Programming HW #7 – 100 Pts

Write a program using 32-bit MASM x86 Assembly Language that does the following:

a. Prompt the user to enter a decimal number with at least 5 decimal places. Also allow for the case where the user can enter numbers greater than 1.
b. Read the user input and display on the screen. Something like the following… The original number entered was *x.xxxxx*.
c. **Prompt the user to enter a precision value between 1 and 4.**
d. Read the user input and display on the screen. Something like the following… You selected to set the precision to *x* decimal places.
e. Round to the specified precision setting and display on the screen. Something like the following…

Original Number: 34.26598

Precision – 3: 34.266 ##Depending on the precision value selected.

f. Include error messages in the event the user indicates a precision value that is not 1, 2, 3, or 4 and then prompt the user to re-enter the precision.
g. Include a non-case sensitive repeat option.
h. **You may not use the rounding instruction for this assignment.**

Suggested Approach

The most common method used for this assignment is to

- Multiply by the appropriate power of 10.
- Convert the value to an integer.
- Convert the integer back to a floating-point value.
- Divide by the appropriate power of 10.

For example, if your value is 1.23456 and the precision is 2

- Multiply by 100: 123.456
- Covert the value to an integer: 123
- Convert the value back to a floating point: 123.00…
- Divide by 100: 1.2300…

The floating-point instructions are in Section 12.2

Note: Floating Point Values will be displayed in Scientific Notation in the console window.

Extra Credit: Display the number correctly in Decimal Notation (not Scientific Notation)

Programming HW #7 – 100 Pts

Do not use magic numbers. Name and initialize variables in the .data section of your programs.

Submit your **source code (.asm)** files to the assignment drop-box by the due date. **Submit a single .asm file.** Do not submit your .sln project file. Do not submit Word or PDF files containing your code.

To ensure consistent grading, please include all necessary INCLUDE and INCLUDELIB statements directly within your code, exactly as directed in the course Syllabus, Policy Contract, and recent announcement. I copy and paste submissions into a standard Visual Studio project for evaluation. Due to the volume of submissions, I'm unable to modify code to accommodate individual Visual Studio configurations. This requirement also helps maintain academic integrity by ensuring only approved files and libraries are used.

Build and debug your code using Visual Studio 2022.

Comment Requirements:
There must be a preamble block of comments at the beginning of the program that includes at least the following information: Your name, course number/section/title, program title, and date

There must be a block of comments before the main procedure of your program, as well as before each additional procedure (if applicable) that describes the inputs, outputs, memory usage, register usage, and functional description of the procedure.

The program must also contain **non-syntax based individual line comments for EACH line** that make the functional goals and processes of the program self-documenting. These comments should be placed in line with each instruction (not above or below). Use tabs to line things up!

Grading

Functionality: 35 Pts
Output Formatting: 15 Pts
Program Structure and Coding Concepts: 25 Pts
Comments and Internal Documentation: 25 Pts