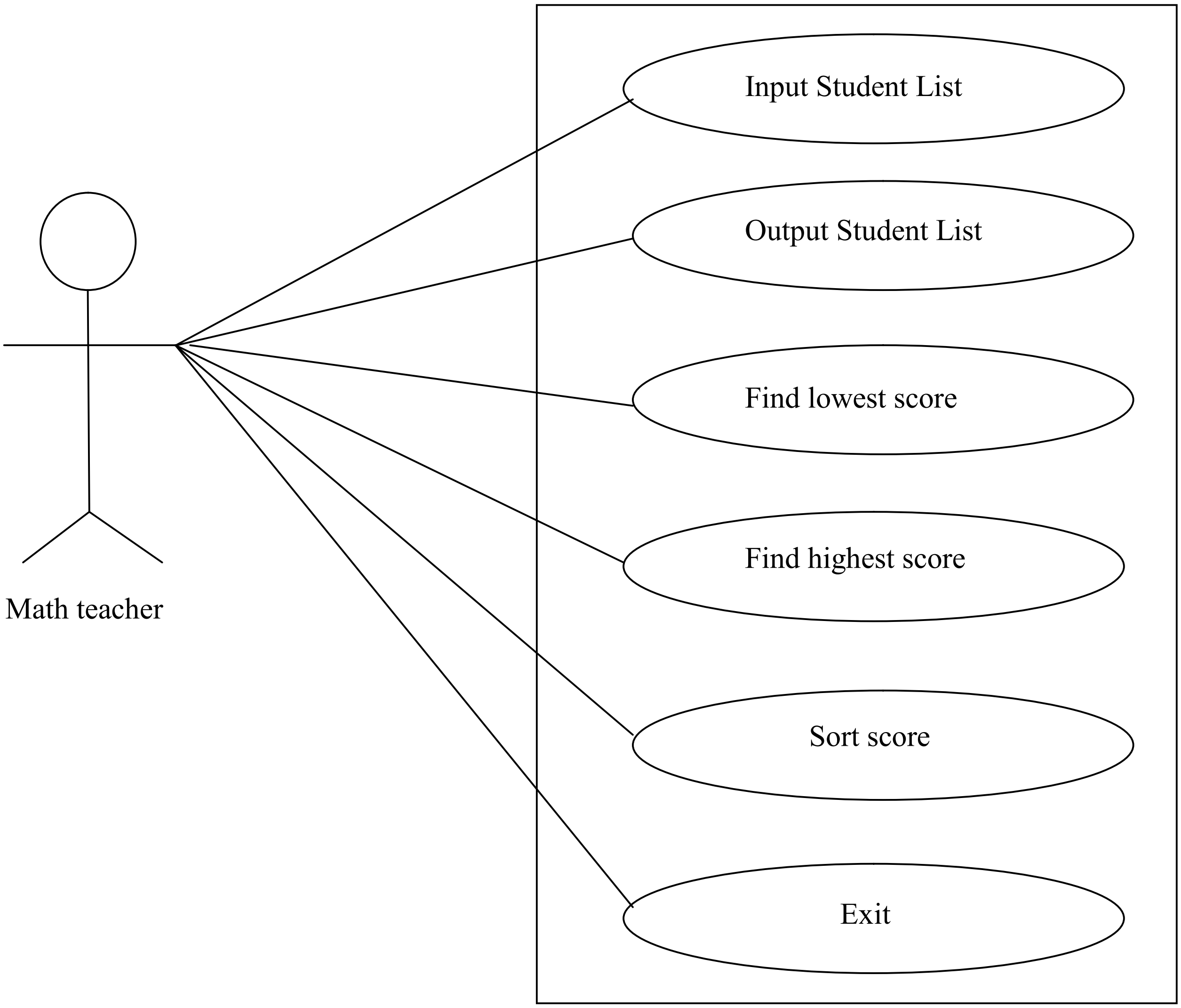
-Used for the user to enter the correct condition of ‘n’.



