ELET 4309 **Homework No. 2** **Dr. F. Attarzadeh**
40 Points **Due 02/26/2019** **02/12/2019**
9 PM

It is very important that you read the notes at the end of each assignment for this and all other assignments.

Develop a program flowchart and then write a menu-driven **C++** program that will solve the following problem. The menu is shown below. Note: the letters shown as bold are for emphasis and you do not have to make them bold in your program. The program uses arrays, references, pointers, and functions to accomplish the tasks outlined below.

Help **I**nputData **M**inimum ma**X**imum mi**N**max **Q**uit

Upon program execution, the screen will be cleared and the menu shown above will appear at the top of the screen and properly centered. The menu items are explained below.

H or **h** (for **H**elp) option will invoke a function named **help()** which will display a help screen. The help screen(s) should guide the user how to interact with the program, type of data to be entered, and what results would the program produce. Each help screen should remain on the monitor until the user strikes any key (e.g., strike the space bar followed by a carriage return). Once the user strikes a key, the screen will be cleared and the menu is displayed again. Restrictions on the options **M**, **m, X**, **x, N,** and **n** would be mentioned here.

I or **i** (for **I**nputData) option will prompt the user for the number of inputs to be tested. The program uses this number to fill a **double** array declared to store up to 35 elements. This option must be executed before options **M**, **X**, and **N**. A boolean flags are to be used here. See data to be used later on this assignment.

M or **m** (for **M**inimum) option will compute and display the minimum and frequency of occurrences of this minimum on the screen. This option implements the following two function prototypes:

```
double& minfunc(double& d1, double& d2);

void  frequency(double& d1, short& freq);
```

The function ***minfunc()*** will receive two references, representing two data items, and returns a reference to the minimum to the calling function. Function ***frequency()*** will compute the frequency of occurrences for the minimum number in a data set. This function may be called from within ***minfunc()*** function. Once the user viewed the results, striking any key will clear the screen followed by the menu display. This option must be executed after option **I or i**.

X or **x** (for ma**X**imum) option will compute and display the maximum and frequency of occurrences of this maximum on the screen. This option implements the following two function prototypes:

```
double& maxfunc( double& d1, double& d2);

void  frequency( double& d1, short& freq);// this is the same as above
```

The function ***maxfunc()*** will receive two references, representing two data items, and returns a reference to the maximum to the calling function. Function ***frequency()*** will compute the frequency of occurrences for the maximum number in a data set. This function may be called from within ***maxfunc()*** function. Once the user viewed the results, striking any key will clear the screen followed by the menu display. This option must be executed after option **I or i**.

N or **n** (for mi**N**max) option will compute and display the minimum, the maximum, and their frequencies of occurrences. This option uses the functions mentioned above and the following function prototype:

```
void minmax( double* minv, short* minFreq, double* maxv, short* maxFreq);
```

the function ***minmax()*** receives values calculated before and processes them accordingly. The minimum value, the maximum value, and their frequencies of occurrences are displayed on the monitor. Once the user viewed the results, striking any key will clear the screen followed by the menu display. This option must be executed after options **M** and **X**. Again, you need to have mechanism in place so that this condition is met.

Q or **q** (for **Q**uit) option will clear the screen and returns the control to the Visual Studio IDE.

Sample Data to be used is shown below. Of course, your program will work with any set of numbers within the specified range of elements.

Test your program with the following data sets and make sure your sample runs include them. Your program uses 2 digits after the decimal point when displayed.

set 1 :	53.6, -31.12, 66.22, 34.21	set 2 :	-42.0, 34.0, -62.0, 32.0, -45.0
set 3 :	-70.0	set 4 :	43.6, -21.5, -32.34, 46.9, 43.6, -12.54
set 5 :	0.0, 0.0, 0.0	set 6 :	6.2, 6.2, 6.2, 6.2
set 7 :	-5.0, -5.0, -5.0, -5.0, -5.0	set 8 :	0.0

Grading:

Flowchart	6 points
Documentation	6 points
Program completeness	24 points, this includes program correctness, efficient programming, using right constructs for the solution, and proper use of coding as emphasized in class
Correct sample outputs	4 points

Notes:(please read very carefully)

1. Make sure the files you upload to the Blackboard are **VIRUS FREE**!(grade of 0 will be given for infected media). Use Technology lab PCs for the test.
2. Comment your program.
3. Use meaningful prompts
4. Provide a brief description of the problem being solved
5. Be sure to include a header file at the beginning of your program as shown in the course syllabus and how to submit your homework document.
6. **NO global declarations of VARIABLES allowed** in any program that you develop in this course.
7. ***arrays*** are passed to functions by reference.
7a. Upon startup, options **M**. **X**, and **N** should not be functional. Very important tests needs to be implemented here.
8. Full function prototyping is required. Functions must have their purposes fully explained.
9. Use pointers, references, and functions discussed in class.
9a. Float type data must have two digits after the decimal point using the ***fixed*** point flag and other flags as needed.
9b. Sample runs will include *all* data shown above.
10. If the number of elements of the array is 0 or negative, your program will flag that input as invalid, warns the user, clears the screen, and prompts the user again.
11. Parameter passing to functions is as specified. Functions may call other functions, as needed, to facilitate modular programming. You may *slightly* modify the function prototypes to fit your design requirements, but minimum function prototypes shown above shall not be compromised. No more than 5 parameters should be passed to functions.
12. Illegal menu selection must be handled properly without terminating the program.
13. At the due date, submit your **H3 containing the components of the program specified in the guidelines.** Create a Word file that contains the header, the flowchart, the list of your .cpp file, and the sample run of the program. Name this file H3.docx. The source file for H3.cpp and the Visio file H3.vsdx will be uploaded as well. Unrelated files should not be present when you upload them to the Blackboard. Homework must be uploaded to Blackboard by **9 PM** of the due date and ***late homework will not be accepted***.
15. Use **Microsoft Visual Studio Enterprise 2015** compiler using default compiler settings.
16. Use **Microsoft Visio 2013** to develop your flowchart.
17. Adherence to the ***ANSI C++*** required.
18. ***Do not*** use **<stdio.h> and <conio.h>** in this assignment and all other assignments.
19. ***Do not*** use any **#define** in your program until the time that is required for class declaration header files.
20. No ***goto*** statements allowed in any program that you develop in this course.
21. ***Non-compliance with these notes will cost you points***.
22. No collaboration on this assignment and all other assignments allowed. If you violate this policy, your grade for the course will be **F**.
23. Homework is due in class at 9 PM on the due date and ***late homework will not be accepted***.