ASSIGNMENT 2 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 0: Procedural Programming		
Submission date	09/14/2020	**Date Received 1st submission**	
Re-submission Date	21/09/2020	**Date Received 2nd submission**	
Student Name	Pham Cao Nguyen	**Student ID**	GCC18074
Class	GCC0801	**Assessor name**	Nguyen Hung Dung

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	CaoNguyen

Grading grid

P4	P5	M3	M4	D2

□ Summative Feedback: **□ Resubmission Feedback:**

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

Learning Outcomes and Assessment Criteria		
Pass	**Merit**	**Distinction**
LO3 Be able to implement procedural programming solutions **LO4** Be able to test procedural programming solutions		
P4 Write a program that implements the designed solution.	**M3** Program is written following coding standards, input data are validated	**D2** Evaluate your program, state lessons learnt and future improvements.
P5 Test the program with proper test plan.	**M4** Analyse test results for future maintenance.	

Submission Format
The submission is in the form of a Word document. You are required to make use of appropriate structure, including headings, paragraphs, subsections and illustrations as appropriate, and all work must be supported with research and referenced using the Harvard referencing system.

Contents

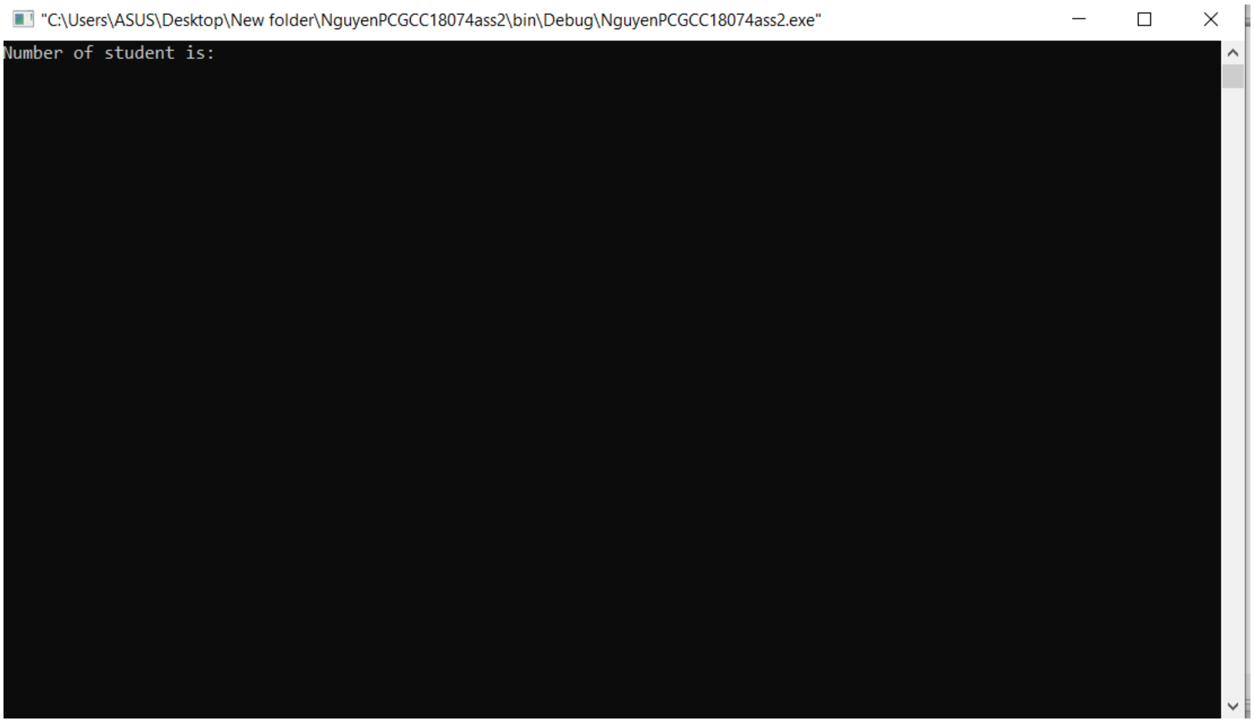
Scenario:

A math teacher wants to manage grades of a class. He asks you to help him to write a small application to do that. He needs to enter student IDs, student's grades and store these information into 2 separate arrays (integer array for IDs and float array for grades). Then he needs to print all student IDs together with their grades. Finally, he needs to know which student has highest grade and lowest grade. Your program should be menu based with the options above. When an option is done, the program should go back to the main menu so he can choose another option. There should be an option to quit program.