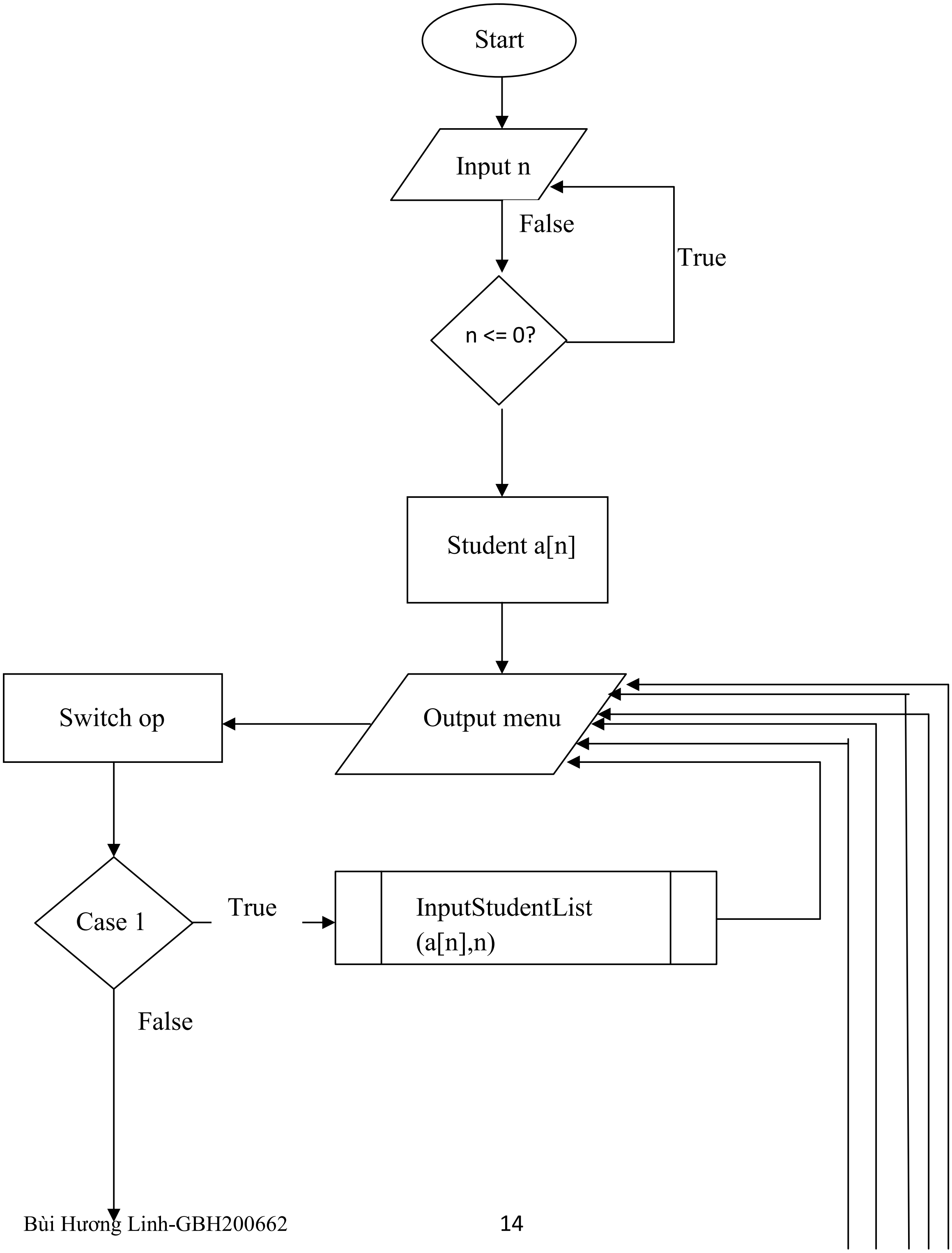
```
do{
    printf("\n Enter the number of student: \n");
    scanf("%d", &n);
}while (n<=0);

ST a[n];


while (true){
```

