Cryptography and Network Security Lab

Assignment 1
Implementation and Understanding of Caesar Cipher

2019BTECS00058
Devang K

Batch: B2

Title: Implementation and Understanding of Caesar Cipher

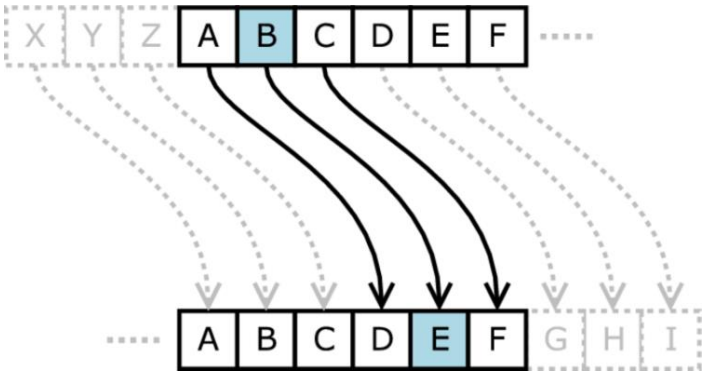
Aim: To Study, Implement and Demonstrate the Caesar Cipher Algorithm

Theory:

Caesar Cipher is named after ‘Julius Caesar’ who used to use this cipher algorithm in his private correspondence. It is a form of Substitution Cipher wherein we replace each letter in the plaintext by a letter some fixed number of positions ahead/behind. Let us called these number of positions as ‘shift key’.

Illustration:

Say we wish to encrypt ‘ABC’ with a shift key of 4. Then, the cipher would work as:



Taking some examples:
Say ‘DEVANG’ with a shift-key 3
Our logic would be like:

For every character:
ASCII of D = 68

To make a general case, we subtract from 65 – ASCII of A.
We get, 68-65 = 3 (0-based index)

We add 3 (shift-key) to it, 3+3 = 3

Then, we take a mod with 26 (all the alphabets, it’s a circular repetition)

6 % 26 = 6

Then add, 65 again and take ASCII
We get 71, which converting to Character is – G.

Similar to this, we go on doing for each character. Thus, the Ciphertext for ‘DEVANG’ would be ‘GHYDQJ’.

To decrypt this, we run the algorithm in reverse order:

So, for every character, we know the shift-key is 3
Say ‘G’, ASCII – 71

We subtract 65 and also the shift-key value:
71 – 65 – 3 = 3

Then taking a mod with 26 for general case:
3 % 26 = 3

We then add 65 and convert it to character.
3 + 65 = 68
which is ‘D’

Similarly, repeating for every character, we finally get – ‘DEVANG’.

Thus, we can encrypt and decrypt in Caesar Cipher.
We shall look at more examples in the implementation.

Code:

The main crux of the code is the function that handles this shift-key for each character and returns its shifted value.

```
def shiftCharWithKeyDecrypt(c, step):
    amt = ord(c)
    amt -= 65
    amt -= step
    amt %= 26
    amt += 65
    return chr(amt)
```

This function can we used for both, encryption and decryption, just by reversing the sign of key in decryption.
Beyond this, we write the code to take input and traverse character-by-character and encrypt and decrypt.

Code for Encryption:

```
def shiftCharWithKey(c, step):
    amt = ord(c)
    amt -= 65
    amt += step
    amt %= 26
    amt += 65
    return chr(amt)

def chooseTypeOfInput():
    print("Choose your method:")
    print("1. Encrypt a File.")
    print("2. Encrypt an Input.")
    print("Please Enter Type of Input: ", end='')
```

```
    n = int(input())
    if n == 1 or n == 2:
        return n
    print("Incorrect choice. Try Again!")
    chooseTypeOfInput()

choice = chooseTypeOfInput()

value=0
shift = 0

if choice == 1:
    f = open("enc.txt", "r")
    value = f.readline()

else:
    print("Enter the Plaintext: ", end='')
    value = input()

print("Enter the Shift Key Value: ", end='')
shift = int(input())

encryptedValue = ""

for c in value:
    encryptedValue += shiftCharWithKey(c, shift)

print("\nEncryption Result:")
print("PlainText: "+value)
print("Ciphertext: "+encryptedValue)
```