P4: Write a program that implements the designed solution.

- **Provide screenshot of your program when running.**

The program will appear as the user runs the program, the user enters the data to use.

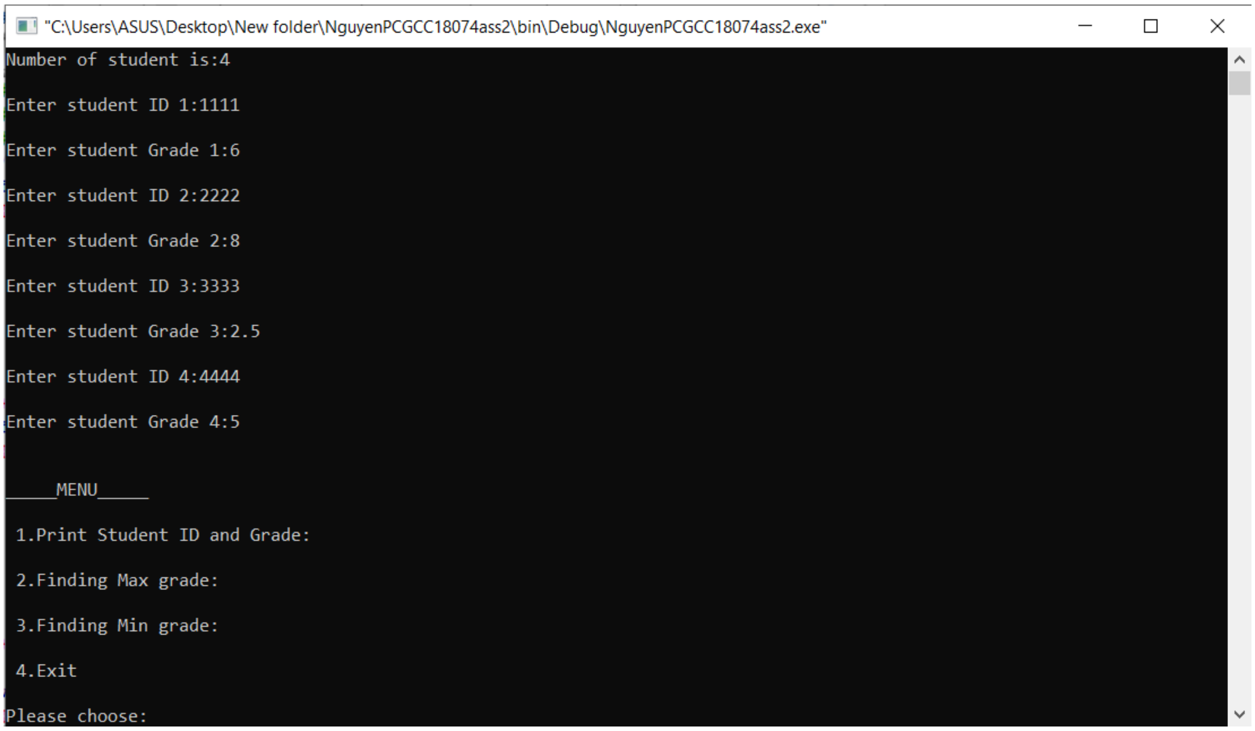


"C:\Users\ASUS\Desktop\New folder\NguyenPCGCC18074ass2\bin\Debug\NguyenPCGCC18074ass2.exe"

```
Number of student is:
```