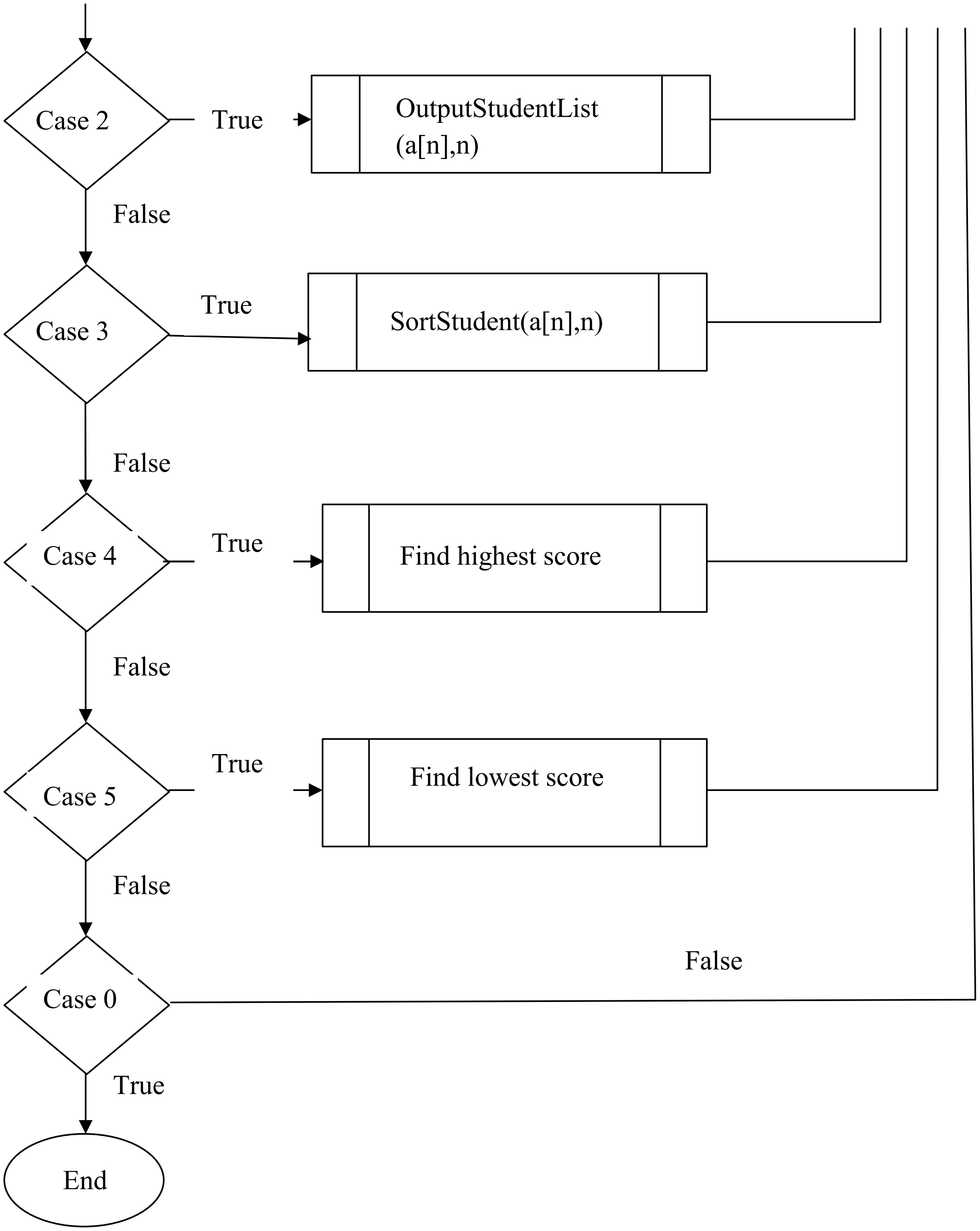
Task 3: Design

I/ Use case:



II/ Flowchart:




Case 2
True
OutputStudentList (a[n],n)
False
Case 3
True
SortStudent(a[n],n)
False
Case 4
True
Find highest score
False
Case 5
True
Find lowest score
False
Case 0
False
True
End

Explain:

At first, when users want to use the program, they have to enter the number of student.

