LAB 02 Tasks

Task 01: Implement Equations – 5 Marks

Imagine you are processing different arithmetic operations. Your job is to execute the following mathematical expressions using assembly language:

1. Calculate the sum of:
    - `50 + 31 + 20 + 15 + 74 + 14 – 0Bh`
2. Perform the subtraction and addition:
    - `10 - 4 + 200 - 150`
3. Sum up binary and hexadecimal values:
    - `10111b + 40Bh + 205d + 1010001b + E`
4. Execute a series of operations:
    - `10001101b – 062h + 255 + 5 + 11101b - A + B`
5. Handle binary, hexadecimal, and octal:
    - `1110b - 7 + 1B2h - 557o`

Task 02: Write a Program – 5 Marks

Now, imagine you are designing a program in assembly language to handle specific expressions. For all expressions, take `eax = ebx = ecx = edx = 0` **on R.H.S.**

1. Update the value in register `edx` based on the expression:
    - `edx = eax + 2 + ebx - ecx + 0Eh - 23o + 63d`
2. Modify the value in register `eax` using:
    - `eax = 6ABh - ebx + 47o + 55d – 110111b + 130`
3. Adjust the value in register `ebx` according to:
    - `ebx = 6BEh - eax + 23d + 61o – 11100101b + 6Ah`
4. Calculate the value in register `ecx` with:
    - `ecx = 1100101101b + 25h - 13o + ebx - ecx + 5`

In these scenarios, your task is to create assembly code that performs these operations, updating the values in the specified registers accordingly. Each line of code should represent one step in the mathematical expression, just like you would mentally solve a math problem.