COAL - LAB 4

Topic: Loops, Bitwise Operations, Shifting and Rotation

Objectives

- To understand and implement **loops** in Assembly.
- To practice **shifting (SHL, SHR)** and **rotations (ROL, ROR)**.
- To apply **bitwise logical operations (AND, OR, XOR, NOT)** for masking and manipulation.

In-Lab Tasks

Part A: Loop Programming

Task 1: Factorial

Write a program to calculate the **factorial of 5** using a loop. Store the result in a memory variable `fact`.

Hint: Start with `AX = 1` and multiply by decreasing values until `CX = 1`.

Task 2: Sum of Even Numbers

Write a program to calculate the **sum of all even numbers from 2 to 20** using a loop. Store the result in a variable `sum_even`.

Task 3: Reverse Array

Given an array of 5 numbers, write a program to **reverse it** (store in another array).

👉 **Example:** `[1, 2, 3, 4, 5]` → `[5, 4, 3, 2, 1]`.

Part B: Shifting, Rotation, and Bitwise Ops

Task 4: Multiply and Divide by 2

- Write a program that takes a number `num = 6`.
- Use **SHL** to multiply by 2.
- Use **SHR** to divide by 2.
  Store both results in memory.

Task 5: Rotate Example

- Take `AL = 1001 0110b`.
- Apply `ROL AL, 1` and `ROR AL, 1`.
- Store results in memory.

Task 6: Masking Bits

- Take a byte `AL = 1010 1101b`.
- Use **AND** to mask (keep) the **lower nibble (last 4 bits)**.
- Use **OR** to set the highest bit.
- Use **XOR** to toggle the second bit.

Record the values after each operation.