- **Input data 's student.**

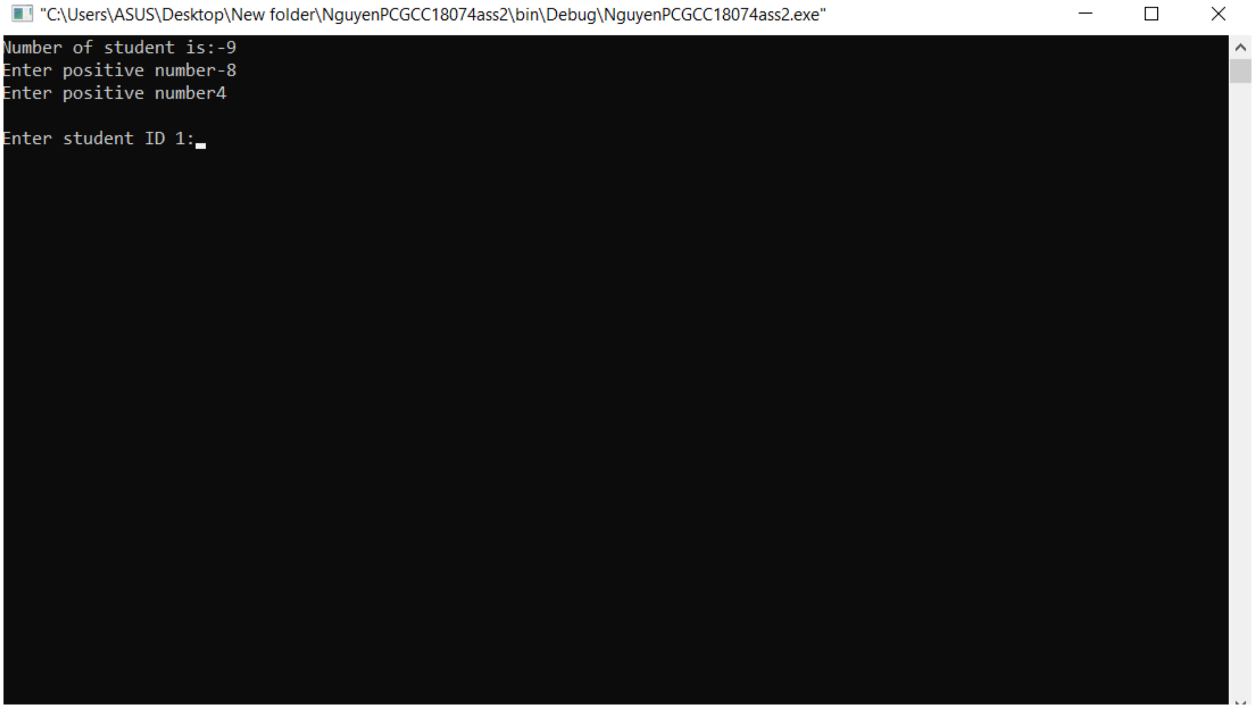
If a user want to show menu enter student ID, and student grade.


"C:\Users\ASUS\Desktop\New folder\NguyenPCGCC18074ass2\bin\Debug\NguyenPCGCC18074ass2.exe"

```
Number of student is:4

Enter student ID 1:1111

Enter student Grade 1:6

Enter student ID 2:2222

Enter student Grade 2:8

Enter student ID 3:3333

Enter student Grade 3:2.5

Enter student ID 4:4444

Enter student Grade 4:5


_____MENU_____

 1.Print Student ID and Grade:

 2.Finding Max grade:

 3.Finding Min grade:

 4.Exit

Please choose:
```

