In [1]:

```python
import pandas as pd


def gcd(a, b):
    """Calculates greatest common divisor of 2 numbers"""

    if (a == 0):
        return b
    return gcd(b % a, a)


def phi(n):
    """A simple method to evaluate Euler Totient Function"""

    result = 1
    for i in range(2, n):
        if (gcd(i, n) == 1):
            result+=1
    return result

def calculate_primitive_roots(prime_number: int) -> list:
    """Calculate all primitive roots of a prime number"""

    num_to_check = 0
    primitive_roots = []
    range_to_check = range(1, prime_number)
    primitive_roots_count = phi(prime_number - 1)

    for _ in range_to_check:
        num_to_check += 1
        candidate_prime_roots = []

        if len(candidate_prime_roots) == primitive_roots_count:
            break

        for i in range(1, prime_number):
            modulus = (num_to_check ** i) % prime_number
            if modulus not in candidate_prime_roots:
                candidate_prime_roots.append(modulus)

            if len(candidate_prime_roots) == prime_number - 1:
                primitive_roots.append(num_to_check)
                break

    return primitive_roots

data = pd.read_csv('ieeja.txt', header=None)
data.columns = ['p']
prime_list = data['p'].values.tolist()

for prime in prime_list:
    primitive_roots = calculate_primitive_roots(prime)

    print("Primitive roots of " + str(prime) + " are:")
    print(primitive_roots)
```

```
Primitive roots of 3 are:
[2]
Primitive roots of 5 are:
[2, 3]
Primitive roots of 7 are:
[3, 5]
Primitive roots of 11 are:
[2, 6, 7, 8]
Primitive roots of 13 are:
[2, 6, 7, 11]
Primitive roots of 17 are:
[3, 5, 6, 7, 10, 11, 12, 14]
Primitive roots of 19 are:
[2, 3, 10, 13, 14, 15]
```

In [ ]: