**CS116 – Lab 1**
**Due Date: September 7**

**Purpose:** Python is great for manipulating text saving you all kinds of effort in editing text changes by hand. You can use loops and array notation to make many changes with just a little bit of code.

**Knowledge:** This lab will help you become familiar with the following content knowledge:

- How loops can iterate over all the characters of a string
- How array notation can assist in accessing elements of strings and lists
- How functions can return a value rather than printing it
- How functions can use the result from other functions

**Task:** Before starting this lab, you should have read Chapter 3 in your text. Follow the steps in this lab carefully and complete the assignments.

**Assignment 1**:

You have written an essay for school, and it has to be at least five pages long. But your essay is only 4.5 pages long! You use your python skills to insert blank lines by writing a function `spaceitout(sentence,n)` which prints $n$ blank lines prior to printing the sentence.

```
>>> spaceitout("It was a dark and stormy night",3)


It was a dark and stormy night
```

**Criteria for Success:** Your code prints $n$ blank lines followed by your sentence. It should work for any positive integer $n$ and any sentence.

**Assignment 2**:
Same problem as before but now you decide to add $n$ spaces between each of the words with a function `spaceout(sentence,n)`

```
>>> spaceout("It was a dark and stormy night",3)
It   was   a   dark   and   stormy   night
```

Hint: You will want to split the sentence into a list of words as described in section 3.3 of your text.

**Criteria for Success:** Your code prints $n$ spaces between each word in the sentence. It should work for any positive integer $n$ and any sentence.

Your text in section 3.25 has an example of building a cipher code system using substitution. You are going to create a cipher function that uses one of the earliest such codes, the *Caesar Cipher*.

Suetonious tells us that Julius Caesar employed a simple substitution cipher to encode messages sent between him and his commanders. The original plaintext message is encoded using an integer as a special secret key value. The key would indicate some shift value $n$. So, the ciphertext would result from taking each letter and replacing it with another letter that is the result of shifting down the alphabet the specified number of times. Let's suppose a secret key value of 3. Thus the letter 'a' would be shifted to a 'd', the letter 'b' to a 'e'. When we get to the end of the alphabet, the shifts wrap around to the beginning. So a letter 'y' would become 'b'.

Therefore the code for the string "caesar" would be "fdhvdu".

**Assignment 3**:
You will start by writing a function `encrypt(letter,n)` that will return the substitution for that letter. (Use a `return` statement rather than a print). In order to get the wrap around you will want to use the modulo operator that is described on p61. If you take the old position of the letter in the alphabet and add $n$ and then take the modulo 26, you will get the new position in the alphabet.

To find the position of a letter in the alphabet you can use the `find` method as described on pp61-62 of your text.

```
>>> encrypt('a',3)
'd'
```

**Criteria for Success:** Your function will give the encrypted substitution for your given letter after shifting it by $n$. Your function should work for any positive integer $n$ and any of the 26 letters. Your function should not use a `print` but should use a `return` instead.

**Assignment 4**:
Now use your `encrypt` function to write the function `caesar(text,n)`.

```
>>> caesar("caesar",3)
fdhvdu
>>> caesar("xyz",2)
zab
```

**Criteria for Success:** Your function should print the encrypted caesar code for your word. It should work for any positive integer $n$ and any word composed of just letters. Make sure that you use your previously written function `encrypt` rather than duplicating that work.

Submit your file containing all your functions using the Lab 1 submission in Canvas. Please

indicate both partner names in your submission file.