Time: 1hr 20 min   **Maximum Marks: 60**

- -

General Instruction:

- Binary marking will be there that is 20 or 0 marks.
- If your code is successfully compiled and the desired output is produced, then only we will check for the code otherwise, 0 marks.
- Time allotted is 1hr 30 mins for solving the problems and 10 min dedicated for zipping the files and uploading them to the classroom
- No request for late submission will be entertained in any circumstances.
- Submit all your codes using the proper naming convention (read the instructions for each question) and put all files in a folder.
- Naming Convention: "Name_Roll Number.zip", and upload it.

* No .rar compression files will be accepted.

- -

Question 1: Addition of two numbers in assembly language   (Marks: 10)

Task:

Create a program to add two numbers and store the result in a register.

Description:

Write an assembly program that:

1. Defines two numbers in the data section.
2. Loads these numbers into registers.
3. Adds the numbers.
4. Stores the result in a memory location named `result`.
5. Exits the program.

Submission Guidelines:

1. **File Naming:** Save your assembly program as `addition.asm`.

- -

Question 2: Fibonacci number in assembly   (Marks: 20)

Task:

Create a program to iteratively find the nth Fibonacci number. The value for n should be set as a parameter (e.g., a programmer defined constant)

Description:

The formula for computing Fibonacci is as follows: fibonacci(n) = { n if n=0 or n=1
fibonacci(n−2) + fibonacci(n−1) if n≥2

1. Correct code
2. Use the debugger to execute the program and display the final results
3. Test the program for various values of n

Submission Guidelines:

2. **File Naming:** Save your assembly program as `fibonacci.asm`.

Question 3: Write a shell script program to print the first 5 lines of a File
(Marks: 10)

Submission Guidelines:

3. **File Naming:** Save your .sh program as `first_05_Line.sh`
4. `If file contains the fewer than 5 lines, then print also`

Question 4: Write a shell script program to count the occurrences of words in a file.
(Marks: 20)

Submission Guidelines:

5. **File Naming:** Save your .sh program as `count_word.sh`
6. `If a word is not present in the file give the proper error message`