program for polyalphabetic substitution cipher uses a separate monoalphabetic substitution cipher for each successive letter of plaintext, depending on a key.

Program

:

# Simple Polyalphabetic (Vigenère) Cipher

def encrypt(plaintext, key):

ciphertext = ""

plaintext = plaintext.lower().replace(" ", "")

key = key.lower()

for i in range(len(plaintext)):

p = ord(plaintext[i]) - ord('a')

k = ord(key[i % len(key)]) - ord('a')

c = (p + k) % 26

ciphertext += chr(c + ord('a'))

return ciphertext

def decrypt(ciphertext, key):

plaintext = ""

ciphertext = ciphertext.lower()

key = key.lower()

for i in range(len(ciphertext)):

c = ord(ciphertext[i]) - ord('a')

k = ord(key[i % len(key)]) - ord('a')

p = (c - k + 26) % 26

plaintext += chr(p + ord('a'))

return plaintext

# Main Program

plaintext = input("Enter plaintext: ")

key = input("Enter key: ")

encrypted = encrypt(plaintext, key)

decrypted = decrypt(encrypted, key)

print("\nEncrypted text:", encrypted)

print("Decrypted text:", decrypted)

output

