**Name: Session:**

**Programming I**

**Lab Exercise 11/27/2023**

Encryption is a mathematical operation. In order to encrypt non-numeric data we must get it into numeric format. There are several built-in Python functions that will aid you with this. These are ord() and chr(). ord() will turn a character into its ASCII equivalent (i.e. ord(‘A’) will return 65). chr() will return the character equivalent of an integer (i.e. chr(65) will return ‘A’). There are also a few string methods that will be useful. These are isalpha(), isupper(), and islower(). See 5.6.1 in the Python docs.

Let’s write a few short practice programs to practice this:

1. Write a program that will print all of the upper case letters.
2. Write a program that will print out digits.
3. Write a program that will print the characters and their ASCII equivalent (0 – 255).
4. Write a program that will print the ASCII representation of the string “Hour of Code”.

The goal of this exercise is to write a cyclic cipher to encrypt messages. This type of cipher was used by Julius Caesar to communicate with his generals. It is very simple to generate but it can actually be easily broken and does not provide the security one would hope for.

The key idea behind the Caesar cipher is to replace each letter by a letter some fixed number of positions down the alphabet. For example, if we want to create a cipher shifting by 3, you will get the following mapping:

Plain: ABCDEFGHIJKLMNOPQRSTUWXYZ

Cipher: DEFGHIJKLMNOPQRSTUVWXYZABC

To be able to generate the cipher above, we need to understand a little bit about how text is represented inside the computer. Each character has a numerical value and one of the standard encodings is ASCII (American Standard Code for Information Interchange). It is a mapping between the numerical value and the character graphic. For example, the ASCII value of ’A’ is 65 and the ASCII value of ’a’ is 97. To convert between the ASCII code and the character value in Python, you can use the following code:

letter = ’a’

# converts a letter to ascii code ascii\_code = ord(letter)

# converts ascii code to a letter letter\_res = chr(ascii\_code)

print (ascii\_code, letter\_res )

Start small. Do not try to implement the entire program at once. Break the program into parts as follows:

1. Create a file called cipherTest.py. Start your program by asking the user for a phrase to encode and the shift value. Then begin the structure of your program by entering in this loop (we’ll build on it more in a bit):

phrase = “Hello World”

encoded\_phrase = “”

for c in phrase:

encoded\_phrase = encoded\_phrase + c

What does this loop do? Make sure you understand what the code does before moving on!

1. Now modify your code (save as pinCipher.py), so that it produces the encoded string using the cyclic cipher with the shift value entered by the user. We will start by encoding a 6 digit pin number. Let’s see how one might do a cyclic shift. Let’s say we have the sequence:

012345

If we use a shift value of 4 and just shift all the numbers, the result will be:

456789

If we only want the values of the numbers to remain between 0 and 5. To do this we will use the modulus operator. The expression x%y will return a number in the range 0 to y-1 inclusive, e.g. 4%6 = 4, 6%6 = 0, 7%6 =1. Thus the result of the operation will be:

450123

3. Now let’s modify (save as cipher.py) the above program so that it encrypts any string. In this case we will only encrypt alphabetic characters.

Hint: Note that the ASCII value of ’A’ is 65 and ’a’ is 97, not 0. So you will have to think how to use the modulus operator to achieve the desired result. Apply the cipher separately to the upper and lower case letters.

Here is what you program should output:

Enter sentence to encrypt: Mayday! Mayday!

Enter shift value: 4

The encoded phrase is: Qechec! Qechec!

Now let’s test our program by filling out the following form:

**Pin Cipher Results “12345”**

|  |  |
| --- | --- |
| Shift | Result |
| 1 |  |
| 6 |  |
| 23 |  |
| -1 |  |

**Cipher Results for “Hello World”**

|  |  |
| --- | --- |
| Shift | Result |
| 1 |  |
| 6 |  |
| 23 |  |
| -1 |  |

4. Submit your source code for problems 1 to 4 as well as pinCipher.py and cipher.py as well as this sheet.