**2021/ICTS/33**

**TICT3113(P) – Computer Architecture & Organization**

**Lab sheet 05**

**Title:** Exercises for Comparison and Loop in ARM Assembly

**Aims:**

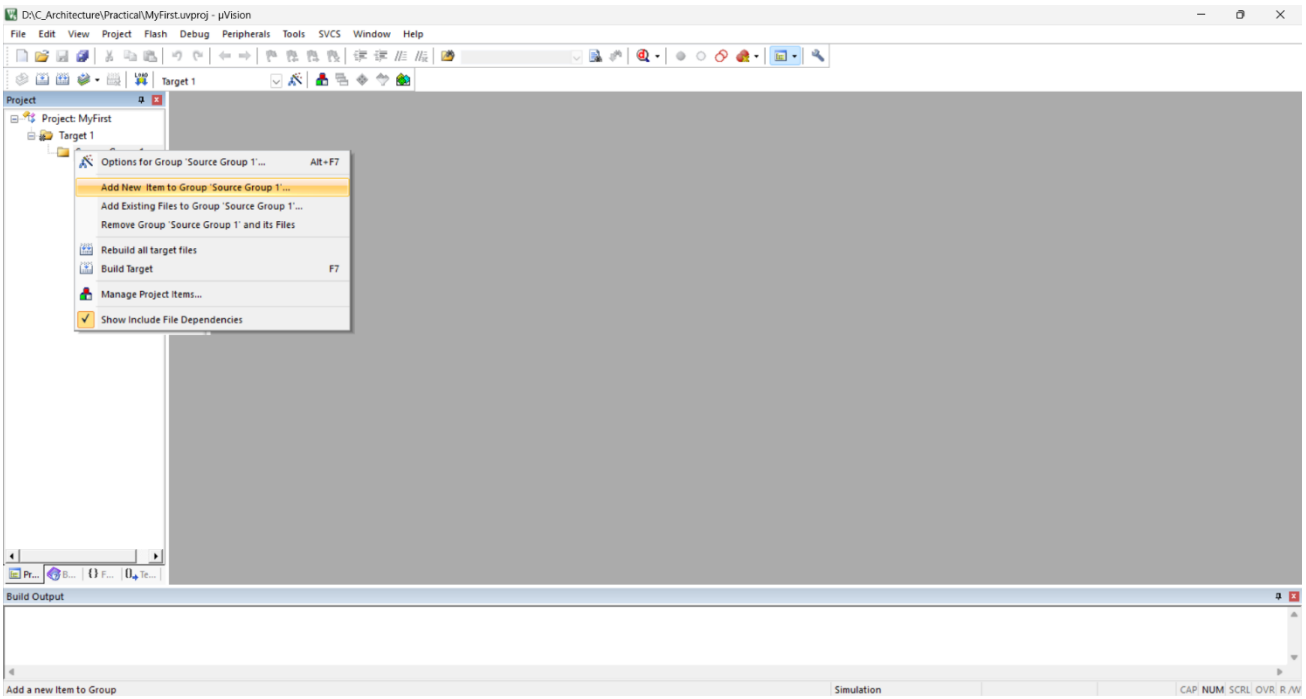
1. To understand how comparison operations are performed using ARM Assembly instructions.
2. To learn how loops are created using conditional and unconditional branch instructions.
3. To develop skills in writing, executing, and debugging loop-based and comparison- based ARM Assembly programs in Keil µVision.

