import string

def affine\_encrypt(plain\_text, a, b):

alphabet = string.ascii\_uppercase

encrypted\_text = ""

for char in plain\_text:

if char in alphabet:

index = alphabet.index(char)

encrypted\_index = (a \* index + b) % 26

encrypted\_char = alphabet[encrypted\_index]

encrypted\_text += encrypted\_char

else:

encrypted\_text += char

return encrypted\_text

def affine\_decrypt(cipher\_text, a, b):

alphabet = string.ascii\_uppercase

decrypted\_text = ""

a\_inverse = 0

for i in range(26):

if (a \* i) % 26 == 1:

a\_inverse = i

break

for char in cipher\_text:

if char in alphabet:

index = alphabet.index(char)

decrypted\_index = (a\_inverse \* (index - b)) % 26

decrypted\_char = alphabet[decrypted\_index]

decrypted\_text += decrypted\_char

else:

decrypted\_text += char

return decrypted\_text

# Example usage:

plain\_text = input("Enter the plain text: ")

a = int(input("Enter the multiplicative value: "))

b = int(input("Enter the additive value: "))

encrypted\_text = affine\_encrypt(plain\_text, a, b)

print("Encrypted text:", encrypted\_text)

decrypted\_text = affine\_decrypt(encrypted\_text, a, b)

print("Decrypted text:", decrypted\_text)