**TASK 2**

Implementing FIR filter in inline-assembler language

Function header:

float myFIR(float insample, float \* state, const float \* coeff, const int length);

The input parameters are:

* **insample**: the current input sample
* **state**: the filter state, i.e., memory of previous input samples same length as number of coefficients
* **coeff**: FIR filter coefficients as a constant vector of length length.
* **length**: the length of the FIR filter

OUTPUT:

0. The updated state vector is 1.409000

1. The updated state vector is 1.417200

2. The updated state vector is 0.671500

3. The updated state vector is -1.207500

4. The updated state vector is 0.717200

5. The updated state vector is 1.630200

6. The updated state vector is 0.488900

7. The updated state vector is 1.034700

8. The updated state vector is 0.726900

9. The updated state vector is -0.303400

10. The updated state vector is 0.293900

11. The updated state vector is -0.787300

12. The updated state vector is 0.888400

13. The updated state vector is -1.147100

14. The updated state vector is -1.068900

15. The updated state vector is 5.000000

------------------------------------

0 The filter output is 45.847505

1 The filter output is -2.580785

2 The filter output is 1.134516

3 The filter output is -0.878654

4 The filter output is -0.810566

5 The filter output is -0.037453

6 The filter output is 0.031537

7 The filter output is 1.890768

8 The filter output is 2.964931

9 The filter output is -0.322657

10 The filter output is 8.065405

11 The filter output is 0.373128

12 The filter output is -0.092003

13 The filter output is 0.322267

14 The filter output is -0.411733

15 The filter output is -0.246374

------------------------------------

The final result is 55.249830.

**Conclusion:** Final result is obtained and are verified with MATLAB. Importing DSP values into MATLAB for verification process.