**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

**4th Semester, Academic Year 2020-21**

Date: 10-02-2021

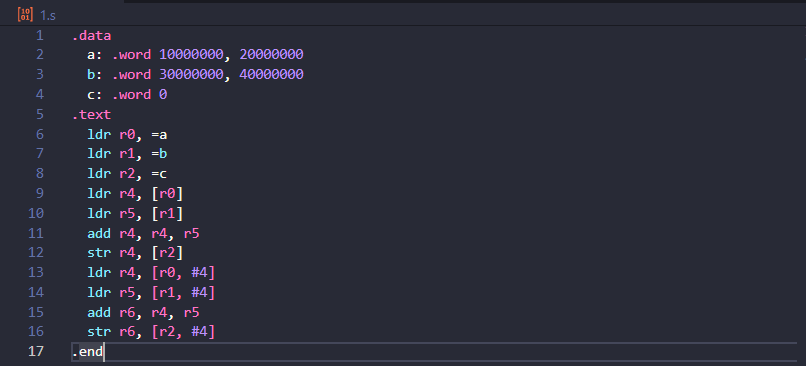
|  |  |  |
| --- | --- | --- |
| **Name:**  Atul Anurag | **SRN:**  PES2UG19CS075 | **Section:**  B |

Week#3

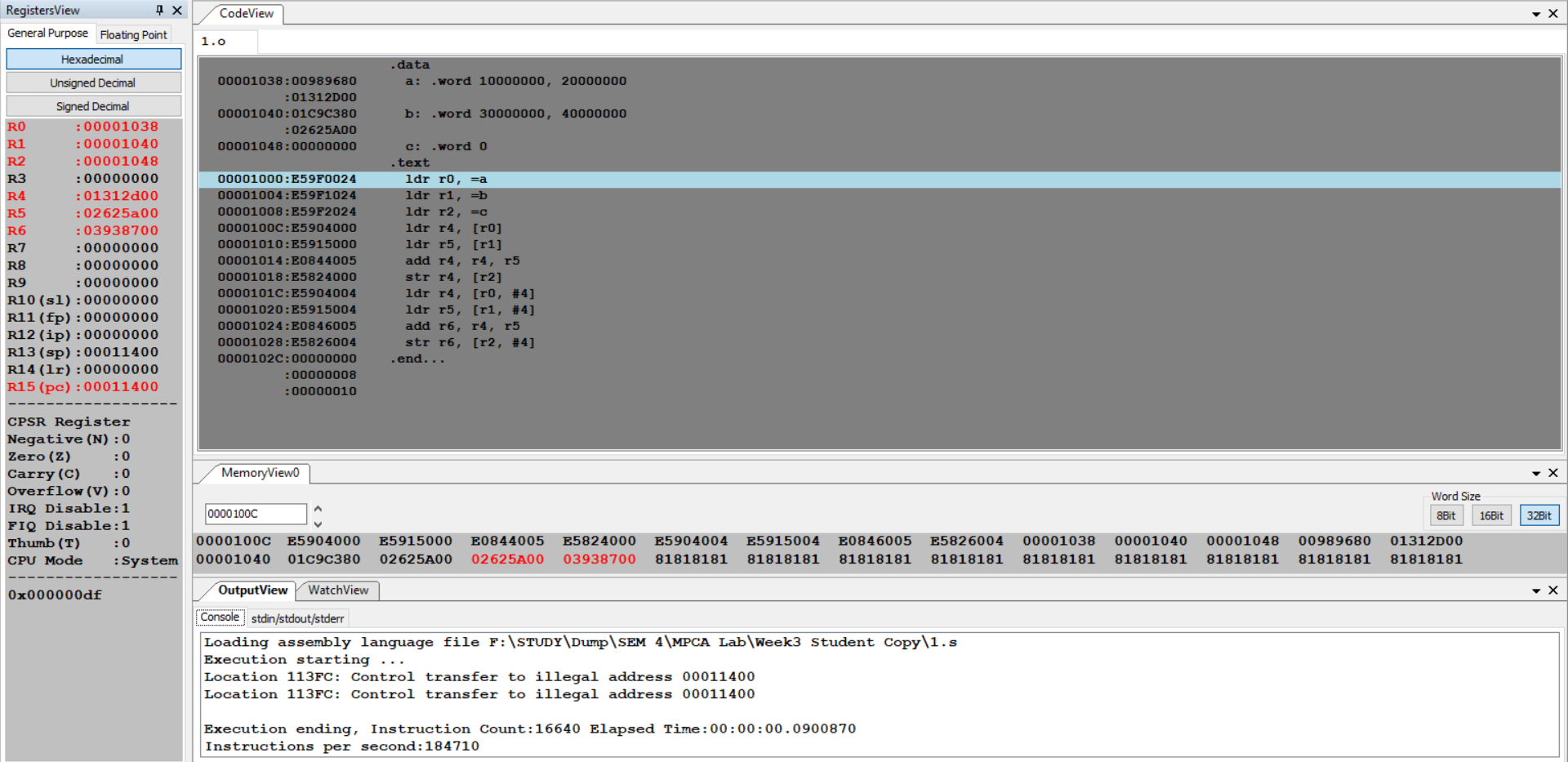
Program Number: 1

**Write an ALP to add two 64-bit numbers loaded from memory and store the result in memory.**

1. ARM Assembly Code for the program..



1. Output Screen Shot (One Example of your choice)



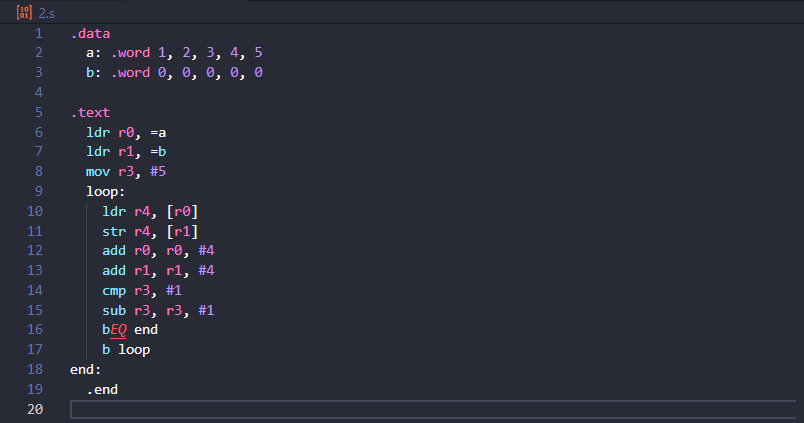
1. Output Table for the program

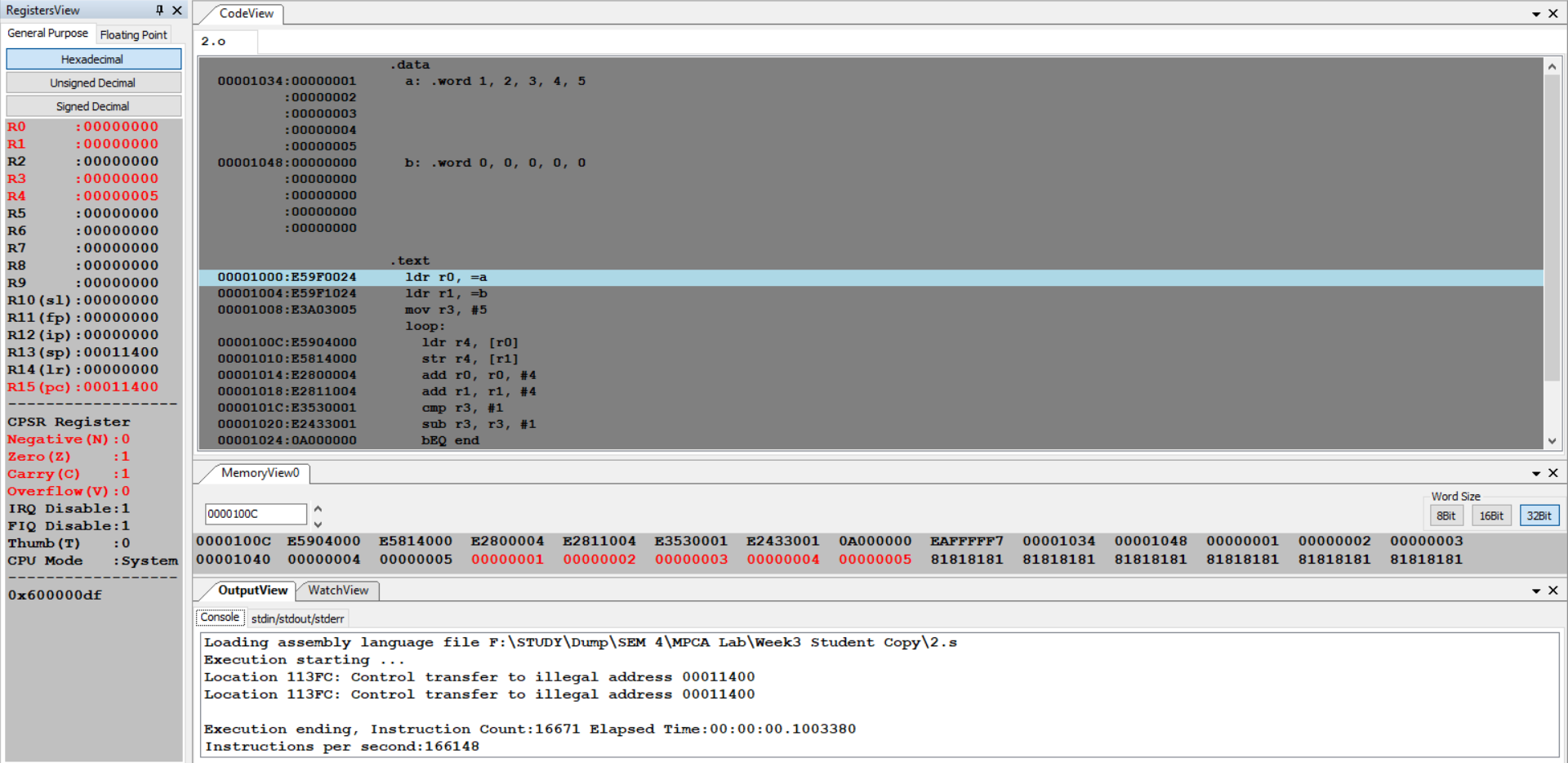
|  |  |  |
| --- | --- | --- |
| a: .word 10000000, 20000000  b: .word 30000000, 40000000 | | |
|  | Upper 32 bits | **Lower 32 bits** |
| **a: .word** | 20000000  **(00989680)** | 10000000  **(01312D00)** |
| **b: .word** | 40000000  (**01C9C380)** | 30000000  **(02625A00)** |
| **c: .word** | 60000000  **(02625A00)** | 40000000  **(03938700)** |

Program Number: 2

**Write an ALP to copy n numbers from Memory Location A to Memory Location B**

1. ARM Assembly Code for the program.



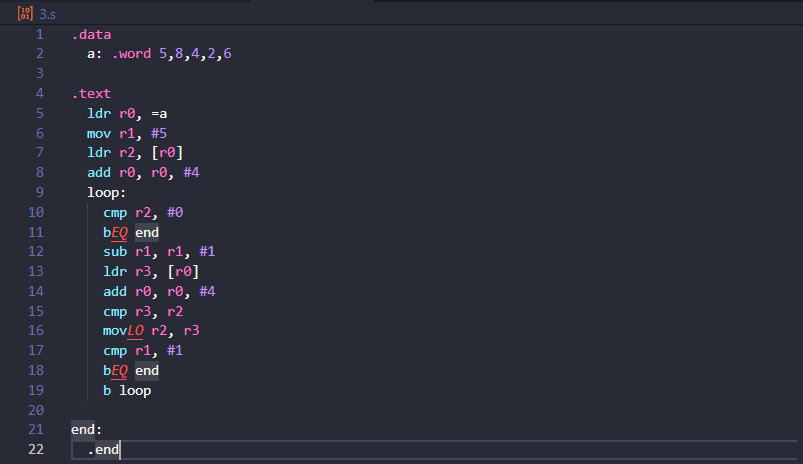
1. Output Screen Shot (One Example of your choice) 
2. Output Table for the program

|  |  |
| --- | --- |
| .data  a: .word 1, 2, 3, 4, 5  b: .word 0, 0, 0, 0, 0 | |
| 1st Iteration | a: .word 1, 2, 3, 4, 5  b: .word 1, 0, 0, 0,0 |
| 2nd Iteration | a: .word 1, 2, 3, 4, 5  b: .word 1, 2, 0, 0, 0 |
| 3rd Iteration | a: .word 1, 2, 3, 4, 5  b: .word 1, 2, 3, 0, 0 |
| 4th Iteration | a: .word 1, 2, 3, 4, 5  b: .word 1, 2, 3, 4, 0 |
| 5th Iteration | a: .word 1, 2, 3, 4, 5  b: .word 1, 2, 3, 4, 5 |

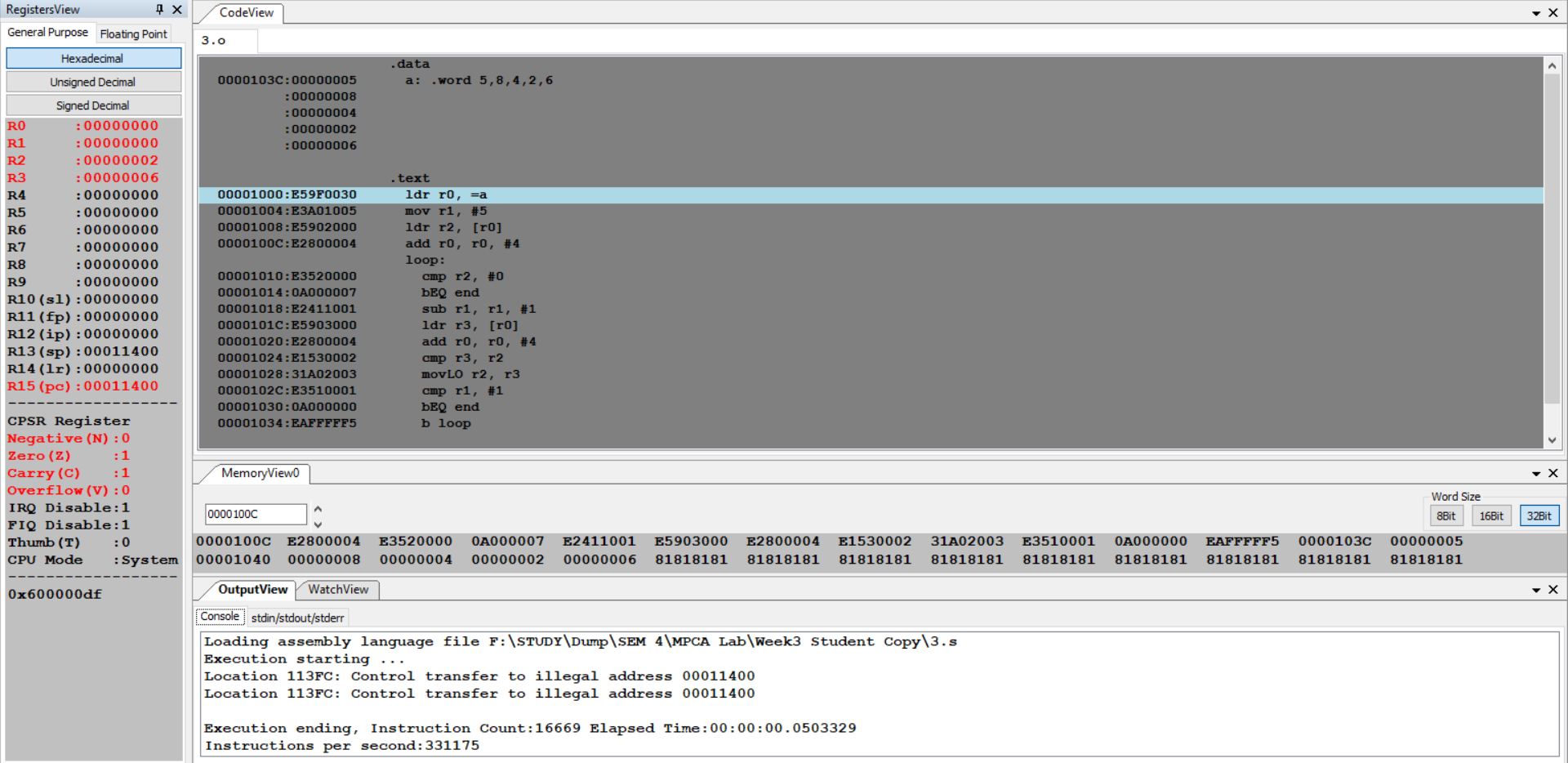
Program Number: 3

**Write an ALP to find smallest number in an array of n 32-bit numbers**

1. ARM Assembly Code for the program.



1. Output Screen Shot (One Example of your choice)



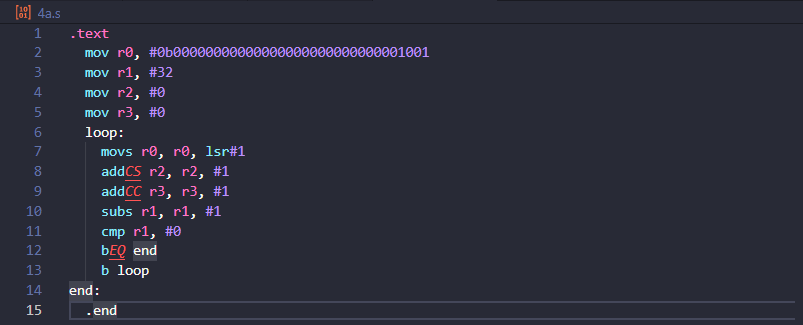
1. Output Table for the program

|  |  |
| --- | --- |
| a: .word 5, 8, 4, 2, 6 | |
| 1st Iteration | R2=5, R3=8 (R3>R2) |
| 2nd Iteration | R2=5, R3=4 (R3<R2) |
| 3rd Iteration | R2=4, R3=2 (R3<R2) |
| 4th Iteration | R2=2, R3=6 (R3>R2) |
| Smallest number 2 is present in R2 | |

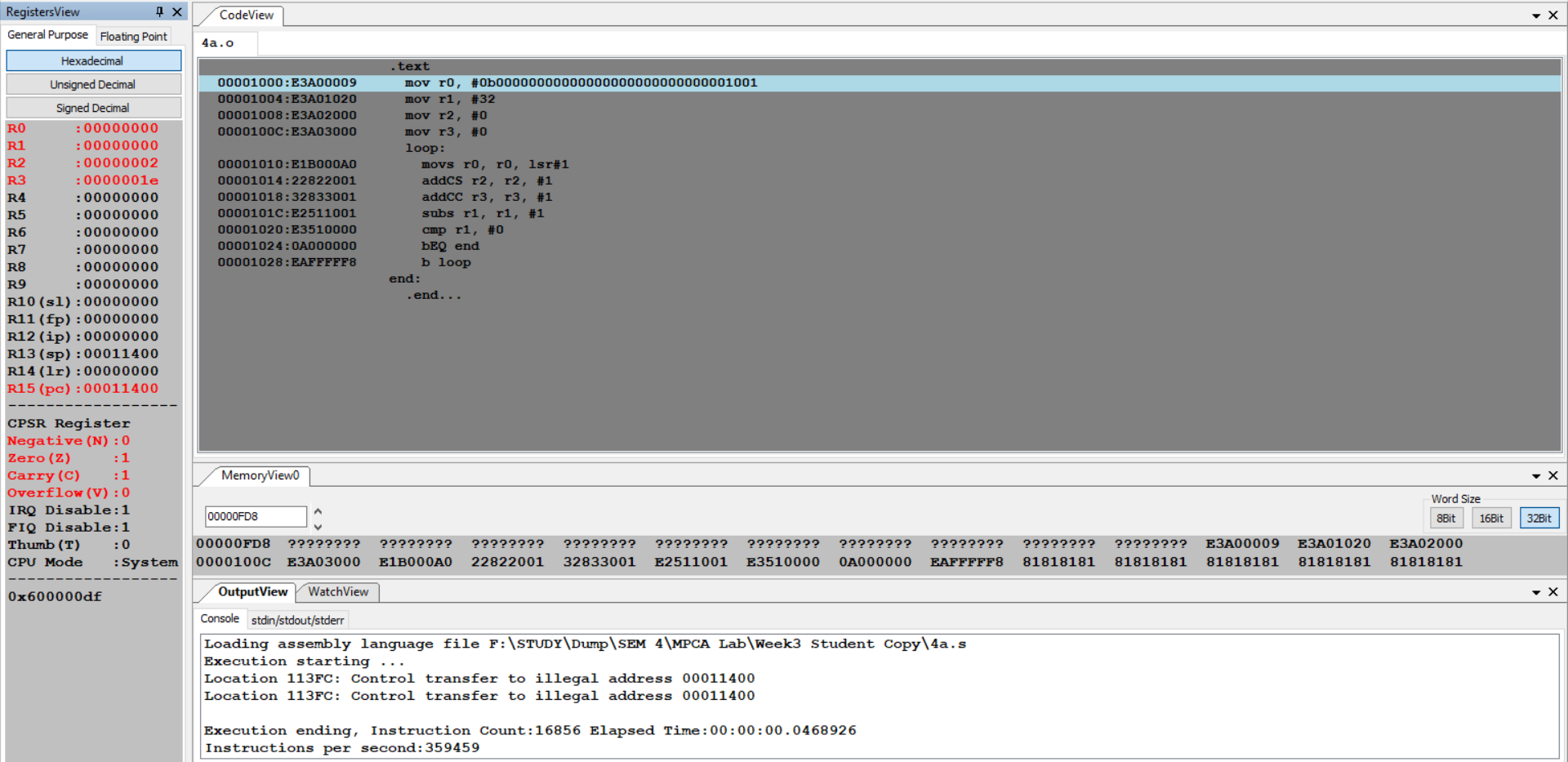
Program Number: 4a

**Write an ALP to count the number of 1’s and 0’s in a given 32-bit number.**

1. ARM Assembly Code for the program.



1. Output Screen Shot (One Example of your choice)



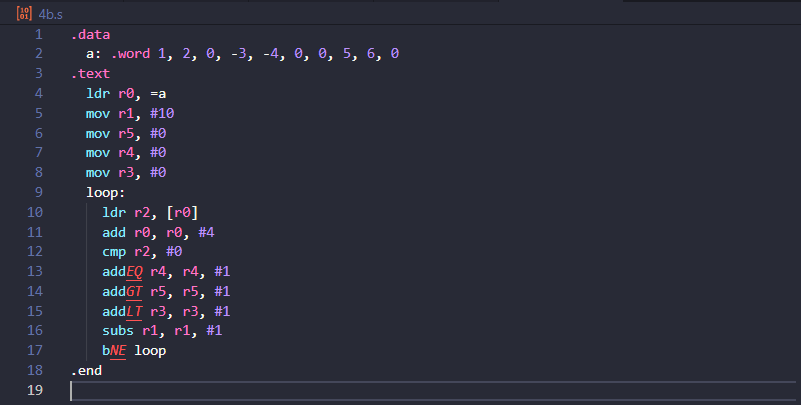
1. Output Table for the program

|  |  |  |
| --- | --- | --- |
| r0= 0b00000000000000000000000000001001 | | |
| **r1** | **32** |  |
| **r2** | **After execution** | **2 (=2 in hex)** |
| **r3** | **After execution** | **30 (=1e in hex)** |

