Math 440/540 Homework 1

1. (# 1.1 in Stein)

   (a) (by hand) Compute the greatest common divisor $\gcd(455, 1235)$ *using the Euclidean algorithm.*

   (b) (using python or SageMath) Compute the greatest common divisor $\gcd(455, 1235)$.

```
import math
def compute_gcd(a,b):
    mod = b % a
    if mod == 0:
        return a
    else:
        return compute_gcd(mod, b)

print(f"1.b {compute_gcd(455,1235)}")
```

   Output:

```
1.b 35
```

2. (using python or SageMath) A *Mersenne prime* is a prime of the form $2^p - 1$, for prime p. Define $M_n := 2^n - 1$.

   (a) What is the least prime p such that M_p is composite?

```
def find_p():
    from sympy import isprime
    p = 0
    x = 0
    while x == 0:
        p+=1
        if not isprime(2**p - 1) and isprime(p):
            x=1
    return p

print(f"2.a {find_p()}")
```

   Output:

```
2.a 11
```

(b) Find all primes up to 100 such that M_p is prime.

```
def find_all_primes():
    from sympy import isprime
    prime_list = []
    for i in range(2,100):
        if isprime(i) and isprime(2**i - 1):
            prime_list.append(i)
    return prime_list

print(f"2.b {find_all_primes()}")
```

Output:

```
2.b [2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107]
```

3. (# 1.14 in Stein)

   (by hand) Prove that a positive integer $n \geq 2$ is prime if and only if n is not divisible by any prime p with $1 < p \leq \sqrt{n}$.

4. (using python or SageMath) Let p_k denote the kth prime, i.e., $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, $p_4 = 7$, etc. Define $N_k := p_1 p_2 \cdots p_k + 1$.

(a) Find the least positive integer k such that N_k is composite.

```
def find_k():
    from sympy import isprime
    p = 0
    x = 0
    k = 0
    prime_list2 = []
    while x == 0:
        p+=1
        if isprime(p):
            k += 1
            prime_list2.append(p)
            n = math.prod(prime_list2) + 1
            if not isprime(n):
                x = 1

    return k

print(f"4.a {find_k()}")
```

Output:

```
4.a 6
```

(b) Find all positive integers k up to 100 such that N_k is prime.

```
def find_ks():
    from sympy import isprime, primerange
    k_list = []
    prime_list = []
    k = 0
    for p in primerange(0,100):
        k+=1
        prime_list.append(p)
        n = math.prod(prime_list) + 1

        if isprime(n):
            k_list.append(k)

    return k_list

print(f"4.b {find_ks()}")
```

Output:

```
4.b [1, 2, 3, 4, 5]
```

5. (#1.3 in Stein)

   (*) (by hand) Prove that there are infinitely many primes of the form $6x - 1$. Note: you are proving a special case of Dirichlet's Theorem, so don't appeal to that theorem in your proof.