

Question 4

This script applies the Euclidean algorithm to determine the GCD of two large numbers, then finds a pair of large numbers whose GCD is 71.

Euclidean Algorithm

```
In [ ]: def GCD(a, b, verbose=True):
    # Handle case where a or b (or both) is zero: GCD(0, x) = GCD(x, 0) = 0
    if a == 0 or b == 0:
        return a if (a != 0) else b
    # Order such that a > b
    a, b = (b, a) if (b > a) else (a, b)
    # Euclidean algorithm
    while b != 0:
        x, r = a // b, a % b
        if verbose:
            print(f"GCD({a}, {b}): {a} = {b} * {x} + {r}")
        a, b = b, r
    return a

print(f"\n{GCD(313451351, 513454536) = }")
```

GCD Reverse

```
In [ ]: import math

# Multiply a and b by different prime numbers until both have length > 10 digits
base = 71; a = base; b = base # NOTE: 71 is prime
while math.log10(a) < 9:
    a *= 17
while math.log10(b) < 9:
    b *= 19
print(a, b)
```

```
In [ ]: # Confirm GCD(a, b) = 71
print(f"\n{GCD(a, b) = }")
```