CSCI 1411: Fundamentals of Computing
Lab 6
Due Date: 8:30 AM September 29, 2020

Name: Kerry Gip

Goals:

- Understanding the string datatype and how strings are represented in the computer
- Becoming familiar with various operations that can be performed on string using built-in functions and string methods.

Development Environment: IDLE

Deliverables:

1. This lab handout with 4 screen shots (2 for part I and 2 for part II).
2. Your Python code for Part II of this lab. Name the file using the following format: yourLastnameFirstnameLab06b.cpp
   Example: If your name is Jamal Jones then you will name the file as follows: JonesJamalLab06b.cpp

How to take a **screen shot**:

- For a Windows 10: Use Snipping Tool to copy and CTRL + V to paste screen shot.
- For Mac: Shift + Command + 4 to copy and CTRL + V to paste screen shot.

Part I – Skill Practice (10 pts)

- Start IDLE
- Create a new file.
- Type the following code in the file. …. **Do not cut and paste.** You will learn more by typing it in.
- Remember to update the first 3 lines with your own first name, last name and the date of the lab.

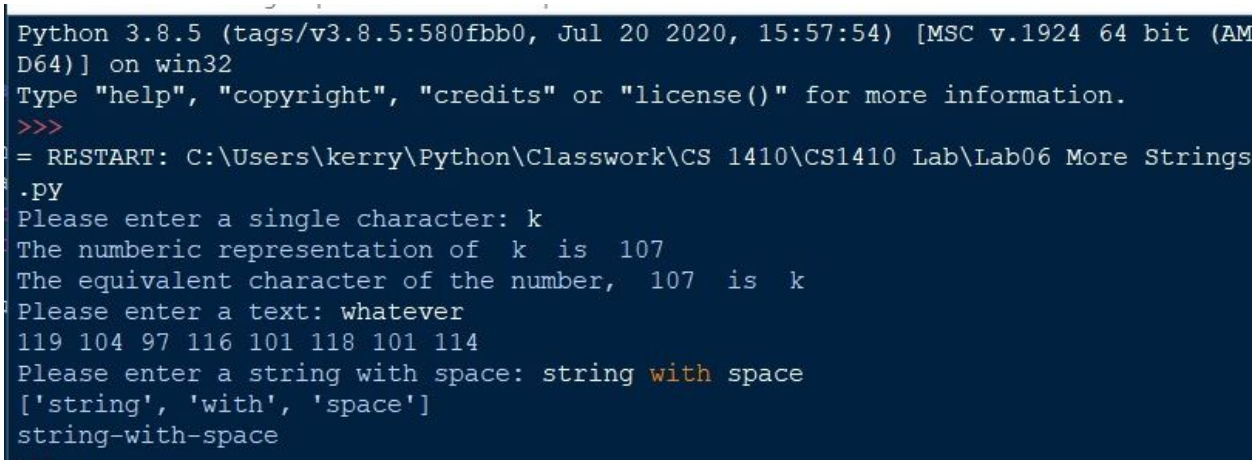
```
# Your first Name
# Your Last Name
# Date: The date
# Description: Lab 6. This lab demonstrates various functions of String datatype


def main():

    # Python translates each character to a number. The ord() function in python
    # accepts a single character string as an argument and returns the numeric code of it.

    ch = input("Please enter a single character: ")
    value = ord(ch)
    print("The numeric representation of", ch, "is", value)

    # The chr() function does the reverse. chr() function takes a integer number
    # as an argument and returns the equivalent character.

    print("The equivalent character of the number", value, "is", chr(value))

    # The following piece of code will take a string as input and it will convert
    # each of the character as a sequence of numbers (separated by whitespace)

    txt = input("Please enter a text: ")
    for ch in txt:
        print(ord(ch), end=" ")


    #split function splits a string into a list of substrings based on a delimiter.

    txt1 = input("\nPlease enter a string with space: ")
    listSubStrings = txt1.split(" ")
    print(listSubStrings)

    #join function joins the list of strings into one string using a delimiter

    newString = "-".join(listSubStrings);
    print(newString)
```