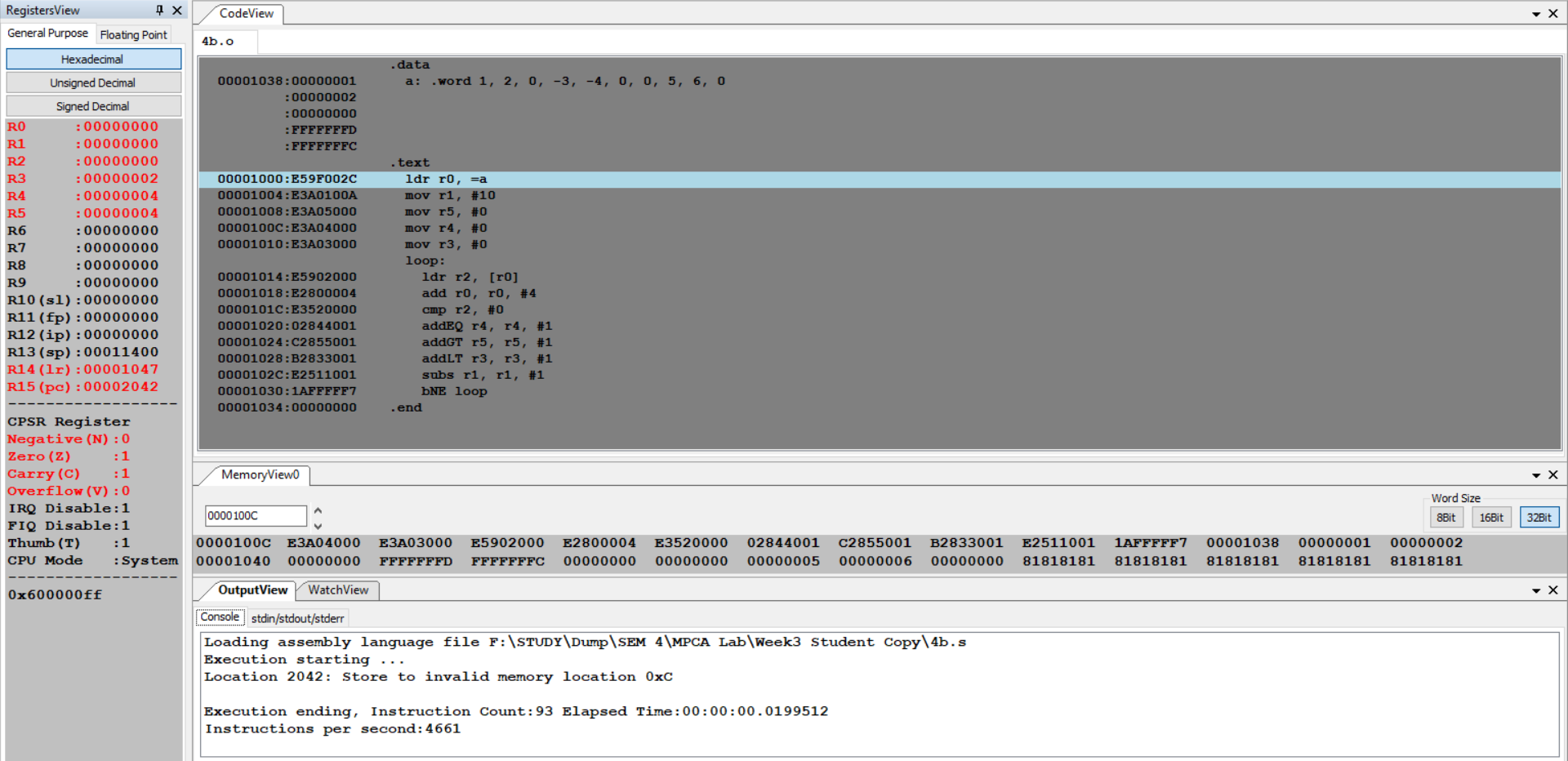
Program Number: 4b

**Write an ALP to find the number of zeroes, positive and negative numbers in a given array**

1. ARM Assembly Code for the program.



1. Output Screen Shot (One Example of your choice)



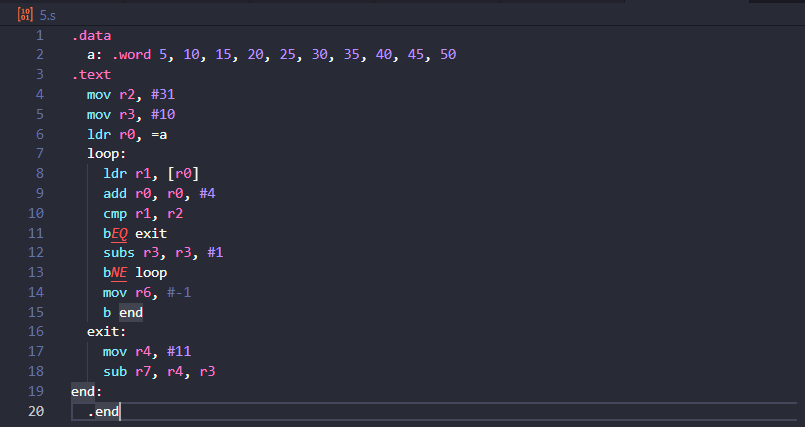
1. Output Table for the program

|  |  |
| --- | --- |
| a:.word 1, 2, 0, -3, -4, 0, 0, 5, 6, 0 | |
| **R3** | **2** |
| **R4** | **4** |
| **R5** | **4** |