Code for Decryption:

```
def shiftCharWithKeyDecrypt(c, step):
    amt = ord(c)
    amt -= 65
    amt -= step
    amt %= 26
    amt += 65
    return chr(amt)

def chooseTypeOfInput():
    print("Choose your method:")
    print("1. Decrypt a File.")
    print("2. Decrypt an Input.")
    print("Please Enter Type of Input: ", end='')
    n = int(input())
    if n == 1 or n == 2:
```

```
        return n
    print("Incorrect choice. Try Again!")
    chooseTypeOfInput()

choice = chooseTypeOfInput()

ciphertext=0
shift = 0

if choice == 1:
    f = open("dec.txt", "r")
    ciphertext = f.readline()

else:
    print("Enter the Plaintext: ", end='')
    ciphertext = input()

print("Enter the Shift Key Value: ", end='')
shift = int(input())

plaintext = ""

for i in ciphertext:
    plaintext += shiftCharWithKeyDecrypt(i, shift)

print("Ciphertext: "+ciphertext)
print("Decrypted Plaintext: "+plaintext)
```

We now solve some examples with the code.
Say we wish to encrypt: ‘CEASEFIRETILLDAWN’

We choose the option to directly type the string.
And let’s take shift-key of 12.

```
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\CaesarCipher> py .\encryption.py
Choose your method:
1. Encrypt a File.
2. Encrypt an Input.
Please Enter Type of Input: 2
Enter the Plaintext:
```

We enter the plaintext and shift-key

```
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\CaesarCipher> py .\encryption.py
Choose your method:
1. Encrypt a File.
2. Encrypt an Input.
Please Enter Type of Input: 2
Enter the Plaintext: CEASEFIRETILLDAWN
Enter the Shift Key Value: 12

Encryption Result:
PlainText: CEASEFIRETILLDAWN
Ciphertext: OQMEQRUDQFUXXPMIZ
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\CaesarCipher>
```

We get that encryption of ‘CEASEFIRETILLDAWN’ with shift-key of 12 is: ‘OQMEQRUDQFUXXPMIZ’.

Now, let’s decrypt it using our decryption code.

```
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\CaesarCipher> py .\decryption.py
Choose your method:
1. Encrypt a File.
2. Encrypt an Input.
Please Enter Type of Input: 2
```

We choose 2 and then enter the Ciphertext and shift-key.

```
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\CaesarCipher> py .\decryption.py
Choose your method:
1. Encrypt a File.
2. Encrypt an Input.
Please Enter Type of Input: 2
Enter the Plaintext: OQMEQRUDQFUXXPMIZ
Enter the Shift Key Value: 12
Ciphertext: OQMEQRUDQFUXXPMIZ
Decrypted Plaintext: CEASEFIRETILLDAWN
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\CaesarCipher>
```

Thus, we get the result, ‘CEASEFIRETILLDAWN’.

Taking another example, we can encrypt and decrypt the content of a file.
We take enc.txt and dec.txt

Putting the example ‘ATTACKWITHALLWEHAVE’ with a shift-key of -10.

We now run our code

```
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\CaesarCipher> py .\encryption.py
ption.py
Choose your method:
1. Encrypt a File.
2. Encrypt an Input.
Please Enter Type of Input: 1
Enter the Shift Key Value: -10

Encryption Result:
PlainText: ATTACKWITHALLWEHAVE
Ciphertext: QJJQSAMYJXQBBMUXQLU
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\CaesarCipher>
```

Ciphertext: ‘QJJQSAMYJXQBBMUXQLU’

Then, we decrypt.

```
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\CaesarCipher> py .\decryption.py
Choose your method:
1. Decrypt a File.
2. Decrypt an Input.
Please Enter Type of Input: 1
Enter the Shift Key Value: -10
Ciphertext: QJJQSAMYJXQBBMUXQLU
Decrypted Plaintext: ATTACKWITHALLWEHAVE
PS C:\Users\marcus\Desktop\College\CNS-Lab-Archives\CaesarCipher>
```

Thus, we demonstrated the working of the code with examples.

Conclusion:

Thus, the Caesar Cipher algorithm was studied and demonstrated with the code. It is observed that Caesar Cipher is really weak as the key can be any of 26 values and therefore is easy to crack in modern cryptography.