Math 440/540 Homework 3

1. (#2.30 in Stein) (by hand) Use fast exponentiation together with Euler's Theorem to compute the last two digits of 3^{45}.

   Answer: First we set up the problem we are trying to solve in modulo terms which is

   $$3^{45}(mod 100)$$

   From there we can use Eulers Theorem

   $$3^{\phi(100)} \equiv 1(mod 100) => 3^{40} \equiv 1(mod 100)$$

   Using that we can then reduce our equation down to

   $$3^{5}(mod 100)$$

   since $3^{40} \equiv 1(mod 100)$ Then we can use fast exponentiation to calculate our final value

   $$3^2 = 9 => 3^4 = 91 => 3^5 = 243 \equiv 43 mod(100)$$

2. (#2.32 in Stein) (using python or SageMath) Find the proportion of primes $p < 1000$ such that 2 is a primitive root modulo p (i.e. 2 has order $p - 1$ modulo p).

```
primes = prime_range(1,1000)
prim_count = sum(primitive_root(p)==2 for p in primes)
print(f"num of primes {len(primes)}")
print(f"primitive root count {prim_count}")
print(f"proportion of primes {prim_count/len(primes)}")
======
num of primes 168
primitive root count 67
proportion of primes 0.39880952380952384
```

3. (using python or SageMath) Determine the smallest composite n that is a base 2 pseudoprime (i.e. $2^{n-1} \equiv 1 \pmod{n}$). What is the least base $a > 1$ for which the Pseudoprimality Theorem proves n is composite?

```
n = 2
while True:
    if not is_prime(n):

        if mod(2^(n-1),n)==1:
            break
```

```
    n+=1
print(n)
===
341

n = 341
for a in range(2,341):
    if mod(a^(n-1),n)!=1:
        print(a)
        break
===
3
```

4. (#3.4 in Stein) (by hand with calculator or using python or SageMath) You and Nikita wish to agree on a secret key s using the Diffie-Hellman key exchange. Nikita announces that $p = 3793$ and $g = 7$. She secretly chooses a number $n < p$ and tells you that $g^n \equiv 454 \pmod{p}$. You choose the random number $m = 1208$. What is the secret key?

```
p = 3793
g = 7
m=1208
key = power_mod(454,m,p)
print(key)
===
2156
```

5. (using python or SageMath) Let $n = 14380057$.
   (a) Determine the largest block size you can use to encode strings of text for use with RSA with this n.
   (b) Using the block size in (a), encode the phrase 'HAPPY HALLOWEEN' as a list of numbers using our method from class.
   (c) If Nikita announces her public RSA key to be $(14380057, 7)$, encrypt your answer from (b) using Nikita's public key.

```
n = 14380057
max_block = floor(log(n,10)/2)
print(max_block)
msg = "HAPPY HALLOWEEN"
word_list = [msg[i:i+3] for i in range(0, len(msg), 3)]
```

```
print(word_list)

def encode_character(c):
    if c == " ":
        return 0
    else:
        return ord(c)-64
encode_list = []
for block in word_list:
    num = 0
    i = 0
    for c in block:
        num += encode_character(c)*27^i
        i += 1
    encode_list.append(num)
print(encode_list)
e = 7
encrypt_list = [pow(b, e, n) for b in encode_list]
print(encrypt_list)
===
3
['HAP', 'PY ', 'HAL', 'LOW', 'EEN']
[11699, 691, 8783, 17184, 10346]
[6961752, 2354377, 9032872, 3150324, 8876056]
```