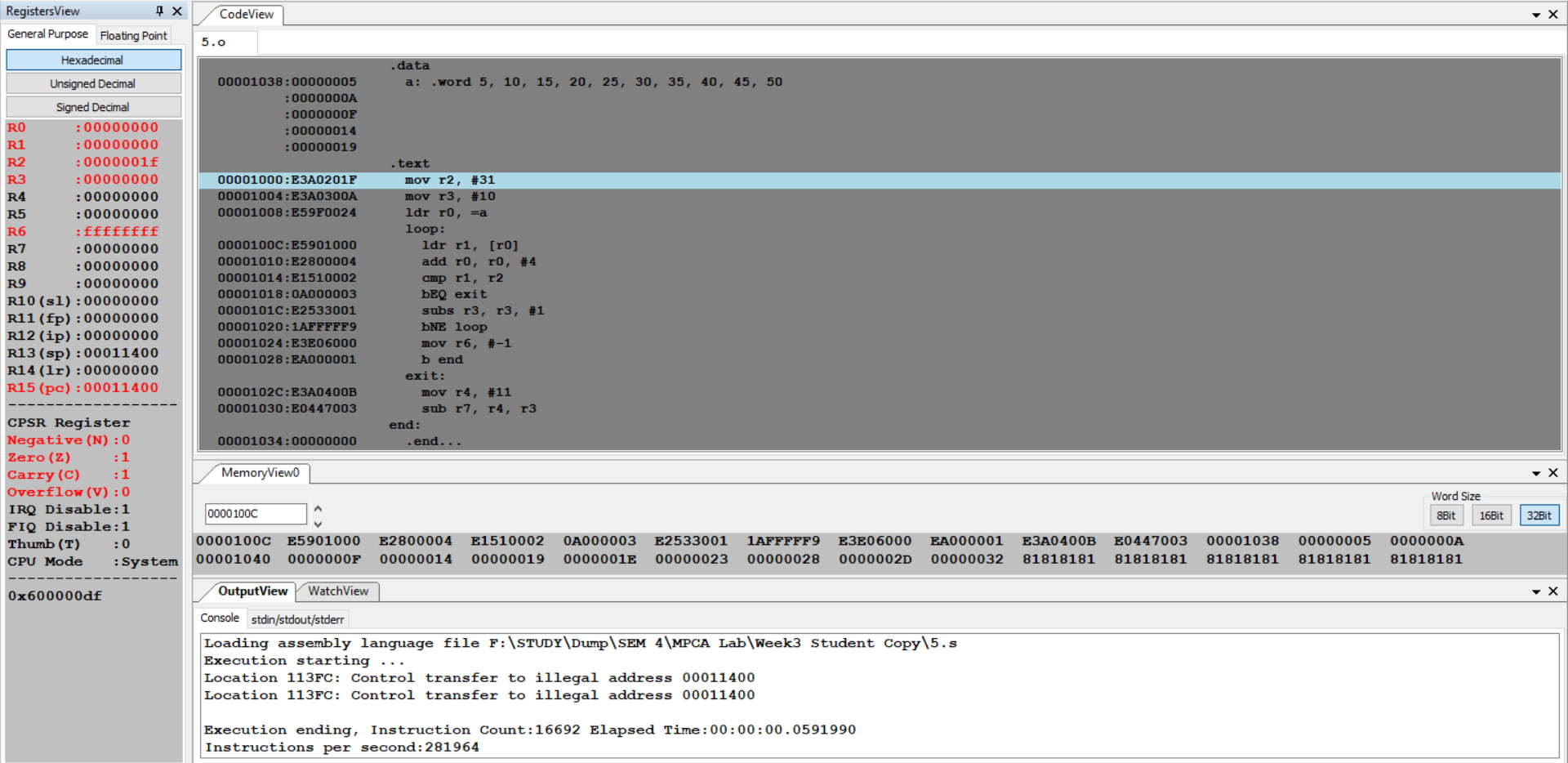
Program Number: 5

**Write an ALP to check whether a given number is present in array using Linear Search (Without SWI 0x02), if found move +1 to R6 and key position to R7 else move -1 to R6 (if number not found)**

1. ARM Assembly Code for the program.



1. Output Screen Shot (One Example of your choice)



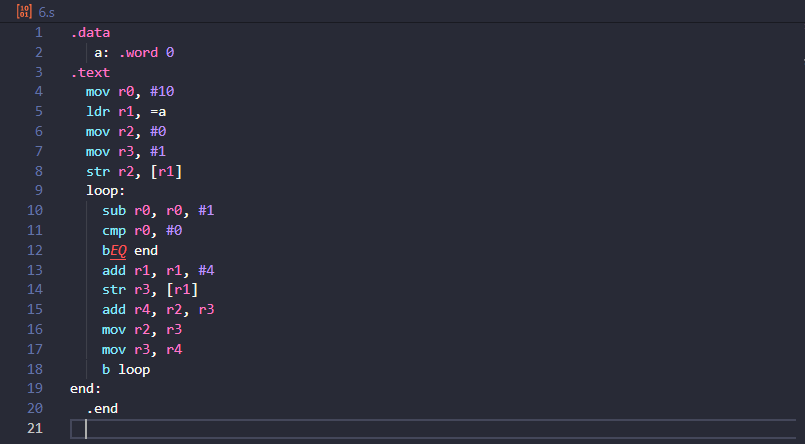
1. Output Table for the program

|  |  |  |
| --- | --- | --- |
| A:.WORD 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 | | |
|  |  | **HEX value** |
| **R2** | **KEY =31** | **1F** |
| **R3** | **COUNT =10** |  |
| **R0** | **Address of A** | **00001038** |
| **R3** | **After Execution =0**  **R6= -1**  **(key not present)** |  |