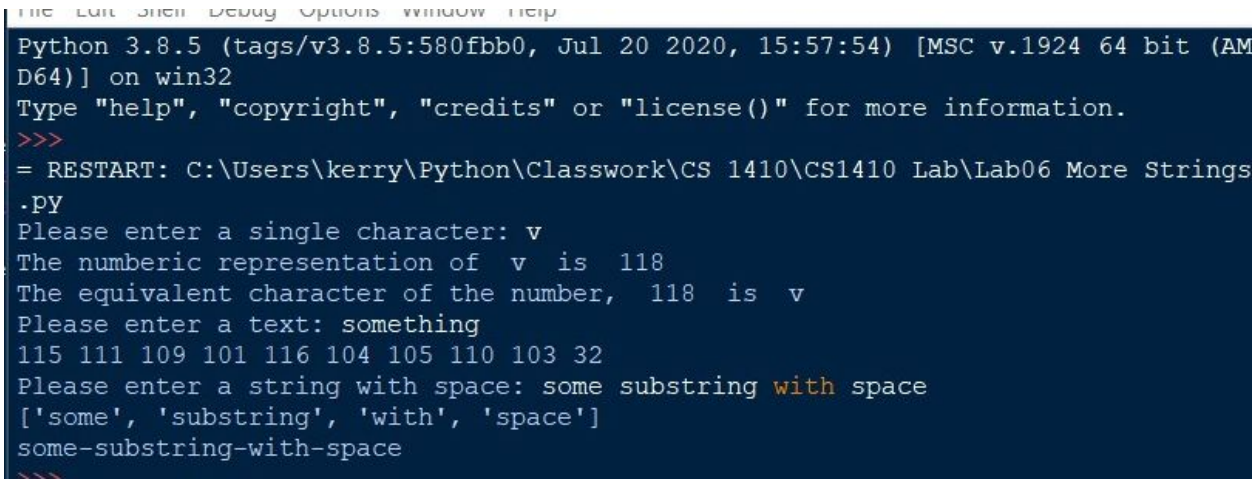
- Save the file as “YourLastNameYourFirstNameLab06a.py”
- Click Run -> Run Module
- If you get any syntax error, try to correct the syntax error.
- If no syntax error, this will redirect you to the output screen.
- Type main()

- Output will look like the following:

```
>>> main()
Please enter a single character: c
The numeric representation of c is 99
The equivalent character of the number 99 is c
Please enter a text: Mango
77 97 110 103 111
Please enter a string with space: I love python programming
['I', 'love', 'python', 'programming']
I-love-python-programming
```

- **Please capture 2 screenshots and paste it here.**





Part II – Find the *Caesar cipher* (15 pts)

- A Caesar cipher is a simple substitution cipher based on the idea of shifting each letter of the plaintext message a fixed number (called the key) of the positions in the alphabet. For example, if the key value is 2, the word “Sourpass” would be encoded as “Uqwtrcuu”. The original message can be recovered by decoding the message using the negative of the key. For example, the encoded word “Uqwtrcuu” can be decoded by encoding the word with key -2.
- In this part, write a python program that can encode and decode Caesar cipher. The input to the program will be a string of plaintext and the value of the key. The output will be an encoded message where each character in the original message is replaced by shifting it key characters in the Unicode character set. For example, if *ch* is a character in the string and key is the amount to shift, then the character to replaces *ch* can be calculated as *chr(ord(ch) + key)*.
- Create a python program and save it as “lastnameFirstnameLab06b.py”
- The program does the following:
    - Ask user for a string input
    - Ask user for a key input (integer)
    - Your program will create the encoded message using the string input and save the encoded message as a variable. Output the encoded message.
    - Your program will recover the original message by decoding the encoded message. Output the original message.

- Sample input/Output:

```
Please enter the string: Sourpass
Please enter the key: 2
The Encoded message is
Uqwtrcuu
The Decoded message is:
Sourpass
>>>
```

- **Capture 2 screenshots with two different inputs and paste them here**
- **Submit the code on Canvas.**

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/kerry/Desktop/GipKerryLab06b.py ==========
This is a sample Caesar Cipher code
Please enter a string without spaces: whatthe
Your string that will be encoded in lower case is  whatthe
Please enter a key(number) to shift by: 2
The encoded message is:   yjcvvjg
The decoded message is:  whatthe
>>>
========== RESTART: C:/Users/kerry/Desktop/GipKerryLab06b.py ==========
This is a sample Caesar Cipher code
Please enter a string without spaces: zanzbar
Your string that will be encoded in lower case is  zanzbar
Please enter a key(number) to shift by: 3
The encoded message is:   cdqcedu
The decoded message is:  zanzbar
>>>
========== RESTART: C:/Users/kerry/Desktop/GipKerryLab06b.py ==========
```