

**COMSATS University Islamabad, Attock Campus**

**Department of Computer Science**

**Sheema**

**SP24-BSE-012**

**20 OCTOBER,2025**

**INFORMATION SECURITY LAB MID**

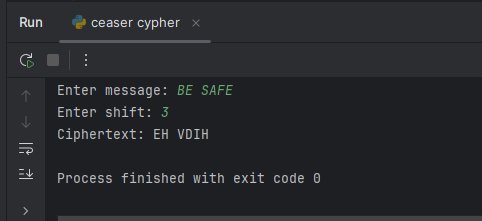
**QUESTION 1:**Write a Python program that encrypts a message using the Caesar Cipher with a user-given shift value.  
  
Example:  
Enter message: hello  
Enter shift: 3  
Ciphertext: khoor  
  
Hint: Use ord() and chr() for shifting letters.

*[CLO4: Implement a cryptographic algorithm to ensure information security.]*

CODE:

message = input("Enter message: ")  
shift = int(input("Enter shift: "))  
ciphertext = ""  
for ch in message:  
 if ch.isalpha():  
 start = ord('A') if ch.isupper() else ord('a')  
  
 ciphertext += chr((ord(ch) - start + shift) % 26 + start)  
 else:  
  
 ciphertext += ch  
  
print("Ciphertext:", ciphertext)

OUTPUT:



**Question 2**

**Vigenère Cipher (Encryption Only)**

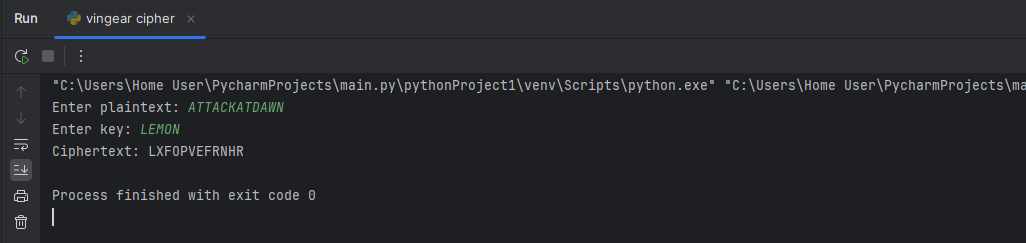
Write a Python program to encrypt a plaintext message using the Vigenère Cipher. Ask the user for plaintext and keyword. Display the ciphertext only.  
  
Example:  
Enter plaintext: ATTACKATDAWN  
Enter key: LEMON  
Ciphertext: LXFOPVEFRNHR

*[CLO4: Implement a cryptographic algorithm to ensure information security.]*

CODE:

def vigenere\_encrypt(plaintext, key):  
 plaintext = plaintext.upper()  
 key = key.upper()  
 ciphertext = ""  
 key\_index = 0  
 key\_length = len(key)  
  
 for ch in plaintext:  
 if ch.isalpha(): # Encrypt only letters  
 shift = ord(key[key\_index % key\_length]) - ord('A')  
 encrypted\_char = chr((ord(ch) - ord('A') + shift) % 26 + ord('A'))  
 ciphertext += encrypted\_char  
 key\_index += 1  
 else:  
 ciphertext += ch # Keep symbols/spaces unchanged  
 return ciphertext  
  
  
plain\_text = input("Enter plaintext: ")  
key\_word = input("Enter key: ")  
  
cipher\_text = vigenere\_encrypt(plain\_text, key\_word)  
print("Ciphertext:", cipher\_text)

OUTPUT:



**QUESTION 4:**

Caesar Cipher Decryption (Simple Brute Force)