**Tasks:**

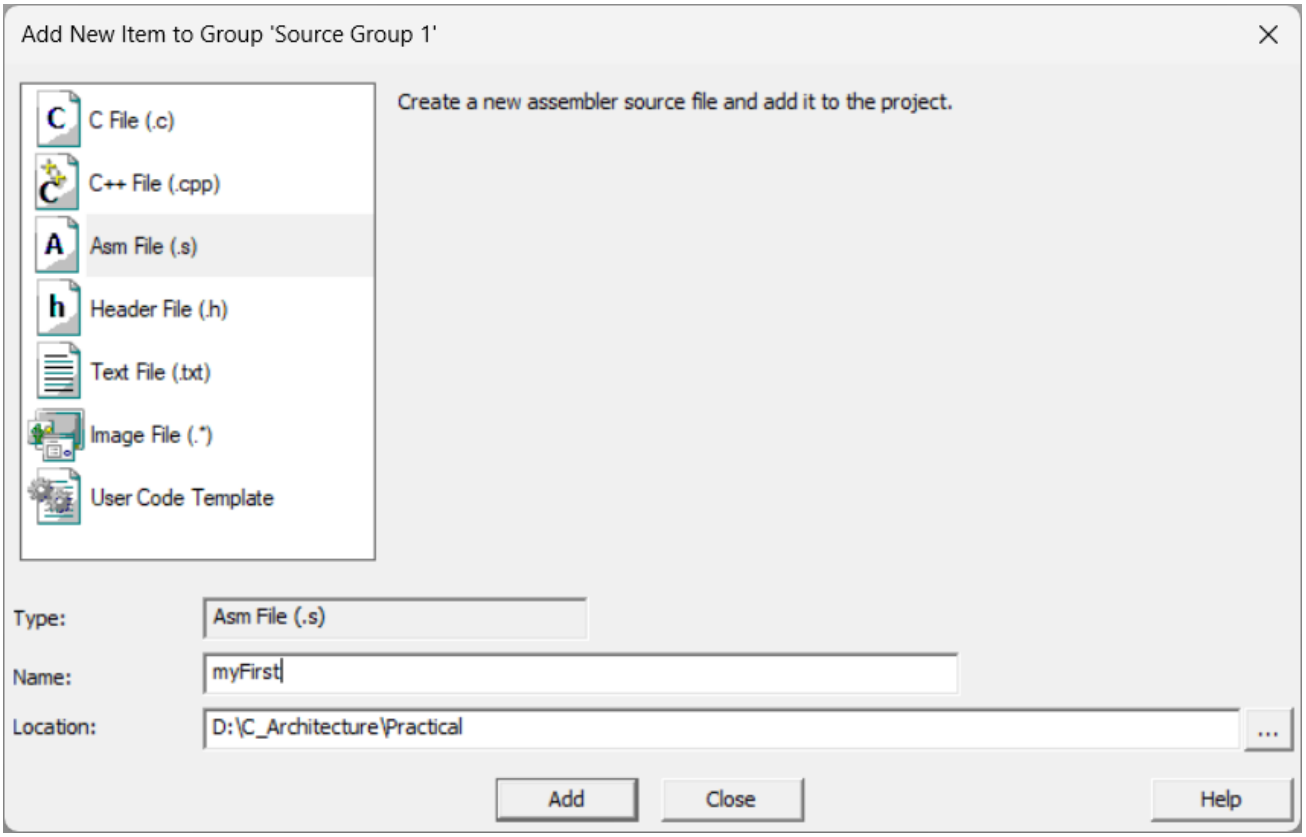
1. Identify and explain comparison and branch instructions such as CMP, BEQ, BNE, BGT, BLT, and B used in ARM Assembly for decision-making and looping.
2. Analyze the program flow by tracing each step to understand:
    a. How Comparisons Affect Branching
    b. How loops repeat based on conditions
    c. How ARM uses condition flags to control program execution

