17) Write a python program for set of blocks encoded with the RSA algorithm and we don’t have the private

key. Assume n = pq, e is the public key. Suppose also someone tells us they know one of the plaintext

blocks has a common factor with n. Does this help us in any way?

PROGRAM:-

```
import math


def gcd(a, b):
    while b:
        a, b = b, a % b
    return a


def gcd_attack(n, plaintext_block):
    """
    Attempts to factor n using a plaintext block with a common factor with n.
    """
    factor = gcd(n, plaintext_block)
    if 1 < factor < n:
        p = factor
        q = n // p
        return p, q
    else:
        return None, None


# Example inputs
n = 3599  # suppose this is the public RSA modulus
e = 31    # public exponent


# Attacker knows that one plaintext block has a common factor with n
plaintext_block = 59  # This is p (i.e., gcd(59, 3599) = 59)
```

```
p, q = gcd_attack(n, plaintext_block)

if p and q:
    print(f"Success! Factors of n found: p = {p}, q = {q}")
    phi = (p - 1) * (q - 1)

    # Compute the private key using Extended Euclidean Algorithm
    def mod_inverse(e, phi):
        def extended_gcd(a, b):
            if a == 0:
                return b, 0, 1
            gcd, x1, y1 = extended_gcd(b % a, a)
            x = y1 - (b // a) * x1
            y = x1
            return gcd, x, y

        gcd, x, _ = extended_gcd(e, phi)
        if gcd != 1:
            raise Exception("No modular inverse exists")
        return x % phi

    d = mod_inverse(e, phi)
    print(f"Private key d = {d}")
else:
    print("No common factor found; attack failed.")
```

OUTPUT:-

```
Success! Factors of n found: p = 59, q = 61
Private key d = 3031
```
