COMSATS UNIVERSITY ISLAMABAD

Attock Campus

Department Of Computer Science

M SUFYAN | FA24-BSE-049

Information security

Instructor: Ms. Amreen Gull

Submission date

FEB 28, 2026

CODE:

```python
def caesar_encrypt(text, shift):
    result = ""
    for char in text:

        if char.isupper():
            result += chr((ord(char) - ord('A') + shift) % 26 + ord('A'))

        elif char.islower():
            result += chr((ord(char) - ord('a') + shift) % 26 + ord('a'))

        else:
            result += char

    return result


def caesar_decrypt(ciphertext, shift):
    result = ""

    for char in ciphertext:
        if char.isupper():
            result += chr((ord(char) - ord('A') - shift) % 26 + ord('A'))

        elif char.islower():
            result += chr((ord(char) - ord('a') - shift) % 26 + ord('a'))
```

```python
        elif char.islower():
            result += chr((ord(char) - ord('a') - shift) % 26 + ord('a'))

        else:
            result += char

    return result


if __name__ == "__main__":
    text = input("Enter your message: ")
    shift = int(input("Enter shift value: "))

    encrypted = caesar_encrypt(text, shift)
    print("Encrypted Text:", encrypted)

    decrypted = caesar_decrypt(encrypted, shift)
    print("Decrypted Text:", decrypted)
```

DESCRIPTION:

> Function: `caesar_encrypt(text, shift)`

This function encrypts (locks) the message.

```
def caesar_encrypt(text, shift):
```

It takes:

- `text` → the message you want to encrypt
- `shift` → how many letters forward you want to move

```
result += chr((ord(char) - ord('A') + shift) % 26 + ord('A'))
```

let's break this:

`ord(char)` → converts letter into number (ASCII value)
Example:

- ord('A') = 65
- ord('B') = 66

```
ord(char) - ord('A')
```

This converts A–Z into numbers 0–25
Example:

- A → 0
- B → 1
- C → 2

```
+ shift
```

We move forward by shift value.

```
% 26
```

This is VERY important.
There are 26 letters in alphabet.
If we go after Z, it comes back to A.

Example:
Z (25) + 3 = 28
28 % 26 = 2 → which is C

So this makes it wrap around alphabet.

`+ ord('A')`
We convert it back to ASCII.

`chr(...)`
This converts number back to letter.

if the letter is small

elif char.islower():

Same logic but for a–z.

We use `ord('a')` instead of `ord('A').`

If it is not a letter:

else:

result += char

If it is space, number, symbol etc
It stays SAME.

Example:
Hello 123 → Khoor 123
Numbers don't change.

Return encrypted result:

return result

Function: `caesar_decrypt(ciphertext, shift)`

This function unlocks the message.

Everything is SAME as encryption
But here:

Instead of `+ shift`
We use `- shift`

Because to go back we move backward.

```
result += chr((ord(char) - ord('A') - shift) % 26 + ord('A'))
```

Main Program

if __name__ == "__main__":

This means:
Run this part only if file is executed directly.

🔻 Taking Input

```
text = input("Enter your message: ")
shift = int(input("Enter shift value: "))
```

User enters:

- Message
- Shift number

🔻 Encrypt

```
encrypted = caesar_encrypt(text, shift)
print("Encrypted Text:", encrypted)
```

Call encryption function and print result.

🔻 Decrypt

```
decrypted = caesar_decrypt(encrypted, shift)
print("Decrypted Text:", decrypted)
```

It decrypts the encrypted message and shows original message again.