**Activities**

1. Click on the + next to Target 1 to expand the tree. Right click on Source Group 1 and choose Add New Item to Group.

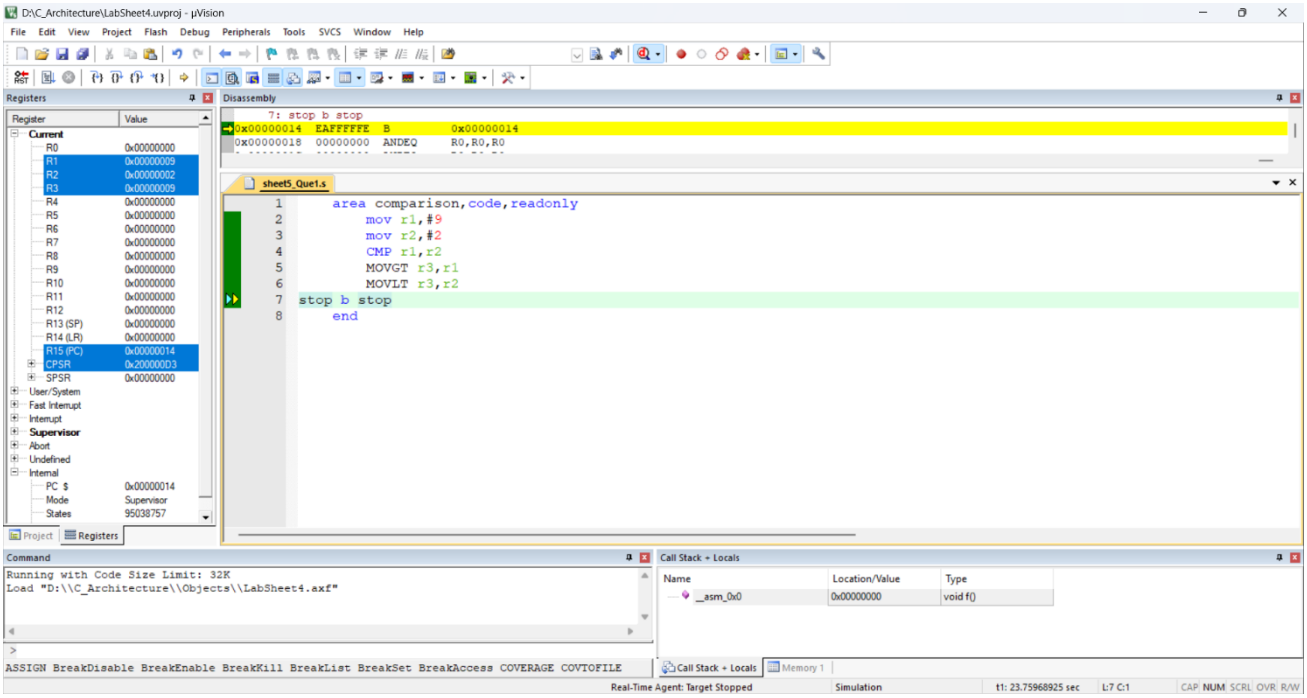



2. Select Asm File (. s) , then type