

## **Introduction:**

This program computes the greatest common factor and the least common multiple of two given numbers. It also computes the factors of each number and the common factors between them. It is an interactive program; it prints out a message asking the end-user to enter the two numbers.

## **Program Design:**

The program design is shown in the 'doc' directory of the project.

## **Results:**

The factorization process is completed without using the variable names primes to represent the prime numbers less than 100. This is done by looping through from 2 to the square root of num because once num is reduced to its prime number, the only factor is the number itself. This reduces the time complexity from  $O(n)$  to  $O(\sqrt{n})$ . Then I just check to see if num can be evenly divided by i (the loop counter). We can do this check using modulo which will return us the remainder. Since as long as we can divide by i, then we know it is a prime number and we just have to update num to be num / i. The key thing to note is we will never get a non-prime number when we reach the inner while loop because I check if num % i == 0 before the inner while loop. A non-prime number will always have a remainder. Once we exit out of the for loop, we know that num is either a prime number or it is equal to 1, so if it is not equal to one, num is added once more. The space complexity of this algorithm is constant since we are not allocating any new memory based on the input.

The greatest common factor is computed using the Euclidean algorithm. The first step is to take the larger of the two given numbers. We loop through until the remainder is not 0 because once the remainder is 0, we know that the previous remainder is the greatest common factor.

The least common multiple is computed using the known formula that  $|a*b|/GCF(a, b)$ . a and b representing the two numbers given. Since there is already an implementation for the greatest common factor, that method is just called in LeastCommonMultiple. There is a quick check to see if either of the numbers given are 0 because if they are then we immediately know that the least common multiple is 0. This optimizes the solution in cases where either one of them is 0.

In order to display the results, there is a helper function called PrintResults that will print everything that is needed after the computations are complete.

**Conclusion and Discussion:**

In conclusion, I think this was a very interesting assignment and a smooth transition to C#. After this assignment, I feel very comfortable with C# because it has such a strong similarity for Java. Sometimes I think that I am writing Java when I am really writing C# but it still compiles because of the strong similarities.