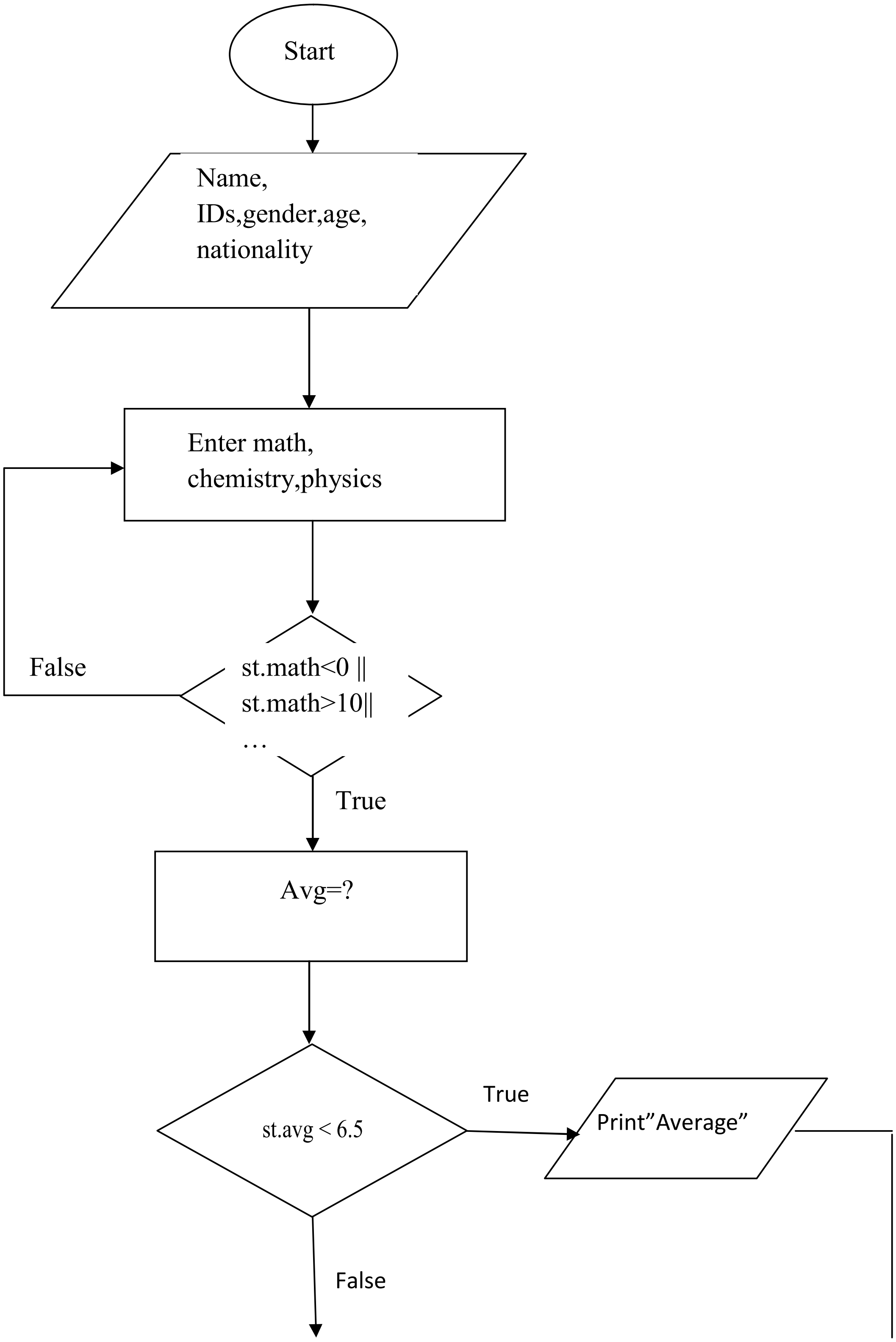
After entering, the user will have 6 options to choose from.

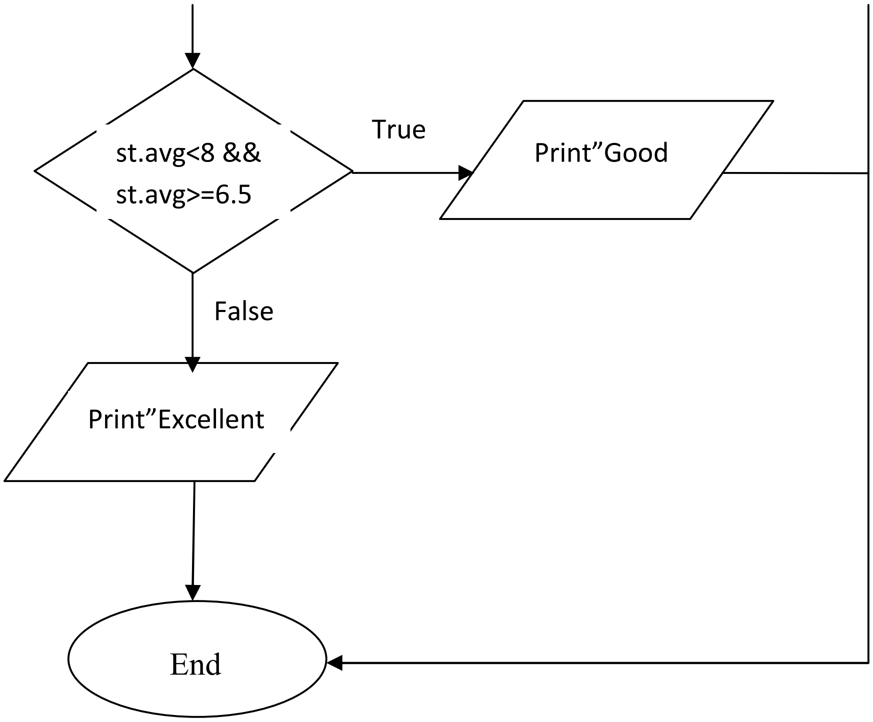
To choose to enter student information, press 1 and then 'enter'.