program name First and Add to source.

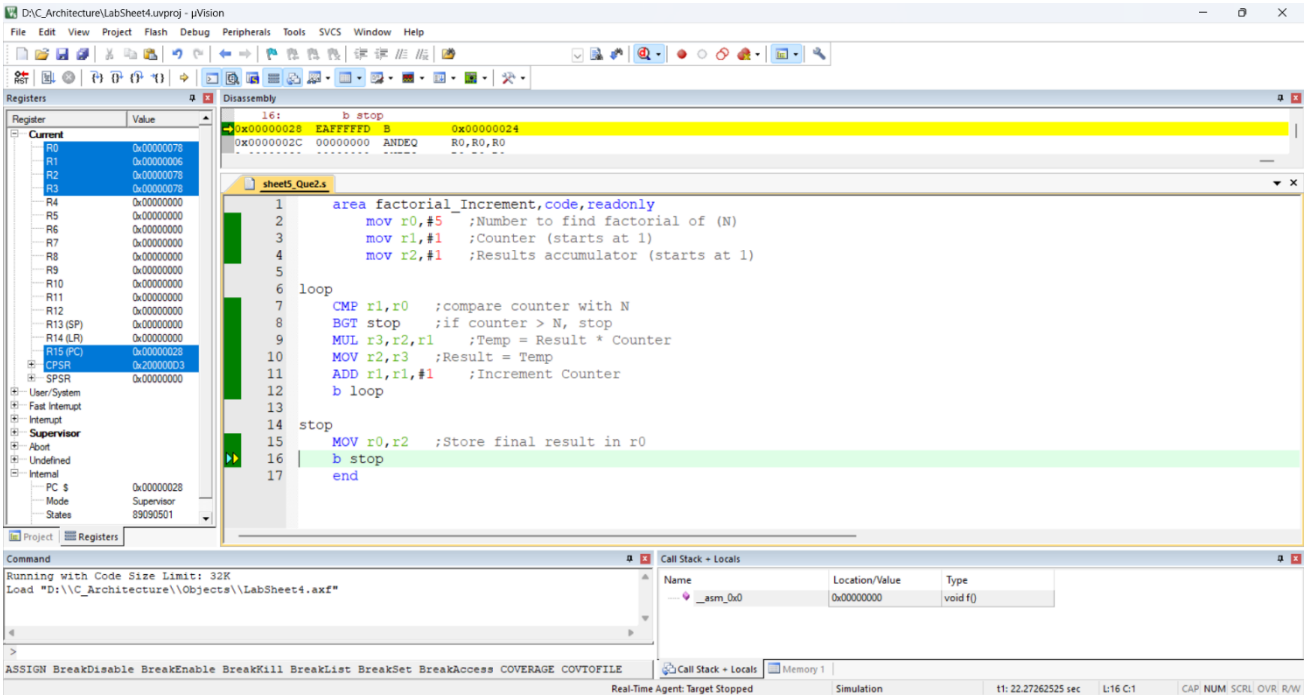



**Exercises:**

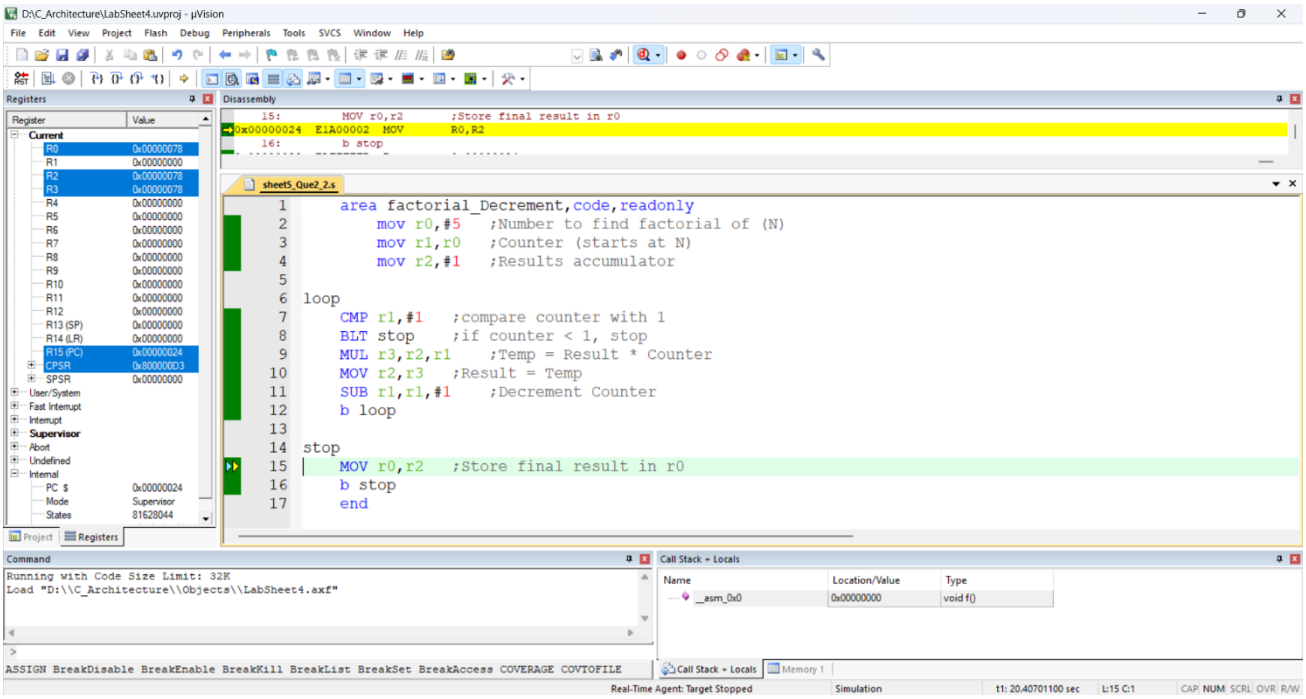
1. Find the larger number between two numbers and store the output in R3 register



