

Prodigy Cyber Security Internship --Task-01

Intern: Abhinandan Kumar

Task: Caesar Cipher

Language: Python

Date: 04/11/2025

Objective

To implement Caesar Cipher encryption and decryption using Python.

Description

Caesar Cipher shifts characters by a fixed number to encrypt and decrypt text.

Algorithm

1. Take input text
2. Take shift value
3. Convert each character using shift
4. Show encrypted and decrypted text

Code

```
def caesar_cipher(text, shift, mode):  
    result = ""  
  
    for char in text:  
  
        if char.isalpha():  
  
            ascii_offset = 65 if char.isupper() else 97  
  
            if mode == "encrypt":  
  
                result += chr((ord(char) - ascii_offset + shift) % 26 + ascii_offset)  
  
            elif mode == "decrypt":  
  
                result += chr((ord(char) - ascii_offset - shift) % 26 + ascii_offset)  
  
        else:  
  
            result += char  
  
    return result  
  
  
message = input("Enter your message: ")
```

```

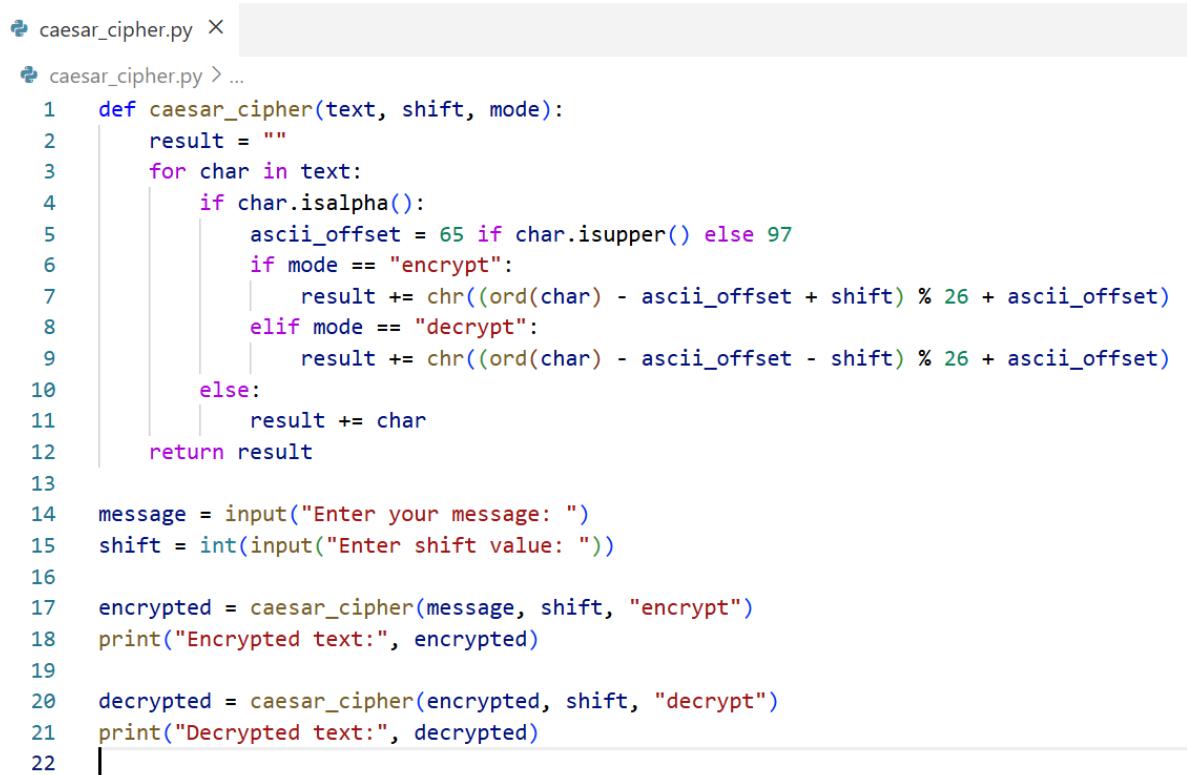
shift = int(input("Enter shift value: "))

encrypted = caesar_cipher(message, shift, "encrypt")
print("Encrypted text:", encrypted)

decrypted = caesar_cipher(encrypted, shift, "decrypt")
print("Decrypted text:", decrypted)

```

Output Screenshots



The screenshot shows a code editor window with a tab labeled 'caesar_cipher.py'. The code is a Caesar cipher implementation. It defines a function 'caesar_cipher' that takes 'text', 'shift', and 'mode' as parameters. The function iterates over each character in the text. If the character is alphabetic, it calculates its ASCII offset (either 65 for uppercase or 97 for lowercase) and applies the shift. For uppercase letters, the formula is $\text{result} += \text{chr}((\text{ord}(\text{char}) - \text{ascii_offset} + \text{shift}) \% 26 + \text{ascii_offset})$. For lowercase letters, it's $\text{result} += \text{chr}((\text{ord}(\text{char}) - \text{ascii_offset} - \text{shift}) \% 26 + \text{ascii_offset})$. If the character is not alphabetic, it is added to the result as is. The script then prompts the user for a message and shift value, performs the encryption, prints the encrypted text, performs the decryption, and prints the decrypted text.

```

caesar_cipher.py > ...
def caesar_cipher(text, shift, mode):
    result = ""
    for char in text:
        if char.isalpha():
            ascii_offset = 65 if char.isupper() else 97
            if mode == "encrypt":
                result += chr((ord(char) - ascii_offset + shift) % 26 + ascii_offset)
            elif mode == "decrypt":
                result += chr((ord(char) - ascii_offset - shift) % 26 + ascii_offset)
        else:
            result += char
    return result
message = input("Enter your message: ")
shift = int(input("Enter shift value: "))
encrypted = caesar_cipher(message, shift, "encrypt")
print("Encrypted text:", encrypted)
decrypted = caesar_cipher(encrypted, shift, "decrypt")
print("Decrypted text:", decrypted)

```

PROBLEMS DEBUG CONSOLE TERMINAL PORTS OUTPUT powershell + ▾ ...

```
PS C:\Users\Abhinandan\Desktop\Prodigy_Task01> python caesar_cipher.py
>>
Enter your message: HELLO WORLD
Enter shift value: 3
Encrypted text: KHOOR ZRUOG
Decrypted text: HELLO WORLD
PS C:\Users\Abhinandan\Desktop\Prodigy_Task01>
```

Conclusion

Successfully implemented Caesar Cipher in Python and verified output.