Program25 C program for set of blocks encoded with the RSA algorithmn

import math

def gcd(a, b):

while b:

a, b = b, a % b

return a

def common\_factor\_attack(n, e, blocks):

common\_factor = None

for i in range(len(blocks)):

for j in range(i + 1, len(blocks)):

if gcd(blocks[i], n) != 1 and gcd(blocks[j], n) != 1:

common\_factor = gcd(blocks[i], n)

break

if common\_factor:

p = common\_factor

q = n // p

phi = (p - 1) \* (q - 1)

d = pow(e, -1, phi) # Modular multiplicative inverse of e modulo

phi

plaintext\_blocks = [pow(block, d, n) for block in blocks]

return plaintext\_blocks

else:

return None

# Example values

n = 3233 # Modulus

e = 17 # Public exponent

blocks = [1791, 123, 2509, 1281] # Encrypted blocks

plaintext\_blocks = common\_factor\_attack(n, e, blocks)

if plaintext\_blocks:

print("Plaintext blocks:", plaintext\_blocks)

else:

print("No common factor found.")

OUTPUT:

No common factor found.