COMSATS University Islamabad

Attock Campus

Department Of Computer Science

Course:	*Information Security*
Instructor:	*Ms. Ambreen Gul*
L-Assignment No:	*01*
Due Date:	*28th February, 2026*

Student Details

Registration No:	*FA24-BSE-030*
Name:	*Sayeda Aliza Hashmi*

Table of Contents

Detailed Code

```
"""
SIMPLE CAESAR CIPHER
This program shifts letters to encrypt messages
Example: "HELLO" with shift 3 becomes "KHOOR"
"""
def caesar_encrypt (text, shift):
    """
    STEP BY STEP:
    1. Take each letter from text
    2. If it's a letter, shift it
    3. If not a letter, keep it as is
    4. Return the new text
    """
    new_text = ""

    # Look at each character one by one
    for character in text:
        # Check if it's a letter
        if character.isalpha():
            # Handle uppercase letters (A-Z)
            if character.isupper():
                # Convert letter to number (A=0, B=1, ... Z=25)
                # ord('A') = 65, so subtract 65 to get 0-25
                old_position = ord(character) - 65

                # Add shift to get new position
                new_position = (old_position + shift) % 26
                # Convert back to letter (add 65 to get ASCII code)
                new_character = chr(new_position + 65)
            # Handle lowercase letters (a-z)
            else:
                # Similar process but with 'a' = 97
                old_position = ord(character) - 97
                new_position = (old_position + shift) % 26
                new_character = chr(new_position + 97)

            # Add the new letter to our result
            new_text = new_text + new_character
        else:
            # If it's not a letter (space, comma, etc.), keep it
            new_text = new_text + character
    # Return the encrypted text
    return new_text

def caesar_decrypt (ciphertext, shift):
    """
    To decrypt, just shift backwards
    (use negative shift)
    """
    return encrypt(text, -shift)

# ===================================
# Let's test the program
# ===================================
print("=" * 50)
print("SIMPLE CAESAR CIPHER")
print("=" * 50)

# Get message from user
user_message = input("\nEnter your message: ")
# Get shift from user
user_shift = int(input("Enter shift number (1-25): "))
# Make sure shift is within range
user_shift = user_shift % 26
# Encrypt the message
encrypted_message = encrypt(user_message, user_shift)
# Show results
print("\n" + "-" * 30)
print("RESULTS:")
print("-" * 30)
print(f"Original:  {user_message}")
print(f"Encrypted: {encrypted_message}")

# Decrypt to verify
decrypted_message = decrypt(encrypted_message, user_shift)
print(f"Decrypted: {decrypted_message}")
# Check if it worked
if user_message == decrypted_message:
    print("\n✓ Success! The message was properly encrypted and decrypted.")
else:
    print("\n✗ Something went wrong.")
# Show example
print("\n" + "-" * 30)
print("HOW IT WORKS:")
print("-" * 30)
print("A → B → C → D ... (shift 1)")
print("HELLO → KHOOR (shift 3)")
print("Only letters change, spaces stay the same")
```

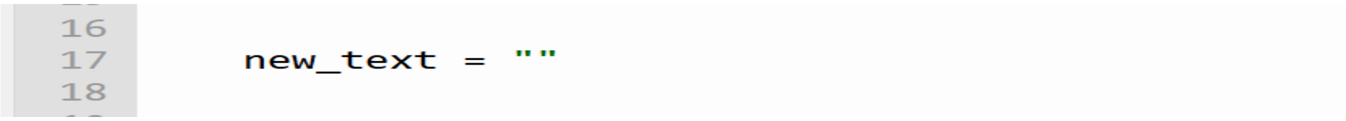
Program Description:

This program implements the Caesar Cipher algorithm to encrypt and decrypt messages. The Caesar Cipher shifts letters by a fixed number in the alphabet.

$$HELLO\ (shift\ 3)\ \rightarrow\ KHOOR$$

Function: caesar_encrypt(text, shift)

Step#1



```
    new_text = ""
```

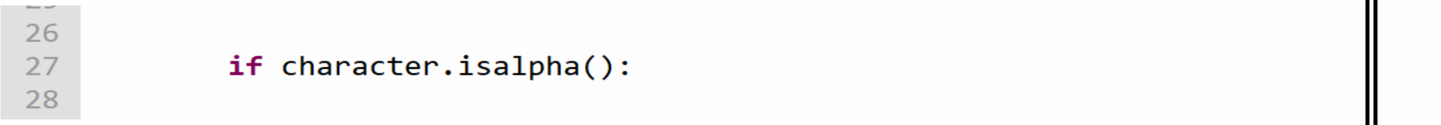
create an empty string to store encrypted result

Step#2

```
    for character in text:
```

Loops through each character of the input message.

