CSE 2312: Computer Organization & Assembly Language Programming
Summer 2015
Program #2

In this assignment, you will implement an iterative solution for computing the GCD of two positive integers. Your program, at a minimum, will consist of the following procedure call:

> GCD_ITERATIVE: Computes the GCD of integers stored in R1 and R2 iteratively and returns the result in R0. An iterative solution for computing the GCD in C code is given below (you do not have to directly port this example to assembly, but you must iteratively solve the problem):

```
int gcd_iterative(int x, int y)
{
     while((x % y) != 0)
     {
          y--;
     }
     return y;
}
```

Your main function will contain a loop that continuously checks for keyboard input in the following pattern:

```
<OPERAND_1><ENTER>
<OPERAND_2><ENTER>
```

Once the 2 lines of input are acquired, the operands should be loaded into the proper registers and the GCD_ITERATIVE procedure should be called. The procedure should return the result in register R0, and the main function should print the value to the console and skip to a new line.

All input test cases will consist of positive numbers only. The value stored in `OPERAND_1` will always be greater than or equal to `OPERAND_2`. Below are some example use cases:

```
24              100             500             200
12              75              500             150
GCD: 12         GCD: 25         GCD: 500        GCD: 50
```

Points will be assigned as follows:

1. Main function correctly retrieves 2 input parameters, prints result in a continuous loop (20 points)
2. GCD_ITERATIVE procedure implemented, registers R0, R1, R2 used as specified (40 points)

3. GCD_ ITERATIVE procedure returns correct value in all cases
(40 points)

Submit your solution as a single “.s” file to Blackboard. Name the file “p2_1000xxxxxx.s”, where 1000xxxxxx is your UTA ID number.

*** Be sure to check http://github.com/cmcmurrough/teaching/assembly for useful code snippets ***