CS-GY6803 - ISSEM - Lab 1

For labs in this class, we will be using Jupyter notebooks which is a useful application for writing and compiling Python code. It contains all the necessary libraries and packages which can be used for labs and projects for this class.

Setup

- Install the Jupyter notebook. You can install in one of the following ways:
  • Refer to link: https://jupyter.org/install
  • Alternatively, you can install Anaconda using below link in your system. Based on the OS you have, please select appropriate version.
  https://www.anaconda.com/products/individual-d
- Post installation, register into the anaconda to access Jupyter notebook.
- After installing and registering on Anaconda, create a new Python 3 notebook.

Initialize

- In the first cell, put the heading as Lab1 using Heading option in cell type.
- Also, in the first cell, change the cell type to Markdown and write the name of all group members and their Net IDs under the heading.
- Create a new cell (type- code) and write the following code snippet and run the cell, adding in your group number.

```
course = "ISSEM"
lab = "Lab1"
group = "#TODO"
print("Group: " + group, "\tLab: " + lab, "\tCourse: " + course)
```

Understanding the Pretest

This is the question from the Python assessment. What operation following piece of code is doing? Use comments to explain the relevant parts of the code.

```
s = "The quick brown fox jumps over the lazy dog"
arr = s.split(" ")
temp = []
temp.append(arr[0])
temp.append(arr[1])
arr[0] = arr[len(arr)-2]
arr[1] = arr[len(arr)-1]
arr[len(arr)-2] = temp[0]
arr[len(arr)-1] = temp[1]
s = " ".join(arr)
```

Write the following function in the next cell.

```
def dummy(var):
    x = str(var)
    y = len(x) - 1
    z = ""
    while y >= 0:
        z = z + x[y]
        y -= 1
    return int(z)
```

Using an example, explain what this function is doing in comments.

In the next cell, use the above dummy function to check if below numbers are palindromes or not.

- 121
- 123
- 414
- 866
- 988
- 101

Discounts

The following is the code for calculate_discount function which you implemented in the Python Assessment.

```
def calculate_discount(price):
    if price >= 500:
        final_price = price * (70/100)
    elif price >= 350 and price < 500:
        final_price = price * (80/100)
    elif price >= 150 and price < 350:
        final_price= price * (90/100)
    elif price >= 10 and price < 150:
        final_price = price * (95/100)
    elif price > 0 and price < 10:
        final_price = price * (50/100)
    else:
        final_price = "Price cannot be negative. Please enter a positive value."
    return final_price
```

Write this code in next cell and calculate the discount for the following prices
(Hint: Use for loop):
• 440
• 265
• 144
• -10
• 8

Basic Cybersecurity

Please explain the following terms.

- Cryptography

- Encryption
- Plaintext
- Ciphertext
- Decryption
- Double Strength Encryption
- Hybrid Encryption

Shift Ciphers

If you have a message you want to transmit securely, you can encrypt it (translate it into a secret code). One of the simplest ways to do this is with a shift cipher. Famously, Julius Caesar used this type of cipher when sending messages to his military commanders. He would take a number and shift the letters of his message by a certain number of places using that number.

We'll call this number the encryption key. It is just the length of the shift we are using. For example, upon encrypting the message COOKIE using a shift cipher with encryption key 3, we obtain the encoded message (or ciphertext): FRRNLH. Notice that we are only dealing with capital letters.

Let's explain the shift cipher for this encryption key:

- Using a table of all letters, with A = 0, B = 1 , etc., we can represent the letters in our message "COOKIE" with their corresponding numbers: 2 14 14 10 8 4.
- Now add 3 (the encryption key) to each number to get: 5 17 17 13 11 7
- Now use the table to replace these numbers with their corresponding letters: FRRNLH

Now, let's try doing the same thing for the word "ZERO".

- What letter would we use to replace "Z" when we encrypt?
    - We use the letter "C", which can be viewed as being 3 places further along than "Z" if, after we reach "z", we cycle the alphabet around to the beginning again.
      Let's explain the shift cipher for this encryption key 3:
- After performing shift cipher encryption key 3, the message "ZERO" becomes CHUR.

In terms of mathematical representation of our letters, the encryption of the message "ZERO" looks like,

- 25 4 17 14 → 28 7 20 17

because if we define the following notation for integers a and b and integer m > 1:

- a ≡ b (mod m) means m is a divisor of a – b.

In summary, our encryption of the message "ZERO" using a shift cipher with encryption key 3 looks like this

- Z → 25 → 25 + 3 ≡ 28 (mod 26) → C
- E → 4 → 4 + 3 ≡ 7 (mod 26) → H
- R → 17 → 17 + 3 ≡ 20 (mod 26) → U
- O → 14 → 14 + 3 ≡ 17 (mod 26) → R

How is the original (plaintext) message recovered from the ciphertext if the encryption key is known? Shift the cipher the opposite way with the same key.

The following ciphertext was produced with encryption key 3: CHUR
To decrypt it (i.e., to recover the plaintext message), we need to subtract 3 (. . . or add 23 . . . think about why is that the same) to each of the numbers representing the ciphertext letters.

- C → 2 → 2 - 3 ≡ -1 (mod 26) → Z
- H → 7 → 7 - 3 ≡ 4 (mod 26) → E
- U → 20 → 20 - 3 ≡ 17 (mod 26) → R
- R → 17 → 17 - 3 ≡ 14 (mod 26) → O

Implement the above shift cipher, with encryption key 3 in Python. You are expected to write two functions:

- Encryption: An encryption function that takes the plaintext as input and returns the ciphertext as output. A good way to check if your function works correctly is to make sure that the output text of your function after encryption is NOT the same as your input text and manually check if each letter on the input has been replaced by 3 letters to the right.
- Decryption: A decryption function will take the output of the above encryption function and decode the message to give out the original plaintext.
  Note: You are expected to handle only alphabetic inputs for both your encryption and decryption functions.

Below is a sample set of inputs that your submission will be tested against.
10 more random inputs on top of these will be used to make sure the functionality works for any alphabetic inputs.

- USA
- ISSEM
- CHICAGO
- CAPPUCCINO
- ZUPPA
- XAVIERS
- YELLOW
- SECURITY
- CYBERHUB
- HURRICANE

Hashing

- What is hash function in cryptography?
- Write a hash function in Python to create the hash value of below message:
  - message = "Information Systems Security Engineering and Management"
  - To hash the above function use the Python library hashlib (SHA256)

Hash verification:

Data can be compared to a hash value to confirm its integrity.

The data is hashed at a certain time and the hash value is protected in some way (for this lab, you can note it down). Later, the data can be hashed again and compared to the protected value. If the hash values match, the data has not been altered. If the values do not match, the data has been corrupted.

Write a Python function which implements hash_verification(data, original_hash) as explained above. Use that function to verify the data integrity of the message in the beginning of this section.

Submission

This is a group submission. Please submit only one python notebook per group per part of the lab. Make sure to run all the functions and then download the notebook in a .ipynb format.

Title of the notebook should be lab1_group<group_number>.ipynb. Good luck!