CSE 2312: Computer Organization & Assembly Language Programming
Spring 2018
Program #3

In this assignment, you will implement a recursive solution for computing the number of integer partitions of a positive integer n with parts up to m. In number theory, an integer partition is a way of writing n as a sum of positive integers (in this case, positive integers up to m). For example, the partitions for $n = 5$ and $m = 3$ are:

$$5 = 1 + 1 + 1 + 1 + 1$$
$$5 = 2 + 1 + 1 + 1$$
$$5 = 2 + 2 + 1$$
$$5 = 3 + 1 + 1$$
$$5 = 3 + 2$$

Thus, there are 5 integer partitions for $n = 5$ and $m = 3$. This problem can be solved recursively using the following function:

```
int count_partitions(int n, int m)
{
  if (n == 0)
    return 1;
  else if(n < 0)
    return 0;
  else if (m == 0)
    return 0;
  else
    return count_partitions(n - m, m) + count_partitions(n, m - 1);
}
```

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

<OPERAND_N><ENTER>
<OPERAND_M><ENTER>

Once the 2 lines of input are acquired, the operands should be loaded into the proper registers and the count_partitions procedure should be called. The procedure should return the result in register R0, and the main function should print "there are x partitions of n using integers up to m" and skip to a new line.

All input test cases will consist of positive numbers only, and `OPERAND_M` will always be less than or equal to `OPERAND_N`. Below is how the input/output should look for the example above:

```
5
3
There are 5 partitions of 5 using integers up to 3
```

Points will be assigned as follows:

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

Submit your solution as a single ".s" file to Blackboard. Name the file "abc1234_p2.s", where abc1234 is your UTA NetID.

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