CPEN 3710 **Lab Exercise 11**

Real Number Processing and Linking to High-Level Language Programs

Required Materials:

- Your textbook, *Assembly Language for x86 Processors* (7th edition)
- Removable or network device (Flash drive, memory card, MyMocsNet account mapped to a drive letter, etc.) for storage of your programs
- These instructions
- Intel-compatible, Windows-based personal computer (like the ones in EMCS 306) with text editor, MASM, and Microsoft Visual Studio (or other available high-level language compiler of your choice)

Preparation for Laboratory:

Read the material on *floating-point binary representation* and the Intel *floating-point unit* in Sections 12.1 and 12.2, pages 511-539 of your textbook. Also read the material on *interfacing assembly language code to high-level languages* in Chapter 13, pages 555-586 of your textbook.

Instructions:

Write an assembly language procedure that will use Intel floating-point instructions to perform the computations required to evaluate the **quadratic formula** (found in any algebra textbook). In other words, given the real numbers **a**, **b**, and **c** which represent the coefficients of the terms in the equation $ax^2 + bx + c = 0$, the procedure must solve for the two roots of the equation. This procedure must be called from a high-level language main program (written in C, C++, Java, Pascal, Fortran, or any other high-level language approved by the instructor for which you have access to a compiler) which prompts the user for the values of the real numbers **a**, **b**, and **c**; passes these three values (as well as pointers **root1ptr** and **root2ptr** to the two memory locations used to return the roots) to the procedure using the standard mechanism supported by the high-level language compiler (most likely through a stack frame); and (after the called procedure returns) displays the results (the two real roots of the quadratic equation) computed by the assembly language procedure. If the roots of the quadratic are complex (if the number under the radical in the quadratic formula is negative), the called procedure should return an error code of -1 to the calling program (typically this is done in register EAX), which should then display a message to that effect. If the roots are real, the assembly procedure should use the supplied pointers to store the two roots in memory and return 0 to the caller, which will then display the real roots of the quadratic equation to the user.

To Hand In: *(due by 3:00 p.m. Tuesday, November 27)*

1. Turn in a printed copy of your ***thoroughly commented assembler listing*** *(.LST file)* for the called procedure, and your ***thoroughly commented high-level language program*** *(.C, .CPP, .java, .PAS, or other source code file as appropriate)*. **Be sure to follow the guidelines given in the programming style and documentation handout.**

2. Submit the ***results*** of your program as follows: run it from the command prompt and capture a “screen shot” of the output produced by the program. Do this for at least **three test cases**: once where the quadratic has **real, unequal roots**; once where the quadratic has **real, equal roots**; and once where the quadratic has **complex roots**.

3. Have the instructor check the operation of your program and **sign in the space below** when you have demonstrated its operation.

Instructor’s signature: ____

Staple your **HLL and Assembly program listings**, **results**, and this **signed sheet** together. Submit these items by the date and time specified above. Late submissions will be penalized substantially.