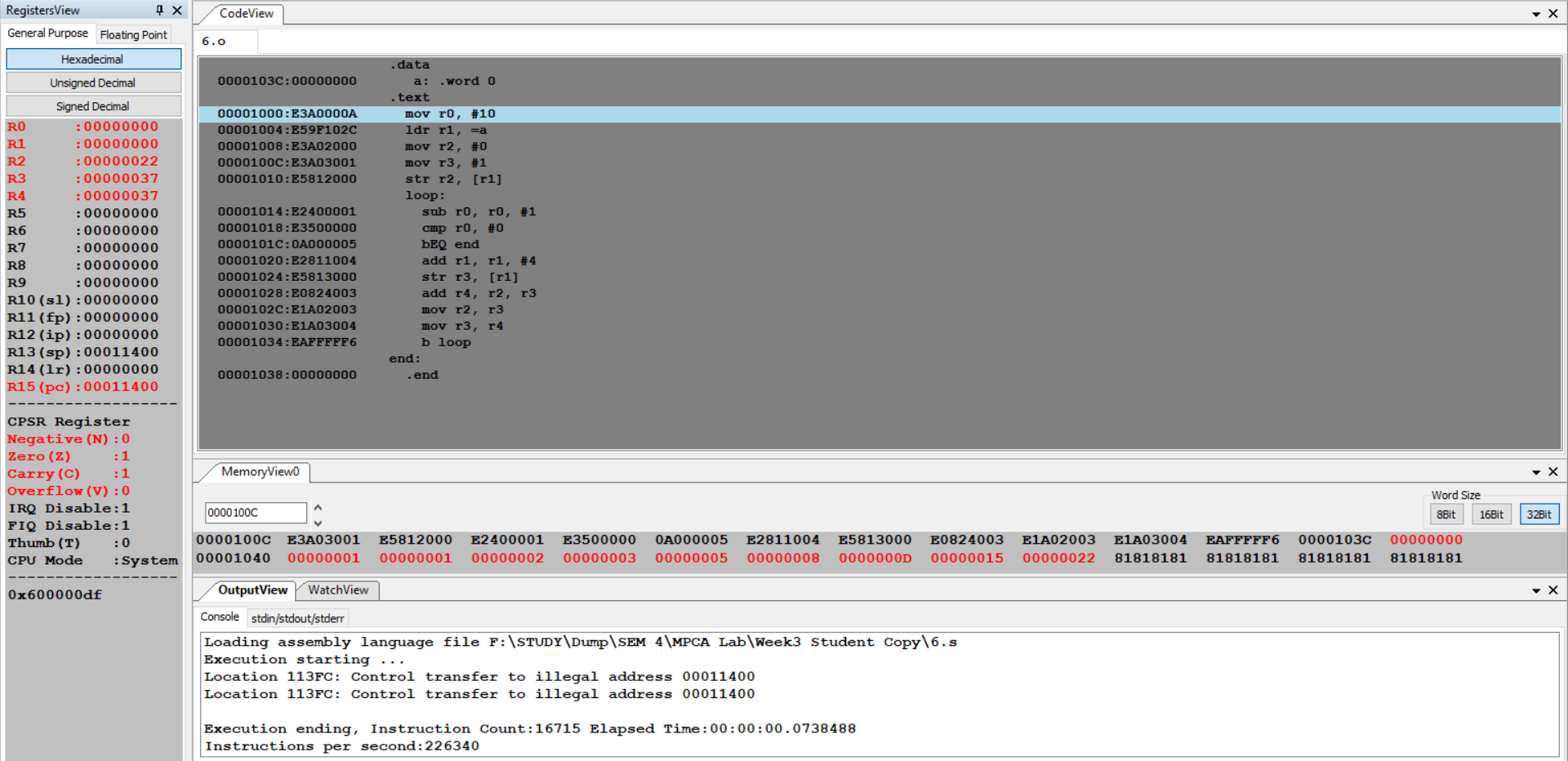
Program Number: 6

**Write an ALP to generate Fibonacci Series and store them in an array**

1. ARM Assembly Code for the program.



1. Output Screen Shot (One Example of your choice)



1. Output Table for the program

|  |  |  |
| --- | --- | --- |
| FIBONACCI SEQUENCE | | |
| **R0** | **Fibonacci Count** | **10** |
| **R1** | **Address of A** | **0000103C** |
| **R2** | **Initially 0** | **0** |
| **R3** | **Initially 1** | **1** |
| **R4** | **1st Iteration** | **0 + 1 = 1** |
| **R4** | **2nd Iteration** | **1 + 1 = 2** |
| **R4** | **3rd Iteration** | **2 + 1 = 3** |
| **R4** | **4th Iteration** | **3 + 2 = 5** |
| **R4** | **5th Iteration** | **5 + 3 = 8** |
| **R4** | **6th Iteration** | **8 + 5 = 13= 0000000D** |
| **R4** | **7th Iteration** | **13 + 8 = 21= 00000015** |
| **R4** | **8th Iteration** | **21 + 13 = 34 = 00000022** |

**Disclaimer:**

* The programs and output submitted is duly written, verified and executed by me.
* I have not copied from any of my peers nor from the external resource such as internet.
* If found plagiarized, I will abide with the disciplinary action of the University.

Signature: Atul Anurag

Name: Atul Anurag

SRN: PES2UG19CS075

Section: B

Date: 10-02-21