- **Enter student with negative number.**

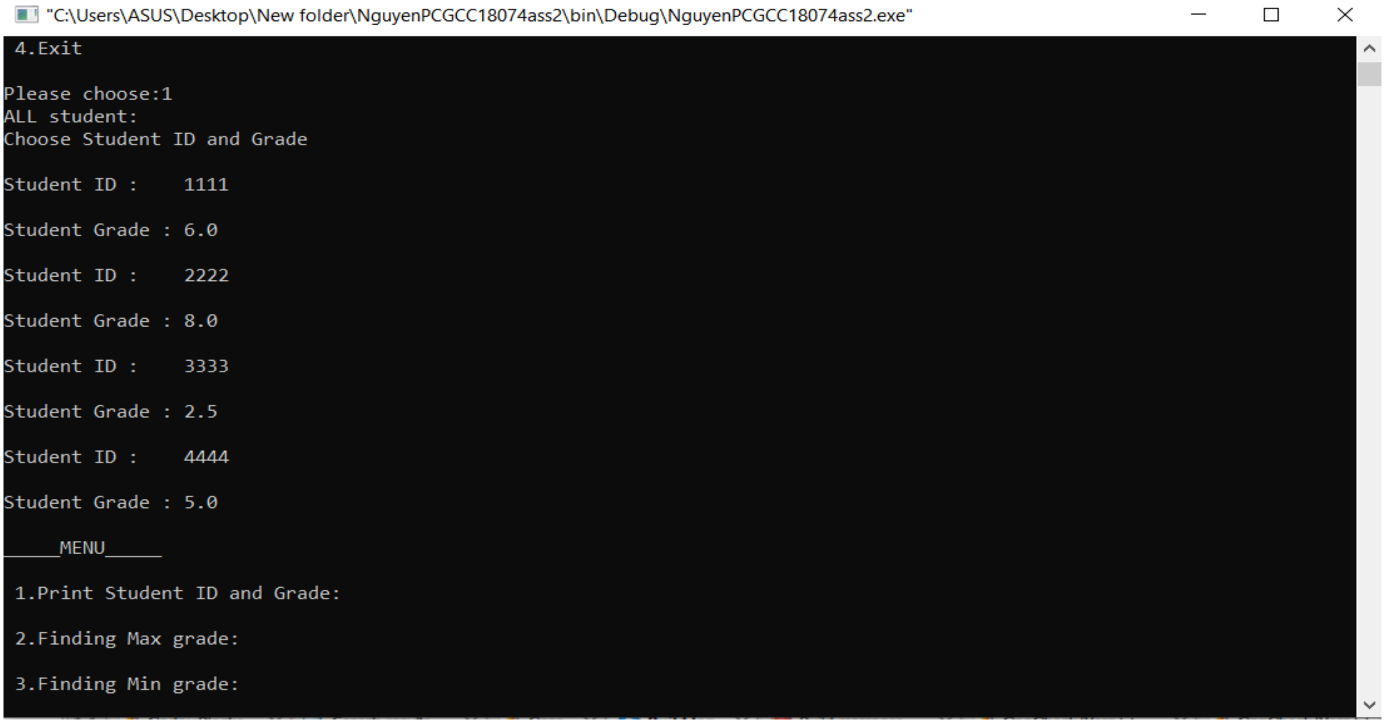
If the user enter number of student with negative number, the program will error.


"C:\Users\ASUS\Desktop\New folder\NguyenPCGCC18074ass2\bin\Debug\NguyenPCGCC18074ass2.exe"

```
Number of student is:-9
Enter positive number-8
Enter positive number4

Enter student ID 1:_
```

- **Display all students, grades, and ID.**

If user wants to display all students, grades, and ID numbers, choice 1.

