program for RSA system, the public key of a given user is e = 31, n = 3599. What is the private key of this user? Hint: First use trial-and-error to determine p and q; then use the extended Euclidean algorithm to find the multiplicative inverse of 31 modulo f(n).

# RSA private key finder (simple, educational)

# Given public key (e, n) it finds p,q by trial division, computes phi(n), then d = e^{-1} mod phi(n).

e = 31

n = 3599

def find\_factors(n):

"""Simple trial-division to find p and q (assumes n is product of two primes)."""

for i in range(2, int(n\*\*0.5) + 1):

if n % i == 0:

return i, n // i

return None, None

def extended\_gcd(a, b):

"""Return (g, x, y) such that a\*x + b\*y = g = gcd(a, b)."""

if b == 0:

return a, 1, 0

g, x1, y1 = extended\_gcd(b, a % b)

x = y1

y = x1 - (a // b) \* y1

return g, x, y

# --- run steps ---

p, q = find\_factors(n)

if p is None:

print("Failed to factor n.")

else:

print("Found factors: p =", p, "q =", q)

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

print("phi(n) =", phi)

g, x, y = extended\_gcd(e, phi)

if g != 1:

print("e and phi(n) are not coprime; no modular inverse.")

else:

d = x % phi # modular inverse (positive)

print("Private exponent d =", d)

# quick verification

print("Check: (e \* d) mod phi(n) =", (e \* d) % phi)

