gcc program.c # Compiles program.c and generates a default executable named a.out **./a.out** # Runs the compiled program
gcc program.c -o my_program # Compiles program.c and creates an executable named my_program **./my_program** # Runs the compiled program
gcc program.c -o input_program # it takes inputs from input1.txt and displays the result in the terminal. **./input_program < input.txt**
gcc program.c -o io_program **./io_program < input.txt > output.txt** # it takes the inputs from input1.txt and creates the myoutput.txt document and writes the result here.
diff output1.txt myoutput.txt # compares the document output1.txt with the document myoutput.txt.
diff --ignore-all-space output1.txt myoutput.txt # compares the document output1.txt with the document myoutput.txt without ignoring the spaces.

QUESTION: GCD and LCM Calculation Using the Euclidean Algorithm

Write a C program:

1. Take two positive integers as input
2. Calculate
    a. **Greatest Common Divisor (GCD)**
        i. The program should use the **Euclidean algorithm** to find the GCD
    b. **Least Common Multiple (LCM)**
        i. Compute the LCM based on the GCD
3. The program must also print the **STEP-BY-STEP PROCESS** of calculating the GCD.

Requirements: The program must use a **DO-WHILE LOOP** to compute the **GCD**.

$\gcd(a,b) = \begin{cases} a, & \text{if } b = 0 \\ \gcd(b, a \bmod b), & \text{otherwise.} \end{cases}$	$lcm(a,b) = \frac{\lvert ab \rvert}{\gcd(a,b)}$

Enter first number (a): 48 Enter second number (b): 18 Step 1: a = 48, b = 18, quotient = 2, remainder = 12 Step 2: a = 18, b = 12, quotient = 1, remainder = 6 Step 3: a = 12, b = 6, quotient = 2, remainder = 0 GCD found in 3 steps: 6 The LCM of 48 and 18 is: 144	Enter first number (a): 56 Enter second number (b): 42 Step 1: a = 56, b = 42, quotient = 1, remainder = 14 Step 2: a = 42, b = 14, quotient = 3, remainder = 0 GCD found in 2 steps: 14 The LCM of 56 and 42 is: 168