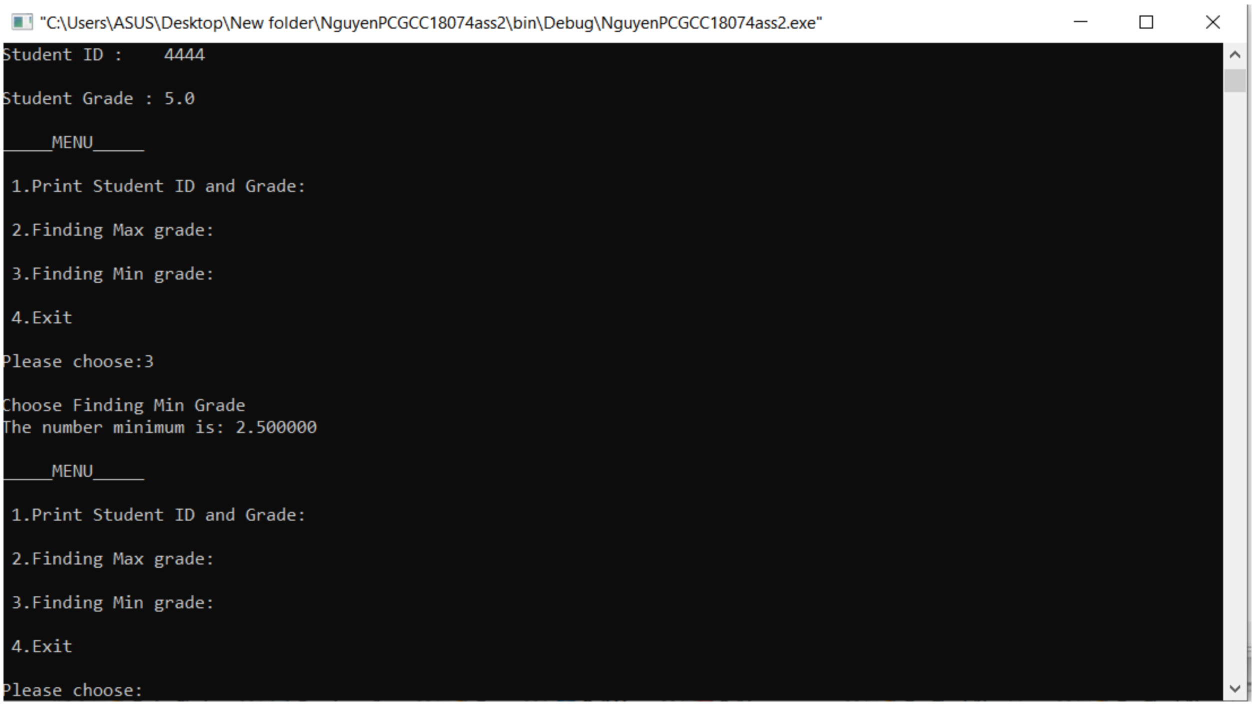


"C:\Users\ASUS\Desktop\New folder\NguyenPCGCC18074ass2\bin\Debug\NguyenPCGCC18074ass2.exe"

```
 4.Exit

Please choose:1
ALL student:
Choose Student ID and Grade

Student ID :    1111

Student Grade : 6.0

Student ID :    2222

Student Grade : 8.0

Student ID :    3333

Student Grade : 2.5

Student ID :    4444

Student Grade : 5.0

_____MENU_____

 1.Print Student ID and Grade:

 2.Finding Max grade:

 3.Finding Min grade:
```

- **Print min grade.**

If want to finding a minimum grade of students the user choice 3.