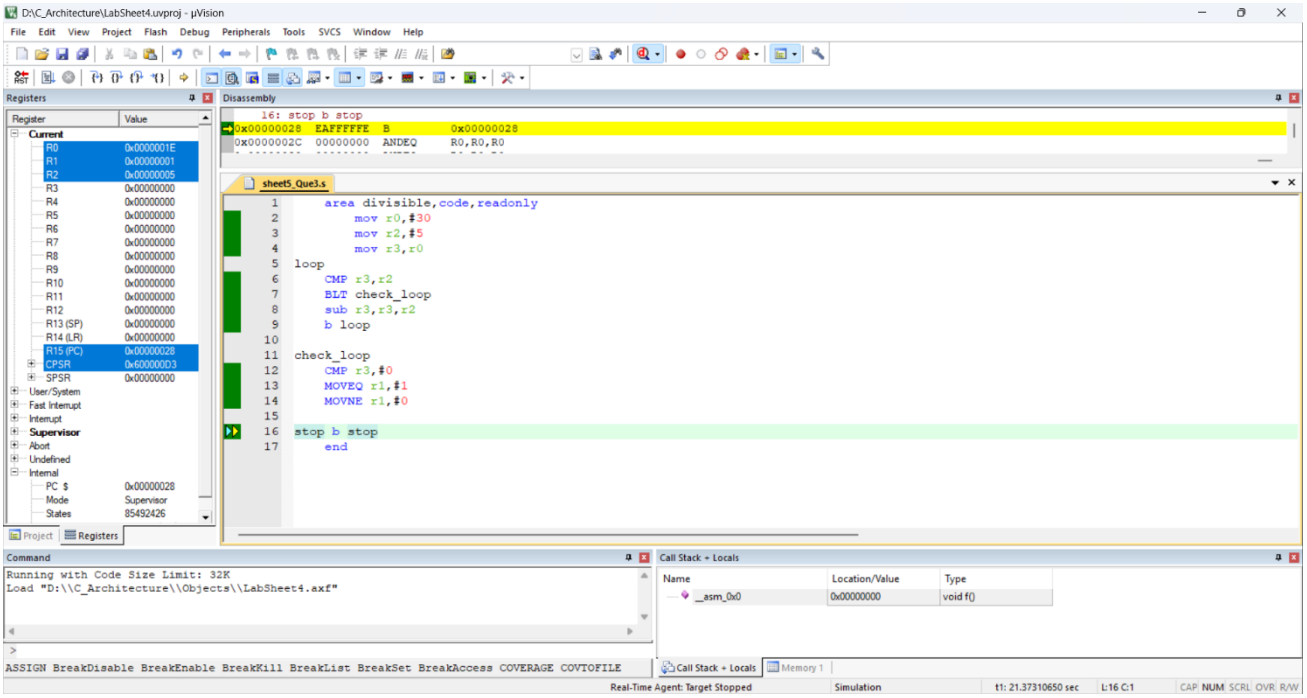
2. Write a program to find the factorial of a number in each way.
   - Increment Hint: 5! = 1*2*3*4*5