Step#3

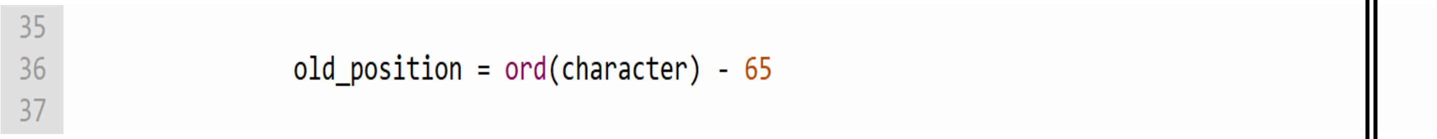


```
        if character.isalpha():
```

Checks if the character is a letter.

- If yes → shift it.
- If no → keep it unchanged.

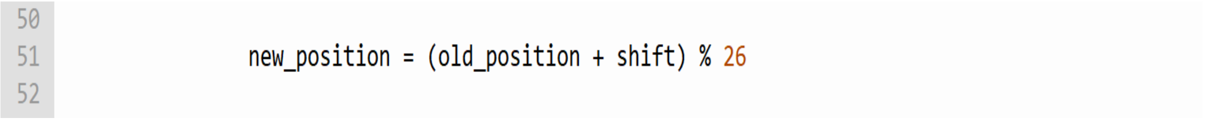
Step 4: Uppercase Handling



```
                old_position = ord(character) - 65
```

Convert letter to number (A=0, B=1...).

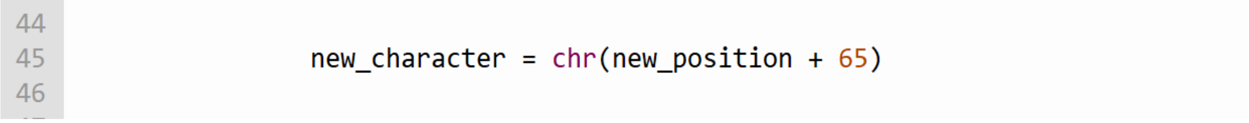
Step 5:



```
                new_position = (old_position + shift) % 26
```

add shift and uses modulo 26 to wrap around alphabet.

Step 6:



```
                new_character = chr(new_position + 65)
```

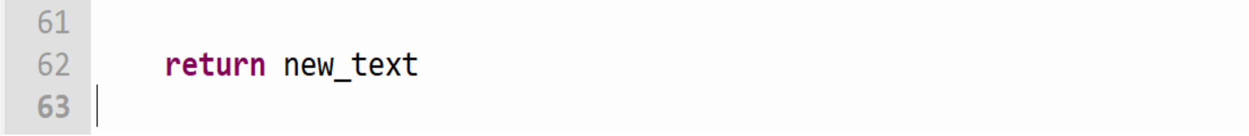
convert number back to letter

Step 7:

If lowercase:

Same process but subtract 97 instead of 65

Step 8:



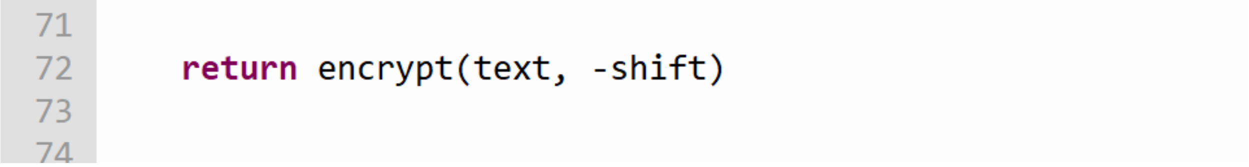
```

    return new_text

```

Returns encrypted message

Decryption Function.



```

    return encrypt(text, -shift)


```

To decrypt, we shift backwards (negative shift).

Security Analysis.

Type of Cipher.

The Caesar Cipher is a **substitution cipher**

Key Space.

Only 25 possible keys (1–25 shifts).

