**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

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

Date: 12/02/2021

|  |  |  |
| --- | --- | --- |
| Name: Pranav R. Hegde | SRN: PES1UG19CS343 | Section: F |

Week#\_\_\_\_3\_\_\_\_\_\_\_ Program Number: \_\_\_1\_\_\_

Title of the Program

1. ARM Assembly Code for each program
2. Output Screen Shot

Problem statement:

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

ARM assembly code:

.text

    ldr R0, =P

    ldr R1, =Q

    ldr R2, [R0], #4

    ldr R3, [R0]

    ldr R4, [R1], #4

    ldr R5, [R1]

    adds R7, R3, R5

    adc R6, R2, R4

    ldr R8, =R

    str R6, [R8], #4

    str R7, [R8]

    ldr R9, =R

    ldr R10, [R9, #4]

    ldr R11, [R9]

    swi 0x11

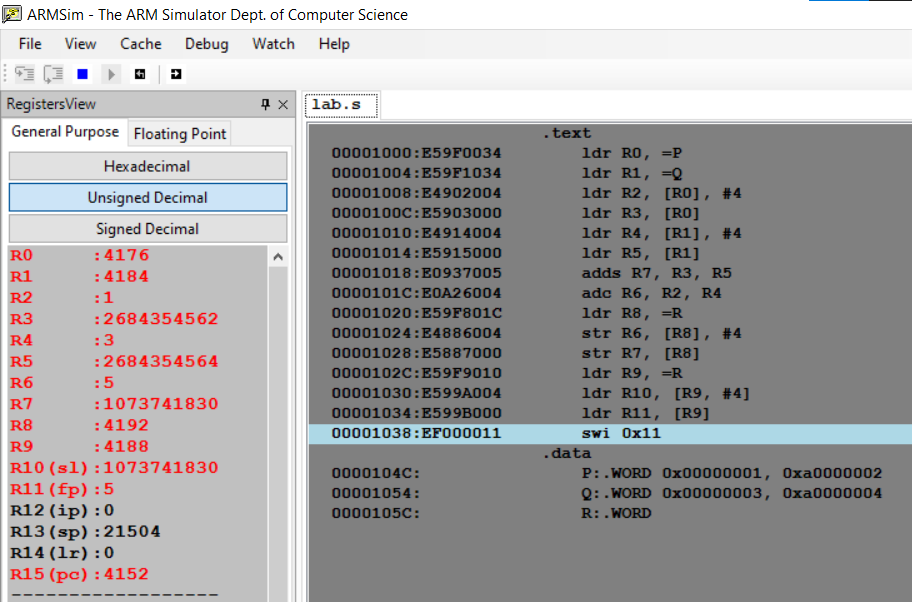
.data

    P:.WORD 0x00000001, 0xa0000002

    Q:.WORD 0x00000003, 0xa0000004

    R:.WORD

Output:



**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

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

Date: 12/02/2021

|  |  |  |
| --- | --- | --- |
| Name: Pranav R. Hegde | SRN: PES1UG19CS343 | Section: F |

Week#\_\_\_\_3\_\_\_\_\_\_\_ Program Number: \_\_\_2\_\_\_

Title of the Program

1. ARM Assembly Code for each program
2. Output Screen Shot

Problem statement:

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

Code:

.text

    ldr R0, =P

    ldr R1, =Q

    mov R2, #10

    L1:

        ldr R3, [R0]

        str R3, [R1]