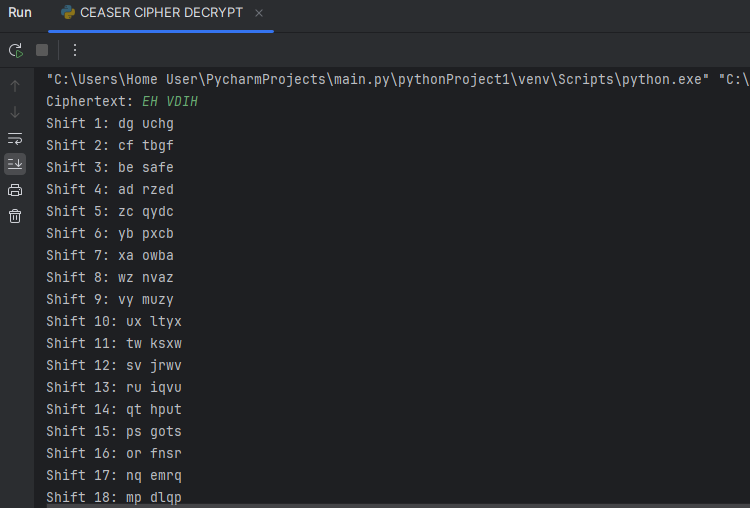
You are given a Caesar Cipher ciphertext. Write a Python program to try all 25 possible shift values and print each possible plaintext.  
  
Example Input:  
Ciphertext: BE SAFE  
  
Expected Output:  
Shift 1: jgnnq  
Shift 2: ifmmp  
Shift 3: hello <-- correct plaintext

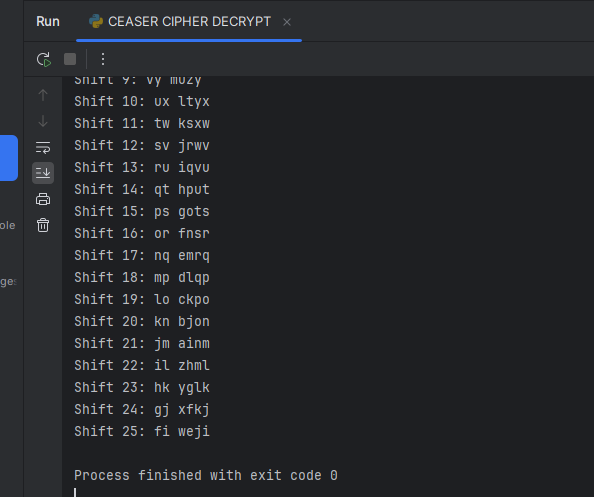
*[CLO4: Implement a cryptographic algorithm to ensure information security.]*

CODE:

ciphertext = input("Ciphertext: ")  
  
# Try all possible 25 shifts  
for shift in range(1, 26):  
 plaintext = ""  
 for ch in ciphertext:  
 if ch.isalpha():  
 start = ord('A') if ch.isupper() else ord('a')  
 plaintext += chr((ord(ch) - start - shift) % 26 + start)  
 else:  
 plaintext += ch  
 print(f"Shift {shift}: {plaintext.lower()}")

OUTPUT:





QUESTION 5:

DES and AES

Answer briefly:  
  
a) One difference between DES and AES.   
b) AES block size and one key size.   
c) One reason why AES is more secure than DES.

1. **One difference between DES and AES.**

* DES is old and weak algorithm that uses 56-bits key to encrypt data of 64-bit blocks
* AES is advanced and strong algorithm as compared to DES it uses

128,192 ,256 bits keys to encrypt data of 128 blocks.

1. **AES block size and one key size.**

AES always works with a **block size of 128 bits,** which means it processes data in fixed chunks of 128 bits at a time. This helps the algorithm handle data efficiently and securely.

An example of AES key size is 128 bits, but AES also supports 192 bits and 256 bits based on the security level required. Stronger encryption is offered by larger key sizes.

1. **One reason why AES is more secure than DES**

AES is safer than DES primarily due to its bigger key sizes, which give it resistance to brute-force attacks. DES with its 56-bit key is easily broken with modern computers, whereas AES with a 128-bit or bigger key is very hard to break and hence gives much better protection for confidential data.