Security Level

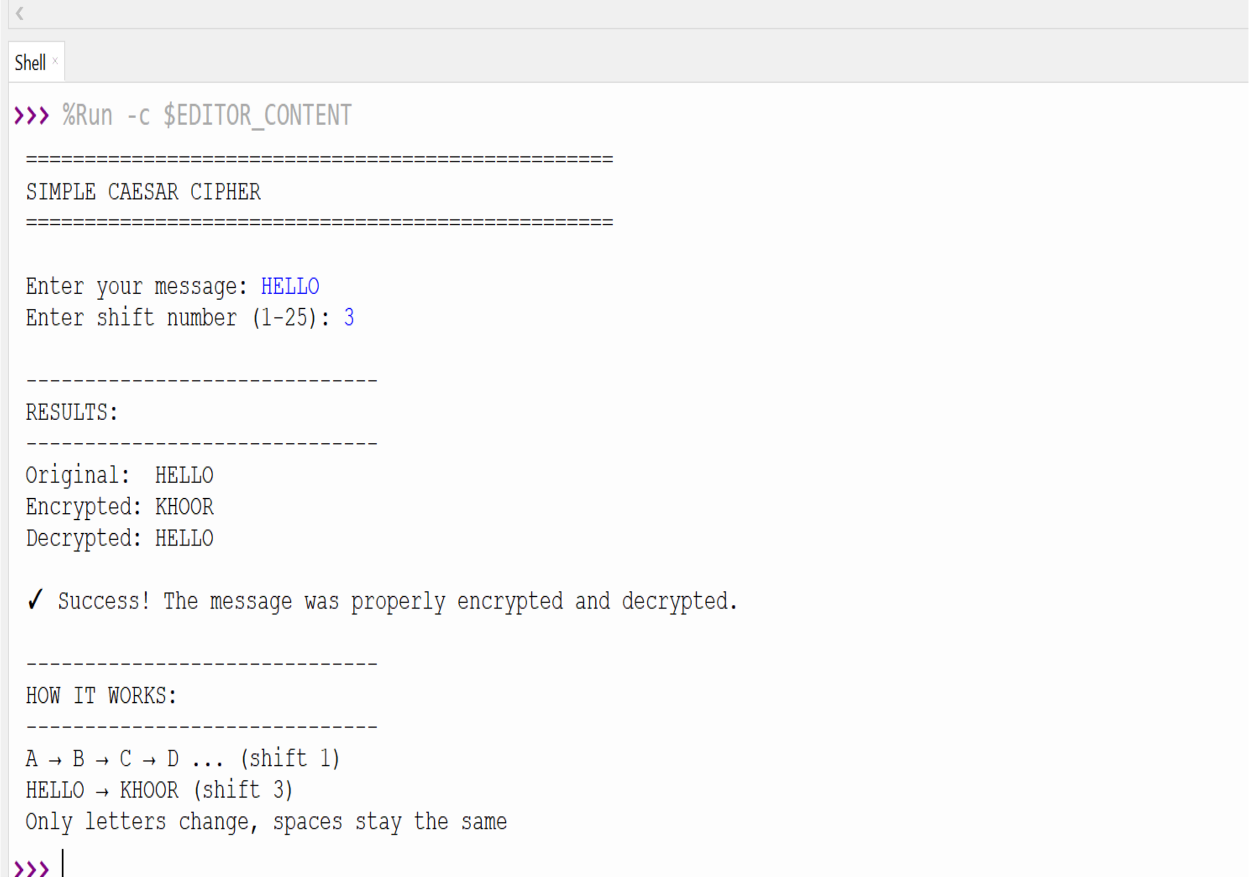
- Very weak.
- Can be broken using brute force easily.
- Vulnerable to frequency analysis.

Why It Is Not Secure Today?

It was used before more than 1000-2000 years. Modern encryption algorithms like:

- Advanced Encryption Standard
- RSA

OUTPUT:



```
>>> %Run -c $EDITOR_CONTENT
 ==================================================
 SIMPLE CAESAR CIPHER
 ==================================================

 Enter your message: HELLO
 Enter shift number (1-25): 3

 ------------------------------
 RESULTS:
 ------------------------------
 Original:  HELLO
 Encrypted: KHOOR
 Decrypted: HELLO

 ✓ Success! The message was properly encrypted and decrypted.

 ------------------------------
 HOW IT WORKS:
 ------------------------------
 A → B → C → D ... (shift 1)
 HELLO → KHOOR (shift 3)
 Only letters change, spaces stay the same
>>> 
```