91.203 Assembly Language Programming
Homework #4 Due Friday, March 4 , 2005

February 25, 2005

- This assignment is due on **Friday, March 4**.

- You must submit your **course cover-sheet**, a short **write-up** (text) of your results, and your **source code**, along with an example of a run, with **output** showing intermediate and final results as described below. The problem you must solve has been described in class and is formalized as follows:

- This assignment will require you to write a program in assembly language to calculate Fibonacci numbers. The first attempt at this program will require that you build a subroutine (or function) to calculate the Fibonacci value of the argument N using the following definition of Fibonacci numbers:

  Fibonacci(0) = 0

  Fibonacci(1) = 1

  Fibonacci(N) = Fibonacci(N-1) + Fibonacci(N-2)

  The sequence of numbers looks like:

N	0	1	2	3	4	5	6	7
Fib (N)	0	1	1	2	3	5	8	13

- The subroutine you will build must compute the required Fibonacci number using a **recursive implementation**. The Fibonacci number for 20 is 6765, which fits well within our 16 bit 2s complement integer environment, so we expect you to be able to try N values up to fib(20).

- Your program should set up **data locations for N** values from 0 to 20 and **corresponding locations** for the Fibonacci numbers for these values. Your main routine should call your subroutine (function) passing each value in turn and placing the result in the location reserved for it. You can use the **halt** command between each call to your subroutine and from the debugger interface, print out the return value that you've stored from the previous call during debugging, but you probably will want only a single halt at the end of the program to show your results once you have the program running correctly.

- You must submit a hardcopy of your source code and of your output results, with each file **clearly labeled**. You must also include a (**2 page maximum) short write-up** to describe your results and success level.

- Indicate **clearly** on the **cover-sheet** and in your short write-up, how much of this assignment you feel you were able to do correctly, and what, if anything, you were not able to do. **Failure to specifically provide this information will result in a 0 grade** on your assignment. If you do not disclose problems on the cover-sheet and in your write up and problems are detected when your program is tested, you will receive a grade of 0.

- Your submitted code may be compiled and executed for testing purposes, so make sure your **cover-sheet** includes the path names to your files.

PARTIAL OUTPUT SHOWN BELOW FROM A HALT INSTRUCTION:
Total cycles : 201409

If you would like to examine memory enter y if not enter n: y
enter decimal address, q to quit or c to continue: 121
the location 121 has value 0000000000000000 , or 0 or signed 0 ← Fib(0)
enter decimal address, q to quit or c to continue: 122
the location 122 has value 0000000000000001 , or 1 or signed 1 ← Fib(1)
enter decimal address, q to quit or c to continue: 123
the location 123 has value 0000000000000001 , or 1 or signed 1 ← Fib(2)
enter decimal address, q to quit or c to continue: 124
the location 124 has value 0000000000000010 , or 2 or signed 2 ← Fib(3)
enter decimal address, q to quit or c to continue: 125
the location 125 has value 0000000000000011 , or 3 or signed 3 ← Fib(4)
enter decimal address, q to quit or c to continue: 126
the location 126 has value 0000000000000101 , or 5 or signed 5 ← Fib(5)
enter decimal address, q to quit or c to continue: 127
the location 127 has value 0000000000001000 , or 8 or signed 8 ← Fib(6)
enter decimal address, q to quit or c to continue: 128
the location 128 has value 0000000000001101 , or 13 or signed 13 ← Fib(7)
enter decimal address, q to quit or c to continue: 129
the location 129 has value 0000000000010101 , or 21 or signed 21 ← Fib(8)
enter decimal address, q to quit or c to continue: 130
the location 130 has value 0000000000100010 , or 34 or signed 34 ← Fib(9)
enter decimal address, q to quit or c to continue: 131
the location 131 has value 0000000000110111 , or 55 or signed 55 ← Fib(10)
enter decimal address, q to quit or c to continue: 132
the location 132 has value 0000000001011001 , or 89 or signed 89 ← Fib(11)
enter decimal address, q to quit or c to continue: 133
the location 133 has value 0000000010010000 , or 144 or signed 144 ← Fib(12)