This code implements various number theory algorithms and mathematical utilities. The code is organized into several classes, each focusing on specific mathematical concepts and operations. This task focuses on the features relevant to the prime calculations.

How It Works:

This code implements a toolkit with different algorithms related to various number theory and mathematical operations. For Perfect Numbers:

- Prime factorization is the process of decomposing a positive integer into a product of prime numbers. The code should correctly identify all prime factors of a given number, including repeated factors
- A perfect number is a positive integer that is equal to the sum of its proper positive divisors (excluding the number itself).
- The code should correctly identify perfect numbers and generate a list of perfect numbers up to a given limit.

Currently, the implementation of this feature contains a logic bug that causes the calculation relevant to the prime to deviate from its intended functionality.

Expected Outcomes:

- IsPerfectNumber(28) returns true.
- GetPrimeFactors(84) returns [2, 2, 3, 7].
- GetPrimeFactors(100) returns [2, 2, 5, 5].
- GeneratePerfectNumbers(100) returns [6, 28].

Actual Outcomes:

- IsPerfectNumber(28) returns true.
- GetPrimeFactors(84) returns [2, 2, 3, 7].
- GetPrimeFactors(100) returns [2, 2, 25].
- GeneratePerfectNumbers(100) returns [6, 28].