COMPUTER EXPERIMENTATION #1 – MATH 5405
SPRING 2016

DUE: NOT DUE

Let's make a simple program in python. Everyone choose a partner (if we have an odd number of people, there will be a group of 3).

Step 1 Open up a terminal.

Step 2 Make a new file that will hold the functions procedures you will write. The file should end with ".py". For instance you could do

```
gedit MyFunctions.py &
```

which will open an editor and edit a file. You may use another editor (instead of `gedit`) if you want.

Step 3 Run

```
python
```

This is the python interpreter. There are other options to make things prettier you can install on your own machine (google "python IDE").

Step 4 You can run various commands here like:

```
>>> print "hello world"
>>> 17//4
>>> 17%4
>>> a = 2
>>> if (a == 2):
...     print("We know a is 2")
...
>>> if (a == 3):
...     print("We know a is 3")
... else:
...     print("we know a is not 3")
...
```

The tabbing here is important! Experiment a little bit.

Step 5 Unfortunately, anything you do here will be lost. That's why we made that file. Let's create a function that checks if an argument (input) is equal to 3. However, instead of putting it in the interpreter directly, we will put it in the file for later use. Here's how I made my file (notice the tabbing).

```
def IsThree(n):
  if (n == 3):
    print("the number is 3")
    return True
  else:
    print("the number is not 3")
    return False
```

Save your file. Within the interpreter, you can load the file and test it out by calling.

```
>>> import MyFunctions
>>> MyFunctions.IsThree(5)
>>> MyFunctions.IsThree(3)
```

Step 6 Now, work with your partner to write a function that computes a gcd. This is already built into a standard math library, but let's do it for practice. Here is roughly how you can do it.

(1) Your function should take two inputs, ie `def gcd(a,b):`
(2) It should return the gcd.
(3) It's easy to make a recursive function (a function that calls itself). In other words, if $a > b$ compute $r = a \bmod b$ and then have your function call `gcd(b,r)` (assuming the remainder is nonzero).
(4) You'll need some `if` and `then` statements.

Step 7 Now make a function that will find the integers x and y such that $\gcd(a, b) = ax + by$. Put it into your library.