Task – 01: Implement Caesar Cipher

1. Introduction

The Caesar Cipher is one of the oldest and simplest encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is shifted by a fixed number of positions in the alphabet. This technique was named after Julius Caesar, who reportedly used it for his private correspondence.

2. Objective

To create a Python program that can encrypt and decrypt text using the Caesar Cipher algorithm.

The program should:

- Allow users to input a message.
- Allow users to input a shift value (key).
- Perform encryption or decryption based on user choice.

3. Algorithm

Step 1: Define the ALPHABET String containing all supported characters (A-Z, a-z, numbers, and symbols).

Step 2: Input:

- Get mode (Encrypt or Decrypt).
- Get message string.
- Get key (Shift integer).

Step 3: processing (Loop through message):

For each character in the message:

- **Find Position:** Get the current index of the character in ALPHABET.
- **Calculate Shift:**
  - If Encrypting: new_index = (index + key) % length(ALPHABET)
  - If Decrypting: new_index = (index - key) % length(ALPHABET)
- **Append:** Add the character at new_index to the result.
- **Handle Unknowns:** If a character is not in ALPHABET, keep it unchanged.

Step 4: Output:

- Print the final result string.

4. Python Implementation (Code)

```
ALPHABET = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789 !@#$%^&*()'


def caesar_cipher(message, key, mode):
    result = ""
    for char in message:
        if char in ALPHABET:
            current_index = ALPHABET.find(char)
            if mode == 'encrypt':
                new_index = (current_index + key) % len(ALPHABET)
            else: # decrypt
                new_index = (current_index - key) % len(ALPHABET)
            result += ALPHABET[new_index]
        else:
            result += char
    return result


def get_valid_mode():
    """Forces user to enter 'encrypt' or 'decrypt'."""
    while True:
        user_input = input("Select Mode (encrypt/decrypt): ").lower().strip()
        if user_input in ['encrypt', 'e']:
            return 'encrypt'
        elif user_input in ['decrypt', 'd']:
            return 'decrypt'
        else:
            print("❌ Invalid mode. Please type 'encrypt' or 'decrypt'.")

def get_valid_key():
    """Forces user to enter a valid integer number."""
    while True:
        user_input = input("Enter shift key (number): ").strip()
        try:
            val = int(user_input)
            return val
        except ValueError:
            print("❌ Invalid key. Please enter a whole number (e.g., 3, 5, 10).")

def get_valid_message():
    """Forces user to enter a non-empty message."""
    while True:
        msg = input("Enter the message: ").strip()
        if len(msg) > 0:
            return msg
        else:
            print("❌ Message cannot be empty. Please type something.")
```

```
def main():
    print("--- Caesar Cipher  ---")
    mode = get_valid_mode()
    message = get_valid_message()
    key = get_valid_key()
    output = caesar_cipher(message, key, mode)

    print("\n" + "="*30)
    print(f"✅ OPERATION SUCCESSFUL")
    print(f"Mode:   {mode.upper()}")
    print(f"Result: {output}")
    print("="*30)

if __name__ == "__main__":
    main()
```

5. Example Run

Encrypt:

```
--- Caesar Cipher  ---
Select Mode (encrypt/decrypt): e
Enter the message: Hello World!
Enter shift key (number): 3
==============================================
✅ OPERATION SUCCESSFUL
Mode:   ENCRYPT
Result: Khoor#Zruog$
===============================================
```

Decrypt:

```
***Caesar Cipher ***

Do you want to encrypt or decrypt?  d

Enter the message to encrypt: Khoor#Zruog$

Enter the shift value (number):3

========================================

☑ OPERATION SUCCESSFUL

Mode:   DECRYPT

Result: Hello World!
```

6. Conclusion

- The Caesar Cipher is a simple technique to teach the basics of encryption and decryption. In this project, the Caesar Cipher algorithm has been successfully implemented using Python, which allows users to enter their message and key to perform encryption/decryption based on their choice.
- The project meets all the requirements of Task 01, providing an understanding of the basic principles of data security and information protection