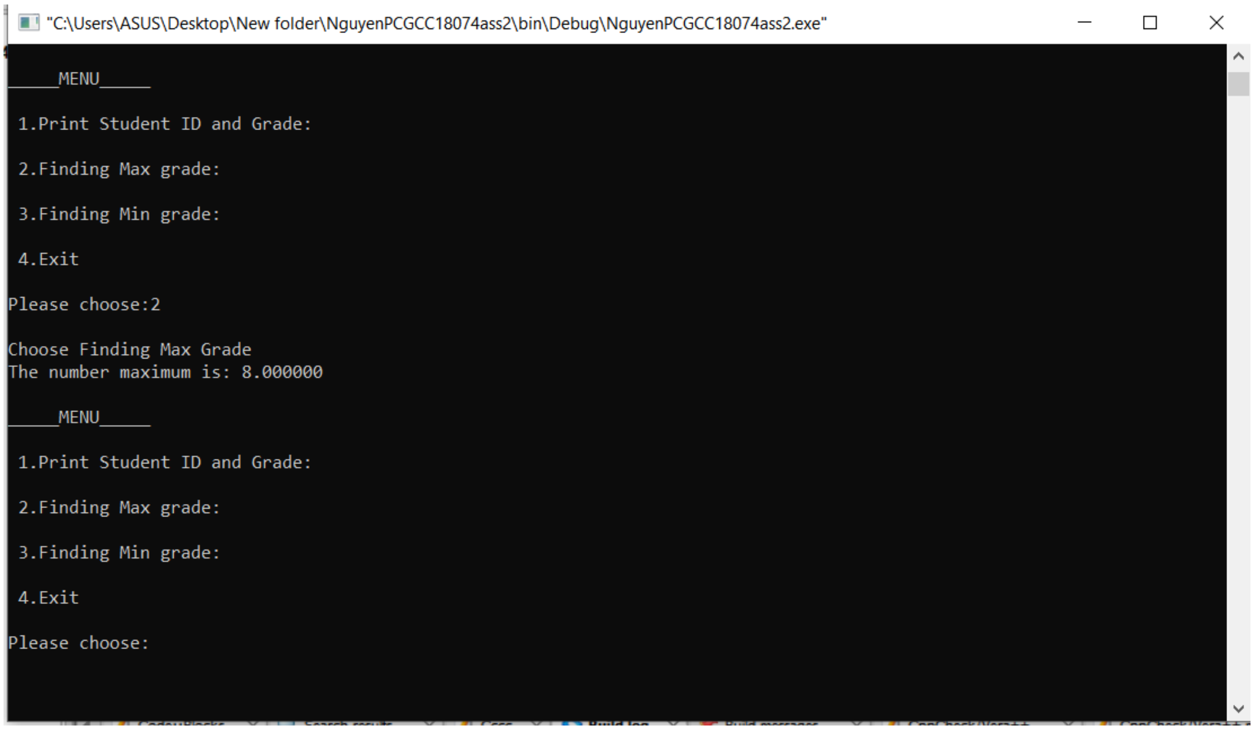


"C:\Users\ASUS\Desktop\New folder\NguyenPCGCC18074ass2\bin\Debug\NguyenPCGCC18074ass2.exe"

```
Student ID :    4444

Student Grade : 5.0

_____MENU_____

 1.Print Student ID and Grade:

 2.Finding Max grade:

 3.Finding Min grade:

 4.Exit

Please choose:3

Choose Finding Min Grade
The number minimum is: 2.500000

_____MENU_____

 1.Print Student ID and Grade:

 2.Finding Max grade:

 3.Finding Min grade:

 4.Exit

Please choose:
```

- **Print max grade.**

If want to finding maximum grade of students the user choice 2.