        sub R2, R2, #1

        add R0, R0, #4

        add R1, R1, #4

    cmp R2, #0

    bne L1

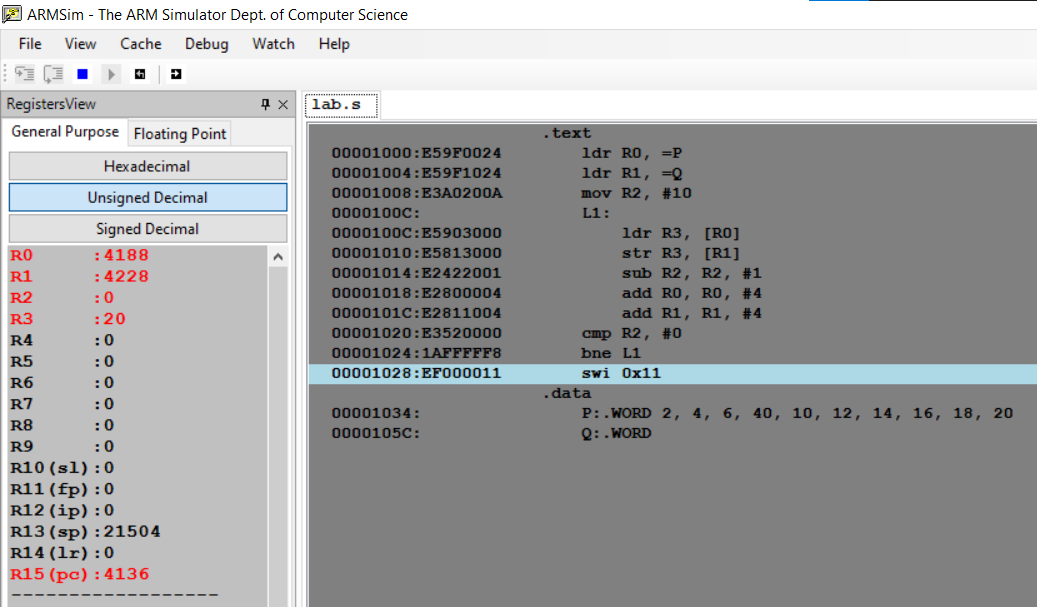
    swi 0x11

.data

    P:.WORD 2, 4, 6, 40, 10, 12, 14, 16, 18, 20

    Q:.WORD

Output:



**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

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

Date: 12/02/2021

|  |  |  |
| --- | --- | --- |
| Name: Pranav R. Hegde | SRN: PES1UG19CS343 | Section: F |

Week#\_\_\_\_3\_\_\_\_\_\_\_ Program Number: \_\_\_3\_\_\_

Title of the Program

1. ARM Assembly Code for each program
2. Output Screen Shot

Problem statement:

3. Write an ALP to find smallest number in an array of n – 32 bit numbers.

Code:

.text

    ldr R0,=A

    ldr R1,=B

    ldr R4,[R0],#4

    ldr R3,[R1]

    beq L1

    sub R3,R3,#1

    mov R6,R4 ; Smallest number to be stored in R6

    L2:

        ldr R4,[R0],#4

        cmp R4,R6

        movlt R6,R4

        sub R3,R3,#1

    cmp R3,#0

    bne L2

    L1:

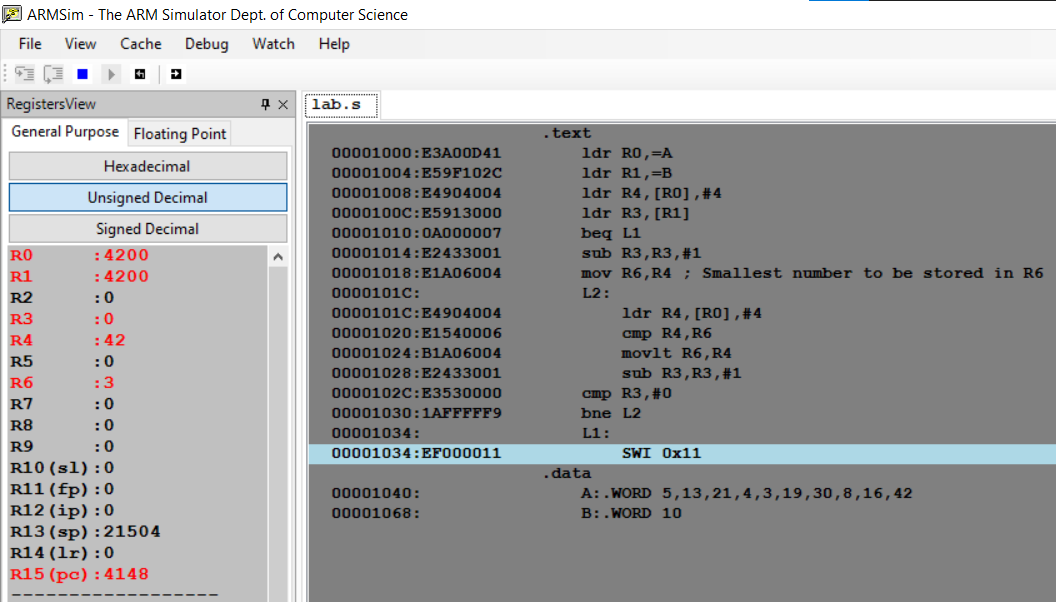
        SWI 0x11

.data

    A:.WORD 5,13,21,4,3,19,30,8,16,42

    B:.WORD 10

Output:



**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

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

Date: 12/02/2021

|  |  |  |
| --- | --- | --- |
| Name: Pranav R. Hegde | SRN: PES1UG19CS343 | Section: F |

Week#\_\_\_\_3\_\_\_\_\_\_\_ Program Number: \_\_\_4a\_\_\_\_

Title of the Program

1. ARM Assembly Code for each program
2. Output Screen Shot

Problem statement:

4a. Write an ALP to count the number of 1’s and 0’s in a given 32 bit number.