When finished, we will press any key to bring up the menu.

After each option the menu will repeat for the user to choose another option.

If the user wants to exit from the menu, they have to choose option 0 then the program will exit.


Start
Name, IDs,gender,age, nationality
Enter math, chemistry,physics
False
st.math<0 || st.math>10|| …
True
Avg=?
st.avg < 6.5
True
Print"Average"
False

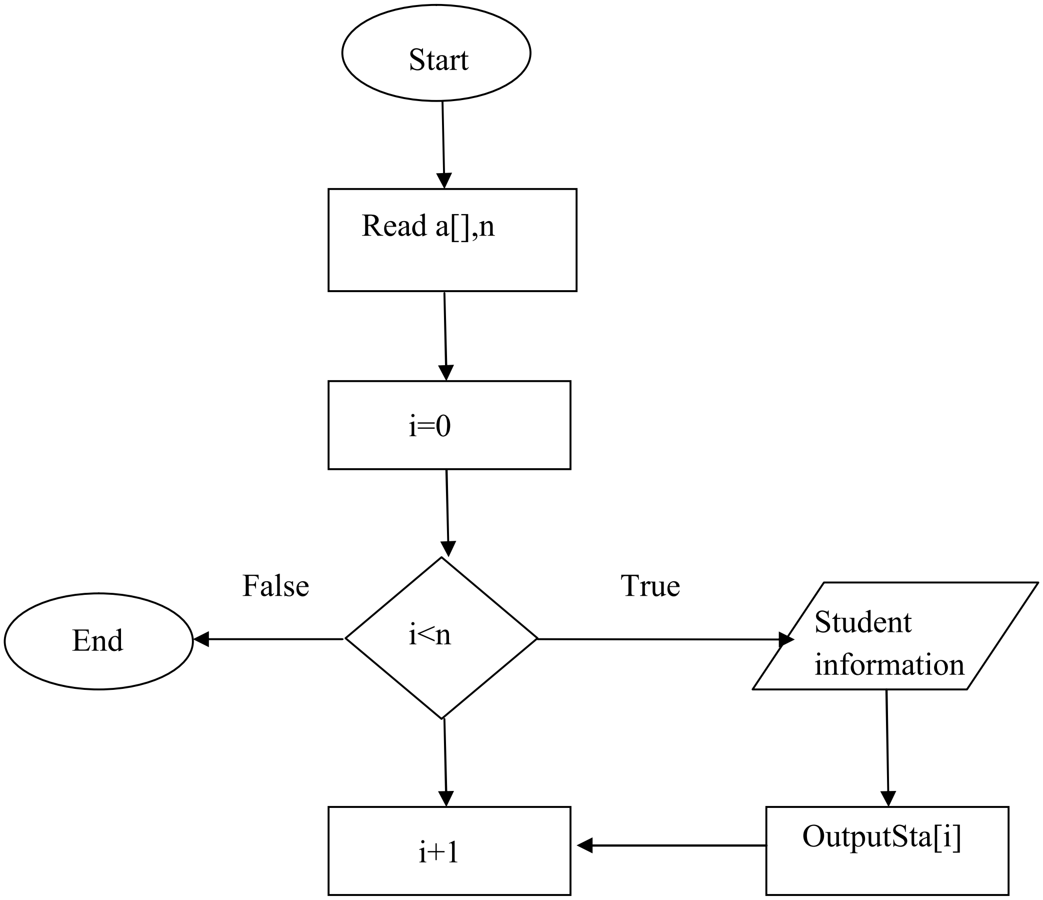


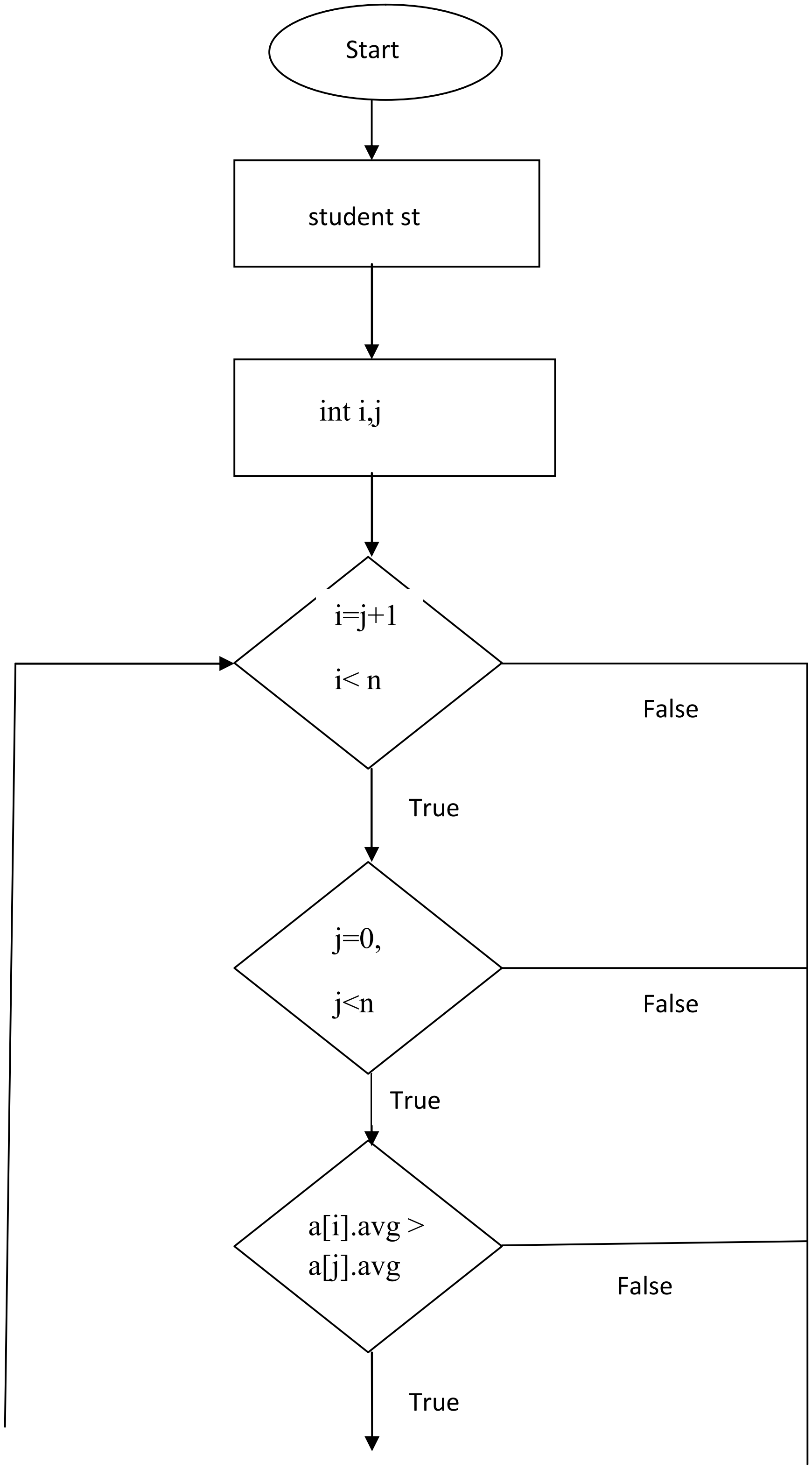
Explain:

