



Department of Computer Science,
CUI, Attock Campus.

Program: BSSE

Assignment # 01

Information Security

Student Name	Ahsan Rehman Khan Niazi
Registration #	FA24-BSE-056

Professor: Ms. Ambreen Gul

Session: Fall 2024

1.Introduction

In this assignment, I implemented the Caesar Cipher using Python.

Caesar Cipher is a simple Encryption technique in which each letter of a message is shifted by a fixed number of positions in the alphabet.

For Example, If the shift is 3:

A → D

B → E

Hello → Khoor

This technique is easy to understand and learning about encryption concept.

2.Code Explanation

I will explain my code step by step.

- Caesar_encrypt:

```
# Function to encrypt text using Caesar Cipher
def caesar_encrypt(text,shift):
    # empty string to store result
    result1 = ""

    for char in text:

        # Check if character is uppercase letter
        if char.isupper():
            result1 += chr((ord(char) - 65 + shift) % 26 + 65)
        # Check if character is lowercase letter
        elif char.islower():
            result1 += chr((ord(char) - 97 + shift)% 26 + 97)
        else:
            result1 += char

    return result1
```

def caesar_encrypt(text,shift):

This line define a function named **caesar_encrypt**.

It take 2 input(text,shift):

Text→Original information

Shift →the number of position to shift letters

result1 = ""

IT will store the Encrypted Text.

for char in text:

This loop goes through each character in the input text 1 by 1.

result1 += chr((ord(char) - 65 + shift) % 26 + 65):

- **Ord(char)** convert the letters into ASCII number
- 65 is subtracted because ASCII value of capital A is 65.
- **% 26** is used to wrap around the alphabet if it goes past z.
- **Chr()** converts the numbers back into a letters.
- The new letters added into **result1**.

result1 += chr((ord(char) - 97 + shift) % 26 + 97):

This work is same as Uppercase , but:

- ASCII value of small “a” is 97
- So 97 is used instead of 65

result1 += char:

If the character is not a letter(like space, number, symbol)

It is added as it without any change.

return result1:

Finally , the encrypted text is returned from the function.

■ **caesar_decrypt:**

```
# Function to decrypt text using Caesar Cipher
def caesar_decrypt(Text, shift):
    result2 = ""

    for char in Text:
        if char.isupper():
            result2 += chr((ord(char) - 65 - shift )% 26 + 65)
        elif char.islower():
            result2 += chr((ord(char) - 97 - shift)% 26 + 97)
        else:
            result2 += char

    return result2
```

if char.isupper():

result2 += chr((ord(char) - 65 - shift)% 26 + 65):

Here the shift is subtracted instead of added.

This move the letter backward to get the original letter.

elif char.islower():

result2 += chr((ord(char) - 97 - shift)% 26 + 97)

Same logic is used for lowercase letters.

else:

result2 += char

non – alphabet characters remain unchanged.

return result2:

The decrypted text is returned

■ Main Program:

```

# Taking input from user
info = input("Enter your information: ")
sh_value = int(input("Enter Shift Value: "))

# Encrypting Information
encrypted = caesar_encrypt(info,sh_value)
print("Encrypted Info: ",encrypted)

# Decrypting Information
decrypted = caesar_decrypt(encrypted,sh_value)
print("Decrypted Info: ",decrypted)

```

`encrypted = caesar_encrypt(info,sh_value):`

Encryption function is called and the result will be stored.

`decrypted = caesar_decrypt(encrypted,sh_value):`

Decryption function is called by using same shift value

3.Program Output:

```

Enter your information: Ahsan Rehman Khan Niazi
Enter Shift Value: 5
Encrypted Info: Fmxfs Wjmrfs Pmfs Snfen
Decrypted Info: Ahsan Rehman Khan Niazi
PS E:\Application\vs-code project\Artificial Intellegience>

```

4.Security Analysis:

Caesar Cipher is a very simple Encryption techniques;

Strength:

- Easy to understand and implement
- Good for learning basic encryption concepts

Weakness:

- It has only 25 possible shift values
- It can easily be broken by brute force attack
- It is not secure for modern communication
- In brute force attack , attacker tries all possibles shift values and finds the correct information
- So, In that case → Caesar Cipher is not secure for real-world application.

5.Conclusion:

In this assignment, I successfully implemented Caesar Cipher in python.

The program Correctly handles:

- Uppercase letters
- Lowercase letters
- Spaces
- Special characters

This assignment helped me to understand about encryption and decryption concepts in information Security.