* + - Module Code: FC723
    - Class/Group: Group A
    - Module Title: Programming Theory
    - Assessment Title: Portfolio Project 1
    - Tutor Name: Sophie Norman
    - Student GUID Number: 3075604
    - Date of Submission: 26/01/2025

I thought of adding the process of finding the Least Common Multiple because it enhances the program's versatility by enabling it to address a broader range of problems. Instead of just calculating the GCD, it now also calculates the LCM, which is beneficial in situations such as scheduling, handling fractions, or solving mathematical problems that involve multiples. By incorporating both functions, the program transforms into a more comprehensive and practical tool for users, saving them time and effort.

Euclidean algorithm pseudocode extension

// Declaring 5 variables and specifying their data types

DECLARE Num1, Num2, originalNum1, originalNum2 : INTEGER

DECLARE Remainder : REAL

//Declaring 2 variables to store the results

DECLARE Gcd, Lcm : INTEGER

//Outputting a message for the user

OUTPUT “Enter two integers”

//Inputting two integers

INPUT Num1, Num2

//Storing the values of Num1 and Num2 before they are altered

originalNum1 <- Num1

originalNum2 <- Num2

//Using the IF statement to ensure that Num1 is the larger number

IF  Num2 > Num1 THEN

//Swapping Num1 and Num2

//Declaring temporary variable to store a value temporarily to swap variables without losing any data

DECLARE Temp : INTEGER

      Temp <- Num1

      Num1 <- Num2   
      Num2 <- Temp

ENDIF

//Using the WHILE loop to continue looping if the remainder is not = 0

WHILE  Remainder NOT = 0 DO

      //Calculating the remainder using MOD operation to return the remainder

Remainder <- Num1 MOD Num2

Num1 <- Num2

Num2 <- Remainder

ENDWHILE

//Putting Num1 in a new variable

Gcd <- Num1

//Calculating the least common multiple using the 2 original values before they were altered

Lcm <- (originalNum1 \* originalNum2) // Gcd

//Outputting the Greatest common divisor

OUTPUT “The Greatest common divisor is = ”,  Gcd

//Outputting the Least common multiple

OUTPUT “The Least Common Multiple is = ”, Lcm