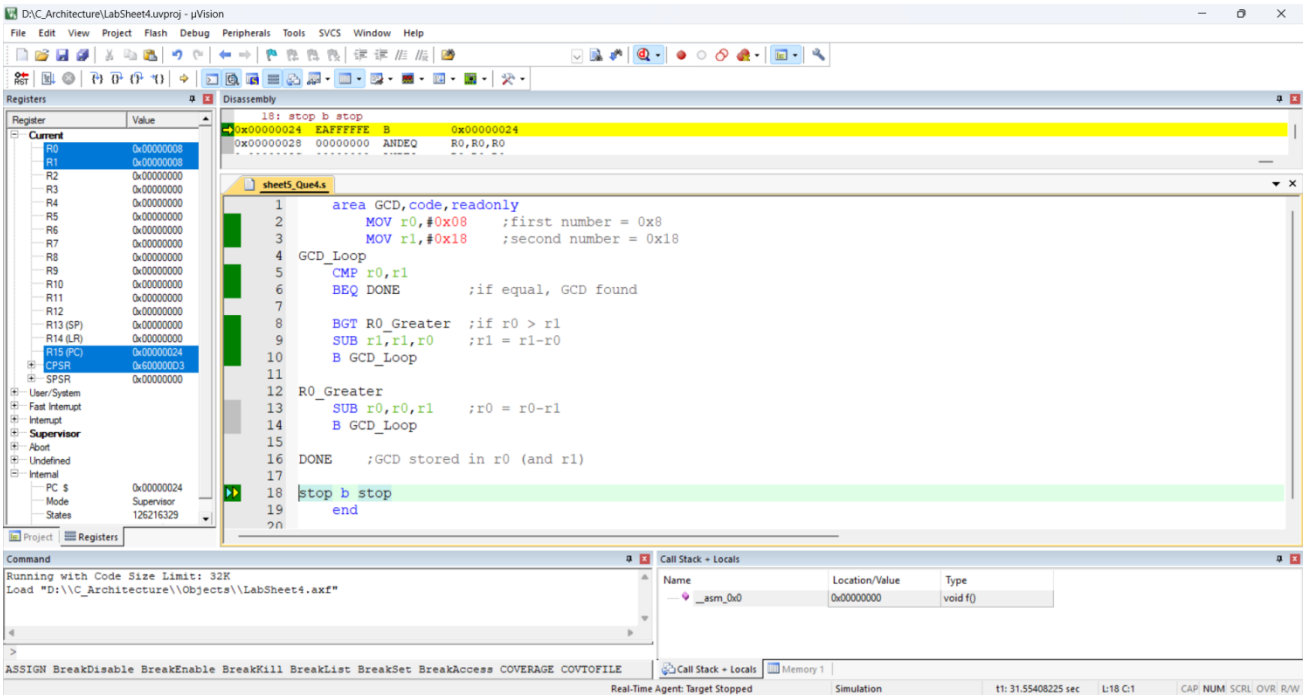

   - Decrement Hint: 5! = 5*4*3*2*1



3. Write a program to check whether a number in R0 is divisible by 5.
   If divisible, set R1 = 1; otherwise, R1 = 0




4. Write a program to find the Greatest Common Divisor (GCD) of $8_{16}$ and $18_{16}$.
   Hint: GCD is the biggest number that can divide both numbers completely.



## Discussion

This lab sheet mainly discusses how comparison operations and loop control are implemented in ARM Assembly language. Through a series of guided exercises, the lab focuses on using instructions such as CMP along with conditional and unconditional branch instructions (BEQ, BNE, BGT, BLT, and B) to make decisions and repeat instructions based on conditions. By solving problems like finding the larger number, calculating factorials using increment and decrement loops, checking divisibility by 5, and finding the GCD of two numbers, the lab helps understand how condition flags control program flow, how loops are formed using branches, and how logical decision-making works at the assembly level. Overall, the lab strengthens practical skills in writing, tracing, executing, and debugging ARM assembly programs using Keil µVision, while clearly demonstrating how comparisons and loops work together in low-level programming.

## Reference


- Lecturer's Notes.