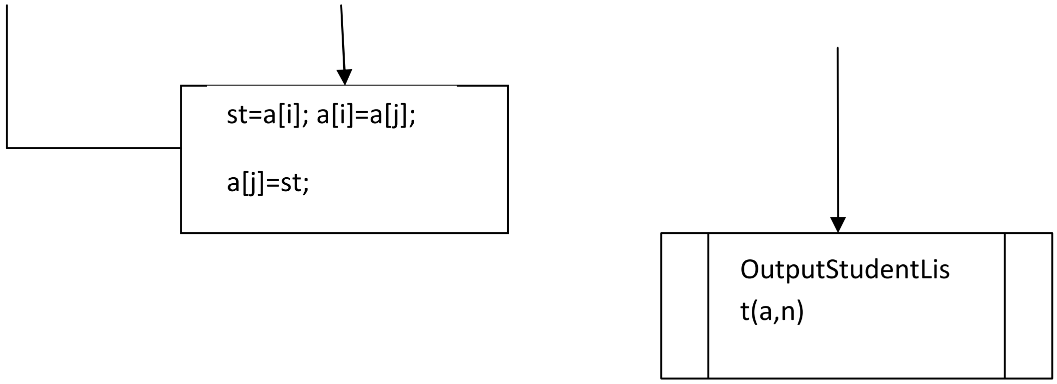
At first, enter student information and math,chemistry,physics scores.

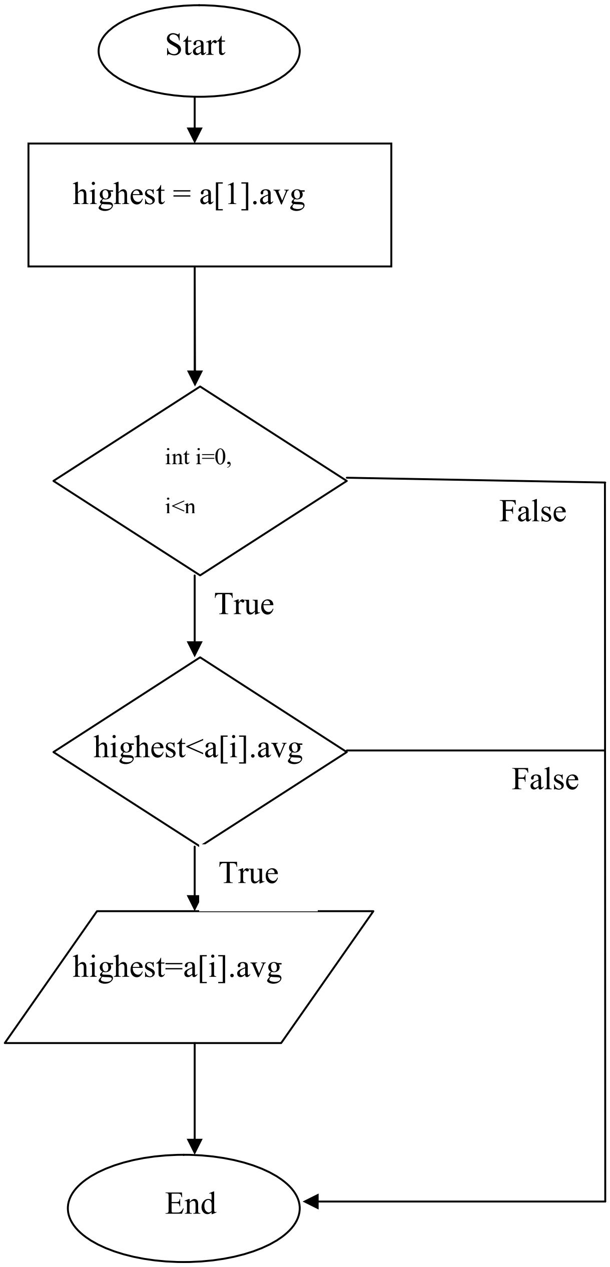
then calculate the average score.