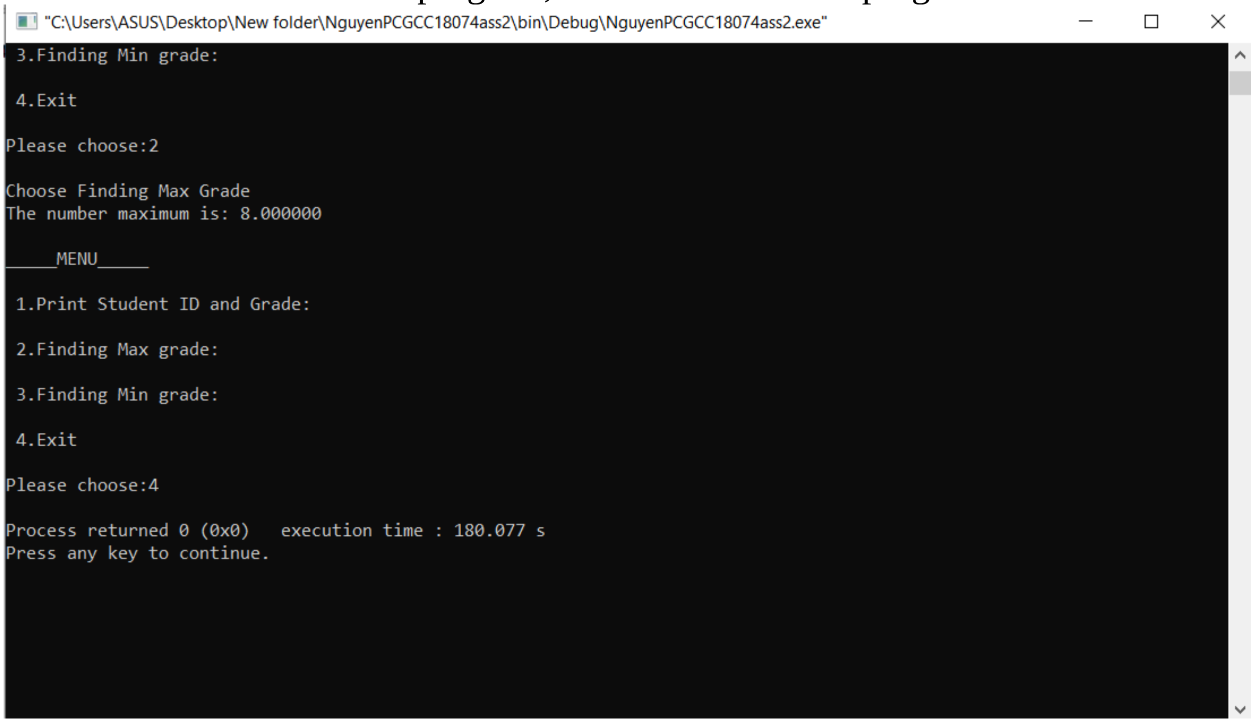


"C:\Users\ASUS\Desktop\New folder\NguyenPCGCC18074ass2\bin\Debug\NguyenPCGCC18074ass2.exe"

```
_____MENU_____

 1.Print Student ID and Grade:

 2.Finding Max grade:

 3.Finding Min grade:

 4.Exit

Please choose:2

Choose Finding Max Grade
The number maximum is: 8.000000

_____MENU_____

 1.Print Student ID and Grade:

 2.Finding Max grade:

 3.Finding Min grade:

 4.Exit

Please choose:
```

- **Print exit.**

If the user wants to end the program, choose 4 to exit the program.



"C:\Users\ASUS\Desktop\New folder\NguyenPCGCC18074ass2\bin\Debug\NguyenPCGCC18074ass2.exe"

```
 3.Finding Min grade:

 4.Exit

Please choose:2

Choose Finding Max Grade
The number maximum is: 8.000000

_____MENU_____

 1.Print Student ID and Grade:

 2.Finding Max grade:

 3.Finding Min grade:

 4.Exit

Please choose:4

Process returned 0 (0x0)   execution time : 180.077 s
Press any key to continue.
```

P5: Test the program with proper test plan.

No	Test case	Function	Input Data	Expected output	Actual output	Result
1.	Verify that Student information will save into array if user enters valid information	Input student ID and grade	- Enter number of students:4 - Student ID1: 1111 - Student grade: 6 - Student ID2: 2222 - Student grade: 8 - Student ID3: 3333 - Student grade:2.5 - Student ID 4:4444 - Student grade:5	Student information save into array: ID[]={1111,2222,3333,444} Grade[]={6,8,2.5,5}	Student information save into array: ID[]={1111,2222,3333,444} Grade[]={6,8,2.5,5}	Pass
2.	Verify that "Enter positive number" is displayed if user enters number of student with negative number.	Input student ID and grade	- Enter number of students: -9	Display error message: "Enter positive number"	Display error message: "Enter positive number"	pass
3.	Verify that "Invalid Student's Grade. Please enter again" is displayed if	Input student ID and grade	-Enter number of student: 4 -Student 1:1111 -Grade 1: 11	Display error message "Invalid Student's Grade. Please enter again"	Program do not display error message	Fail

