This week's lab will focus on some of the skills that you will need to complete your second assignment. As in previous weeks, you will work in teams of two: one of you will be the *driver* and the other will be the *navigator*. The *driver* is the person who will perform entry on the system. The *navigator* is the person who tells the driver what to enter. You must alternate roles throughout the lab. Take advantage of this situation – bounce ideas off each other and discuss things when they do not seem clear. If you get stuck, raise your hand and your TA will help you as soon as they are free.

Once again – remember that the programming is moot: it is the problem-solving and understanding that determines how good our solution will be. Take problems, create a plan, and program it. Do not hesitate to create pseudo-code or flowcharts to better illustrate your plan before you code it.

Here are some Python programming hints:

1. Lists:
   - To create a blank list use the `list()` function:
     ```
     a = list() # a is an empty list
     ```
   - To add an item to a list use the `append(<item>)` function with the list identifier:
     ```
     a.append(123) # appends the number 123 to the end of the list a
     ```
2. String processing:
   - to concatenate two strings use the `+` symbol (plus)
     ```
     a = 'how are'
     b = ' you'
     c = a + b  # c now has a value of 'how are you'
     ```
   - to convert a number to a string use the `str(<number>)` function
     ```
     x = 123
     y = 'abc'
     z = y + str(x)
     ```
   - to convert a string to an integer use the `int(<string>)` function
     ```
     x = '123'
     y = 2
     z = int(x) * y # z should have a value of 246
     ```
   - to select a part of a string use a : (colon). Use the position (number) of the letter you wish to start at before the colon and the position (number) of the letter you wish to stop before after the colon. If you want to start at the beginning, put nothing before the colon; to select to the end of the string, put nothing after it.
     ```
     a = 'abcdefg'
     b = a[1:3] # b contains 'bc'
     b = a[:3]  # b now contains 'abc'
     b = a[3:]  # b now contains 'defg'
     ```
3. The modulus function (%):
   Given any two numeric expressions `x` and `y` and the modulus operator `%`, the modulus operator, (also known as the remainder operator,) divides `x` by `y` and returns only the remainder.
   e.g. `5 % 2` is `1`
   `(2 + 5) % 4` is `3`
   `15 % (5 + 5)` is `5`
   `3 % 3` is `0`
4. To call a function from another program file:
   First import the file that contains the function you required. e.g., `import lab5`
   Then, assuming you have a function named `func1()` in the file `lab5.py`, you can execute (call) it as follows:
   `lab5.func1()`

Now you must open IDLE or Wing and create the following functions:

Step 1

- Write a function called `replace(oldString, oldChar, newChar )` which accepts a string, replaces every occurrence of `oldChar` in the string with `newChar`, and returns the resulting new string.
- Save your function in a file called `lab5.py`
- Now create a file called `lab5Test.py` and write the following to test your function:
  ```
  import lab5
  a = 'It makes me happy to wake up for class at 9 a.m.'
  print a
  print lab5.replace(a, 'm', 't')
  ```
  run it and see if your code works ok

Step 2

- Write a function called `replaceNTimes(oldString, oldChar, newChar, n)` which accepts a string (`oldstring`), replaces every occurrence in the string of `oldChar` with `n` copies of `newChar`, and returns the resulting new string.
- Save this function in `lab5.py` and test if it works using a print statement in `lab5Test.py`

Step 3

- Write a function `addLists(list1, list2)` that accepts two number lists of equal length, adds them together to create a third list which is returned.
- The lists are added one element at a time: the first item in list1 is added to the first item in list2 the sum is stored in our new list (list three), then the second item in list1 is added to the second item in list2 and the sum is stored as the second element in list three … and so on, until all the list items have been added.
- When all the elements have been added, `return` the newly created list three.
- Test your function by calling it in a statement in `lab5Test.py`

Step 4 – Euclid's algorithm

Euclid's algorithm is an algorithm to determine the greatest common divisor (GCD) of two elements of any Euclidian domain (e.g., integers). Its major significance is that it does not require factoring the two integers, and it is also significant in that it is one of the oldest algorithms known, dating back to the ancient Greeks. (from wikopedia)

The pseudocode for Euclid's algorithm can be expressed as follows:

```
procedure gcd(number1, number2) {
        while number1 modulus number2 does not equal 0 do
        {
                temp ← number2
                number2 ← number1 modulus number2
                number1 ← temp
        }
        return number2
}
```

- Convert this pseudocode to the python function gcd(num1, num2).
- Once again, test your function by calling it in a statement in `lab5Test.py`

Step 5 – Martian numbers

In class we discussed converting to and from base 2 (binary). In these next few steps, we will convert to and from any given base. While we humans work in base 10 (probably because we have 10 fingers), with this new skill, we will be prepared for encounters with martians who may have a different number of fingers and therefore count using a different number base.

- Write a function `toMartian(decNum, newBase)` that converts an integer decNum from base 10, to the Martian number base newBase and returns the converted value.
- Now test your function by calling it in a statement in `lab5Test.py`

Step 6

- Now write a function `toDec(martNum, martBase)` that converts a number (`martNum`) from the Martian number base `martBase` to base 10, and returns the converted value.
- Finally, test your function by calling it in a statement in `lab5Test.py`

Step 7

Chances are, you didn't finish everything in this lab this week. Not to worry – you can work on it during the week – we will post the lab on the schedule page of the website. If you have questions, please come to instructor office hours or extra TA lab hours and we can walk through (explain) solutions.

Before you leave today – you may want to email these solutions to yourself and your partner so that you can use them to study from for future tests and assignments.