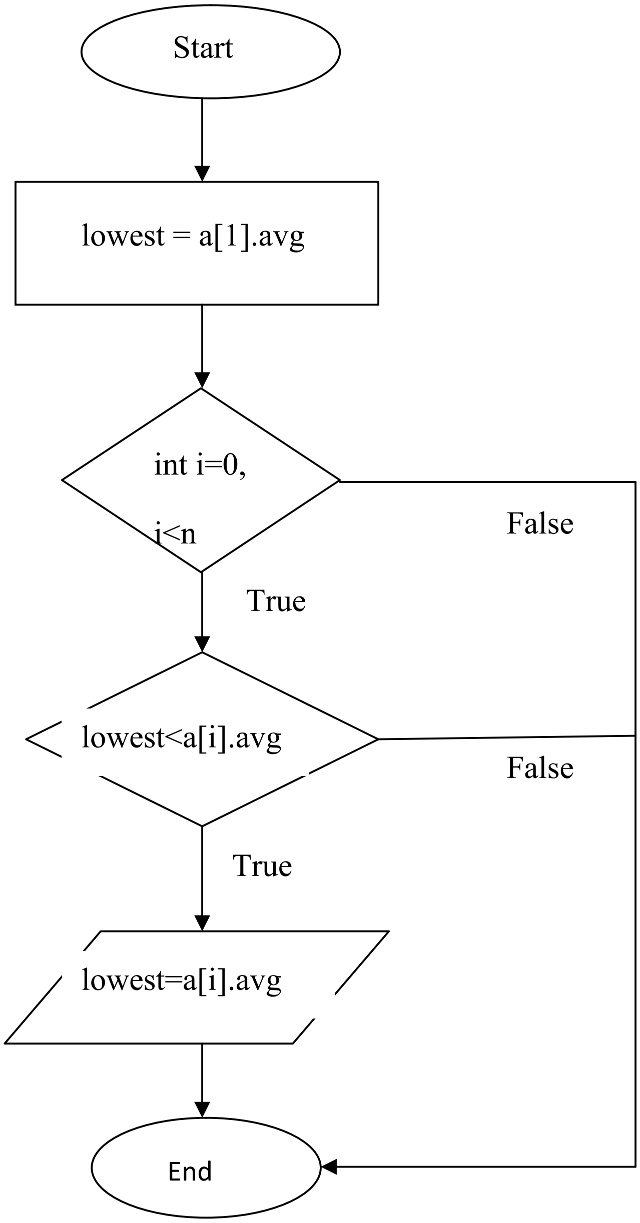
Finally rank 'Excellent', 'Good', 'Average' for students based on the average score.


Start
Read a[],n
i=0
i<n
False
True
End
Student information
OutputSta[i]
i+1


Start
student st
int i,j
i=j+1
i< n
False
True
j=0,
j<n
False
True
a[i].avg >
a[j].avg
False
True


st=a[i]; a[i]=a[j];
a[j]=st;
OutputStudentList(a,n)


Start
highest = a[1].avg
int i=0,
i<n
False
True
highest<a[i].avg
False
True
highest=a[i].avg
End


Start
lowest = a[1].avg
int i=0,
i<n
False
True
lowest<a[i].avg
False
True
lowest=a[i].avg
End

Task 4: Evaluation

I/ Advantages:
+ Can edit or add another problem or algorithm.
+ Easy to use.
+ It allows to move data freely around the system.
+ The program is divided into multiple sub-programs to make it easy for programmers to fulfill the requests that users need.

II/ Disadvantages:
+ Data security is not high.
+ Data is not cared in procedural programming

Task 5: Conclution

Managing student information is a very important job for every school. with an information management software that will help faculty manage students' grades effectively. From there, they can improve the quality of teaching.
Finally, procedural programming plays an important role in the creation of student management applications. I hope this software can help faculty manage students more effectively.

Bibliography

(2021, 10 22). Retrieved from https://www.learncomputerscienceonline.com/: https://www.learncomputerscienceonline.com/procedural-programming/

Cordrey, L. (2021, 10 22). Retrieved from https://popularask.net: https://popularask.net/which-of-the-following-is-a-main-characteristic-of-procedural-programming-language/?fbclid=IwAR1EOdIZqEnRmfwDA15-V1zZHECGGma92dS13gFxfN8_yiPUP5unXvtaX8M

Grant, A. (2020, 6 10). Retrieved from makeuseof.com: https://www.makeuseof.com/tag/high-level-low-level-programming-languages/

JavaTpoint. (2021, 10 22). Retrieved from javatpoint.com: https://www.javatpoint.com/classification-of-programming-languages

LearnComputerScience. (2021, 10 22). Retrieved from learncomputerscienceonline.com: https://www.learncomputerscienceonline.com/