MAT 331 Fall 2023 Project
Primality testing

In this project we will investigate two methods to test whether or not a particular whole number is prime, called primality testing. An efficient method to decide whether or not a number is prime is particularly important in cryptography. For instance, the RSA method needs two very large prime numbers. How does one generate these prime numbers?

(1) Recall that a prime number doesn't have any divisors other than 1 and itself. A number is *composite* if it has at least one additional divisor, say d. Immediately, n/d is also a divisor of n. So, either $d \leq \sqrt{n}$ or $n/d \leq \sqrt{n}$. Using this to check for primality is called trial division. Check every number from 2 to $\sqrt{n}$ if it divides n. If it does, then n is definitely not prime, otherwise it is prime. Implement this trial division method in matlab as a function that takes in as an input a whole number and returns whether or not that number is prime.

(2) Using this method, write code to give a list of all primes up to 100,000, or some other large number. Do not display this list. How long does this take?

(3) Fermat's little theorem says that for a prime p and some number a relatively prime to p, the following holds

$$a^{p-1} = 1 \mod p. \tag{FLT}$$

In general, Fermat's little theorem does not hold for composite numbers. We will use this to create a primality test. Given a "suspected" prime p, choose an integer a so that $2 \leq a \leq p-1$. Check if the above equation (FLT) holds (you will want to use the command `powermod`). If not, ($a^{p-1} \neq 1 \mod p$), then p can not be prime. Implement this Fermat method as a matlab function which takes in as an input a whole number p and the value a and returns whether or not that number is prime.

(4) Using this method, write code to give a list of all primes up to 100,000, or the same large number in (2), with a random value of a chosen for each number. Do not display this list. How long does this take?

(5) Compare the list in (2) to the list in (4). Are they the same? Display a list of those numbers that appear in (4), but not in (2).

It is possible that (FLT) holds for a composite number. So, just because the equation holds does not mean it is prime, just probably prime. If the number was composite after all, we call a a Fermat liar. Moreover, it is possible for there to exist composite numbers that pass the test for any value a. Such numbers are called *Carmichael numbers*.

(6) For every value of a between 2 and $p-1$, check whether or not p passes the Fermat method. Do this for all p between 3 and 1,000 that are not prime (composite). Plot your results on a graph with the x-axis representing the value p and the y-axis representing the proportion of the values a that pass the test. Can you tell from the graph which numbers are Carmichael numbers?