	user enters invalid					

4.	Verify that “Student information will displayed when user choose menu “1.Print Student ID and grade”.	Display student ID and grade	ID[]={1111,2222,3333,444}
			Grade[]={6,8,2.5,5}
				Display list of student ID and grade:
				- Student ID 1: 1111
				- Student grade: 6
				- Student ID 2: 2222
				- Student grade: 8
				- Student ID 3: 3333
				- Student grade: 2.5
				-Student ID 4:4444
				- Student grade: 5
					Display list of student ID and grade:
					-Student ID1:1111
					-Student grade:6
					-Student ID 2:2222
					-Student grade:8
					-Student ID 3:333
					-Student grade:2.5
					-Student ID 4:4444
					-Student grade: 5
						Pass

5.	Verify that the highest grade will be displayed when user choose “2. Finding max Grade”	Finding student max grade	-Student grade[]={6,8,2.5,5}	Display: The number maximum is: 8.000000	Display: The number maximum is: 8.000000	Pass
6.	Verify that the lowest grade will be displayed when user choose “3.Finding min grade”	Finding student min grade	-Student grade []= {6,8,2.5,5}	Display: The number minimum is: 2.500000	Display: The number minimum is: 2.500000	Pass
7.	Verify that the the program will be exit when user choose “4.Exit”	Exit the program	Enter Choice 4	The program will be exited	The program will be exited	Pass

APENDIX:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
//For loop is used to repeatedly compare points and find the min or max value required.
float max_array(float max_array[],int n)
{
    int i;
    float max;
    max = max_array[0];
    for(i = 1 ; i < n ; i++)
        if(max < max_array[i])
        max = max_array[i];
    return max;
}
float min_array(float min_array[],int n)
{
    int i;
    float min;
    min = min_array[0];
    for(i = 1 ; i < n ; i++)
        if(min > min_array[i])
       min = min_array[i];
    return min;
}
void main()
{
    int n, i, choice, ID[30];
    float max, min, Grade[30];
```

```
  void output(int ID[], float Grade[], int n);
   printf("Number of student is:");
   scanf("%d", &n);
  if(n<0)
{
do
  {
  printf("Enter positive number");
  scanf("%d",&n);
  }
  while(n<0);
   }
  if(n>0)
  input(ID,Grade,n);
  do
//Menu
  {
    printf("\n\n_____MENU_____");
    printf("\n\n 1.Print Student ID and Grade: ");
    printf("\n\n 2.Finding Max grade: ");
    printf("\n\n 3.Finding Min grade: ");
    printf("\n\n 4.Exit");
    printf("\n\nPlease choose:");
    scanf("%d", &choice);
  switch(choice)
  {
  case 1:
    {
      printf("ALL student:");
```

```
            output(ID,Grade,n);
          }
          break;
      case 2:
          {
             printf("\nChoose Finding Max Grade");
             max = max_array(Grade, n);
             printf("\nThe number maximum is: %f", max);
          }
          break;
      case 3:
          {
             printf("\nChoose Finding Min Grade");
             min = min_array(Grade, n);
             printf("\nThe number minimum is: %f", min);
          }
          break;
      case 4:
          {
             exit (0);
          break;
      default:
          printf("\nYou need to enter the option less than 5.");
          break;
          }
      }
      }
      while(choice!=4);
  }
```

```
input(int ID[], float Grade[],int n)
{
    int i;for(i = 0; i < n; i++)
    {
       printf("\nEnter student ID %d:", i + 1);
        scanf("%d",&ID[i]);
        printf("\nEnter student Grade %d:", i + 1);
        scanf("%f",&Grade[i]);
    }
}
void  output(int ID[], float Grade[], int n)
{
    int i;
     printf("\nChoose Student ID and Grade");
            for(i = 0; i < n; i++)
            {
                printf("\n\nStudent ID :\t%d", ID[i]);
                printf("\n\nStudent Grade :\t%.1f", Grade[i]);
            }
}
```

References

Cognizant, 2007. *Problem solving and C progamming.* s.l.:s.n.