CSE 2312: Computer Organization & Assembly Language Programming
Fall 2016
Program #2

In this assignment, you will implement a recursive solution for computing the GCD of two positive integers. You may use the following C solution as a guide:

```
int gcd(int n1, int n2)
{
    if (n2 != 0)
       return gcd (n2, n1 % n2);
    else
       return n1;
}
```

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

```
<OPERAND_N1><ENTER>
<OPERAND_N2><ENTER>
```

Once the 2 lines of input are acquired, the operands should be loaded into the proper registers and the `gcd` procedure should be called. The procedure should return the result in register R0, and the main function should print "the gcd of *n1* and *n2* is *x*" (where n1, n2, and x are replaced with the proper numbers) and skip to a new line.

All input test cases will consist of positive numbers only. An example test case and its output is provided below:

```
100
40
The GCD of 100 and 40 is 20
```

Points will be assigned as follows:

1. Main function correctly retrieves 2 input parameters, prints result in a continuous loop
   (20 points)
2. `gcd` procedure implemented, registers R0, R1, R2 used as specified
   (30 points)
3. `gcd` procedure returns correct value in all cases
   (50 points)

Submit your solution as a single “.s” file to Blackboard. Name the file “p2_XXXYYYY.s”, where XXXYYYY is your UTA NetID.

*** Be sure to check http://github.com/cmcmurrough/cse2312 for useful code snippets ***