CMPSC-122: Intermediate Programming
Spring 2018

Lab #4

Due Date: 02/09/2018, 11:59PM

Instructions:

- The work in this lab must be completed alone.
- If you need guidance, attend to your recitation class.
- Read the "Submitting assignments to Vocareum" file for instructions on how to submit this lab
- Do not change the function names or given code on your script
- The file name must be LAB4.py (incorrect name files will get a 0 score)
- You are responsible for testing your code. Use `python -i LAB4.py` in your terminal (or command prompt) to provide input to your functions. Test with as many data as you feel comfortable
- Remove all your testing code before uploading your file. If you are using input() to insert values in your functions and print to see the values, remove them.

Exercise 1 [3 pts]. Without using the Python interpreter, what value is printed by the following statements? Insert your answers in the function *answers()* provided in the starter code. If your answer has multiple values, separate them by commas.

a)

```
alist = []
for i in list(range(12, 26, 2)):
     alist.append(i)
print(sum(alist))
```

b)

```
for i in range(10):
    if i%2==1:
        continue
    else:
        print(i)
```

c)

```
for i in range(10):
    if i%2==1:
        break
    else:
        print(i)
```

EXAMPLES:

```
question_a= "1,2,2"
question_b= "3,5,8"
question_c= "4"
```

Exercise 2 [7 pts]. As future programmers, you will often find yourself in a situation where you may have to understand a large body of code that you did not write. That is why is so important to develop good writing coding skills. Those skills will help others to better understand your code.

Imagine you got an internship at company XYZ. On your first day, the team manager approaches you with a new task: A fellow teammate was working on an encryption/decryption program but has been sick with the flu and is unable to complete the project before its due date. Now, you are in charge of finishing the project.

This is what you are told about the project:

- Encryption code was done using the Caesar cipher method
- The key is selected by the owner of the message
- Current code only encrypts lowercase messages, which means uppercase letters do not change in the encrypted message
- The recipient of the encrypted message must have the same key to decrypt it

What to do:

- Read carefully the *encrypt*(*message*, *key*) function and understand what it does
- Modify the function *encrypt*(*message*, *key*) so it can encrypt messages that contain both uppercase and lowercase letters [3 pts]
- Write the function *decrypt*(*message*, *key*) to retrieve the original message encrypted by the encrypt function. The function takes the encrypted message (a string) and the key (an integer) that was used for encryption and returns a string with the original message. [4 pts]

Tips:

- Read https://learncryptography.com/classical-encryption/caesar-cipher to learn more about Caesar cipher
- Before writing your own code, test the current version of *encrypt*(*message*, *key*) to see what it does:

```
python -i LAB4.py
>>> encrypt("HELLO there!!!",3)
'khoor wkhuh!!!'
>>> encrypt("my name is JOHN",6)
'se tgsk oy punt'
```

- You can use this online tool https://www.xarg.org/tools/caesar-cipher/ to create test cases

EXAMPLES:

```
python -i LAB4.py
>>> encrypt("We are Penn State!!!",6)
'Ck gxk Vktt Yzgzk!!!'
>>> decrypt("Ck gxk Vktt Yzgzk!!!",6)
'We are Penn State!!!'
>>> decrypt(encrypt("We are Penn State!!!",6),6)
'We are Penn State!!!'
```

```
>>> encrypt("We are Penn State!!!",5)
'Bj fwj Ujss Xyfyj!!!'
>>> decrypt("Bj fwj Ujss Xyfyj!!!",5)
'We are Penn State!!!'
>>> decrypt(encrypt("We are Penn State!!!",5),5)
'We are Penn State!!!'
```