

Lab 03 Arithmetic Operations

November 19, 2024

Objective

Develop an assembly program that first calculates the greatest common divisor (GCD) of two numbers using the Euclidean Algorithm and then computes the factorial of the resulting GCD.

Lab Work

1. Input:

Define two variables, `num1` and `num2`, with initial values. Initialize variables to store the GCD (`gcd`) and the factorial result (`factorial`).

2. GCD Algorithm:

Implement the Euclidean Algorithm to find the GCD of `num1` and `num2`. Use a loop to repeatedly apply the algorithm until the remainder becomes zero. The GCD will be stored in `gcd`.

3. Factorial Algorithm:

Use the GCD result as the input for the factorial calculation. Implement a loop to multiply the current result by decreasing values from the GCD down to 1.

4. Output:

Display both the GCD and the factorial result as the output.

5. Evaluation:

Test your program with different pairs of `num1` and `num2`. Verify that the GCD and the factorial calculation are correct.

Hints

- Use the `div` instruction for division in the GCD algorithm.
- Use the `mul` instruction for multiplication in the factorial calculation.
- Initialize the factorial result variable to 1.
- Create separate loops for the GCD calculation and the factorial calculation.

Example Code Template

```
org 100h
mov ax, num1
mov bx, num2
gcd_loop:
;your code here
mov gcd, ax          ; Store the GCD in the result variable
factorial_loop:
; ;your code here
ret
num1 dw 36
num2 dw 48
gcd dw 0
factorial dw 1        ; Variable to store the factorial result
```