Code:

.text

    ldr R0,=0b11110010100101001101001100110111

    mov R1,#32

    mov R5,#0

    mov R6,#0

    L1:

        and R2,R0,#1

    cmp R2,#1

    addeq R5,R5,#1 ;Number of 1's is stored in R5

    addne R6,R6,#1 ;Number of 0's is stored in R6

    mov R0,R0,LSR #1

    sub R1,R1,#1

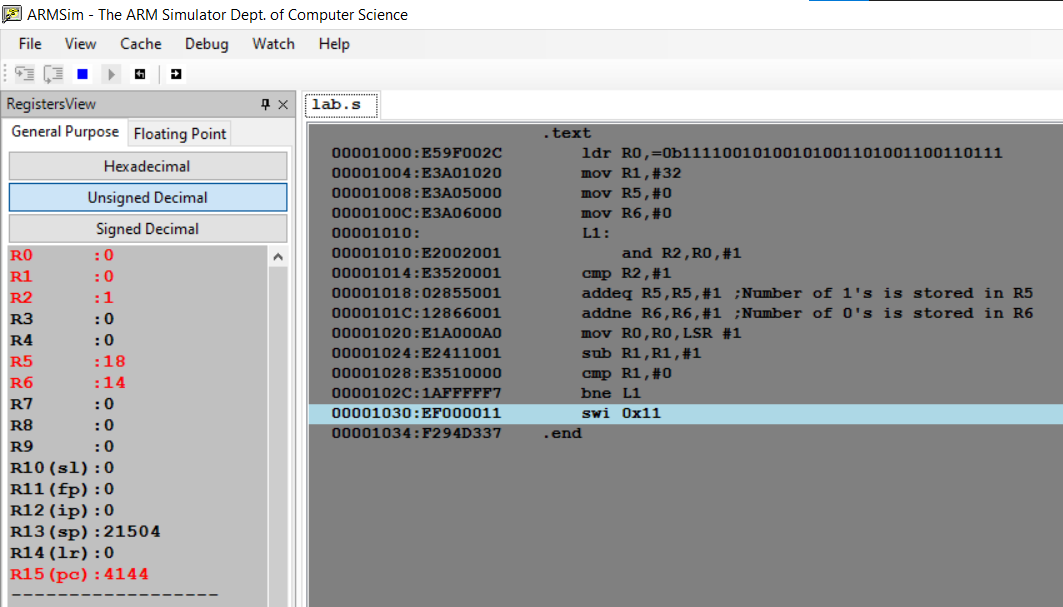
    cmp R1,#0

    bne L1

    swi 0x11

.end

Output:



**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

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

Date: 12/02/2021

|  |  |  |
| --- | --- | --- |
| Name: Pranav R. Hegde | SRN: PES1UG19CS343 | Section: F |

Week#\_\_\_\_3\_\_\_\_\_\_\_ Program Number: \_\_\_4b\_\_\_

Title of the Program

1. ARM Assembly Code for each program
2. Output Screen Shot

Problem statement:

4b. Write an ALP to find the number of zeroes, positive and negative numbers in a given array.

Code:

.text

    ldr R1,=A

    ldr R2,=B

    ldr R3,[R2]

    mov R7,#0

    mov R8,#0

    mov R9,#0

    L1:

        ldr R4,[R1],#4

    cmp R4,#0

    addeq R7,R7,#1 ;Number of 0's is stored in R7

    addlt R8,R8,#1 ;Number of negative numbers is stored in R8

    addgt R9,R9,#1 ;Number of positive numbers is stored in R9

    sub R3,R3,#1

    cmp R3,#0

    bne L1

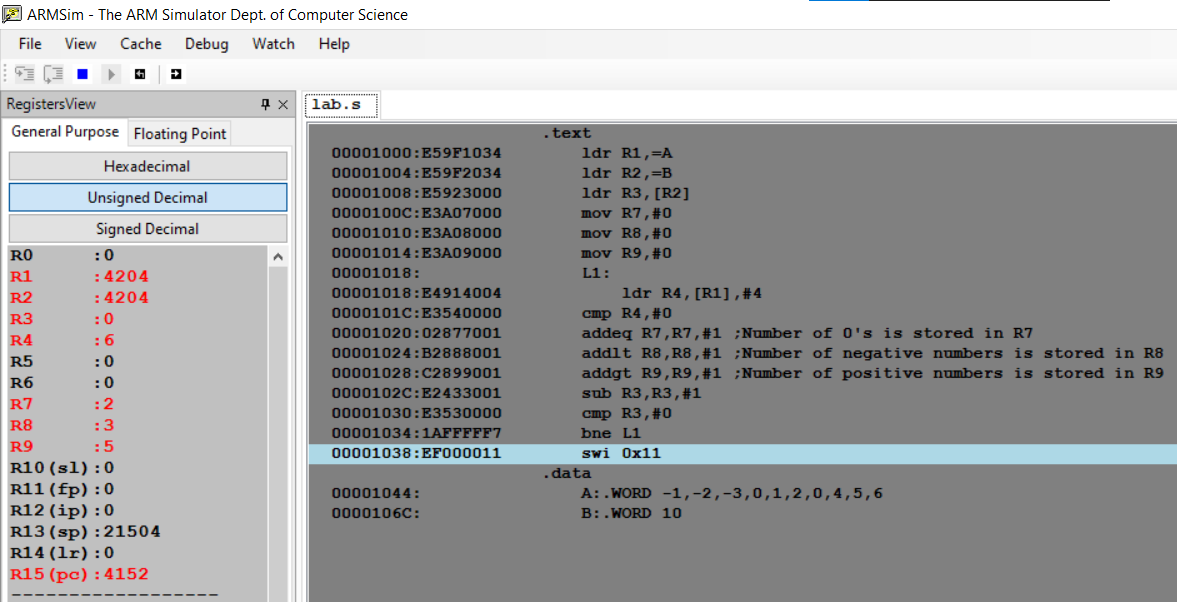
    swi 0x11

.data

    A:.WORD -1,-2,-3,0,1,2,0,4,5,6

    B:.WORD 10

Output:



**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

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

Date: 12/02/2021

|  |  |  |
| --- | --- | --- |
| Name: Pranav R. Hegde | SRN: PES1UG19CS343 | Section: F |

Week#\_\_\_\_3\_\_\_\_\_\_\_ Program Number: \_\_\_5\_\_\_

Title of the Program

1. ARM Assembly Code for each program
2. Output Screen Shot

Problem statement:

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).

Code:

.text

    ldr R0,=A

    mov R1,#10

    mov R3,R1

    L1:

        ldr R2,[R0],#4

    cmp R2,#16 ;Element 16 is to be searched in the array

    beq L2

    subs R1,R1,#1

    cmp R1,#0

    bne L1

    mov R6,#-1

    swi 0x11

    L2:

        mov R6,#1

        sub R1,R1,#1

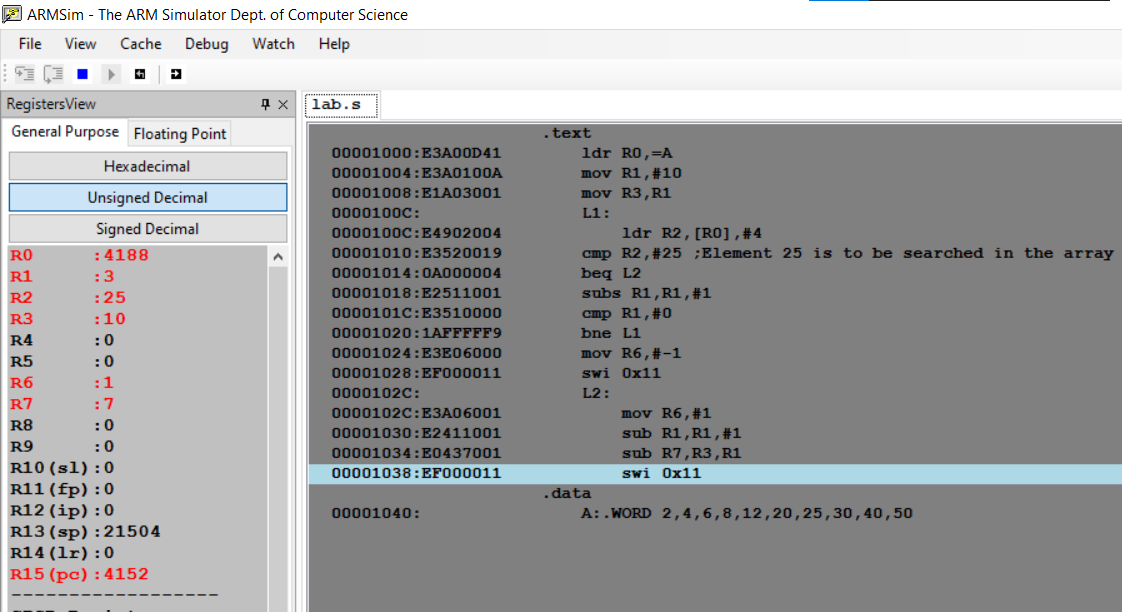
        sub R7,R3,R1

        swi 0x11

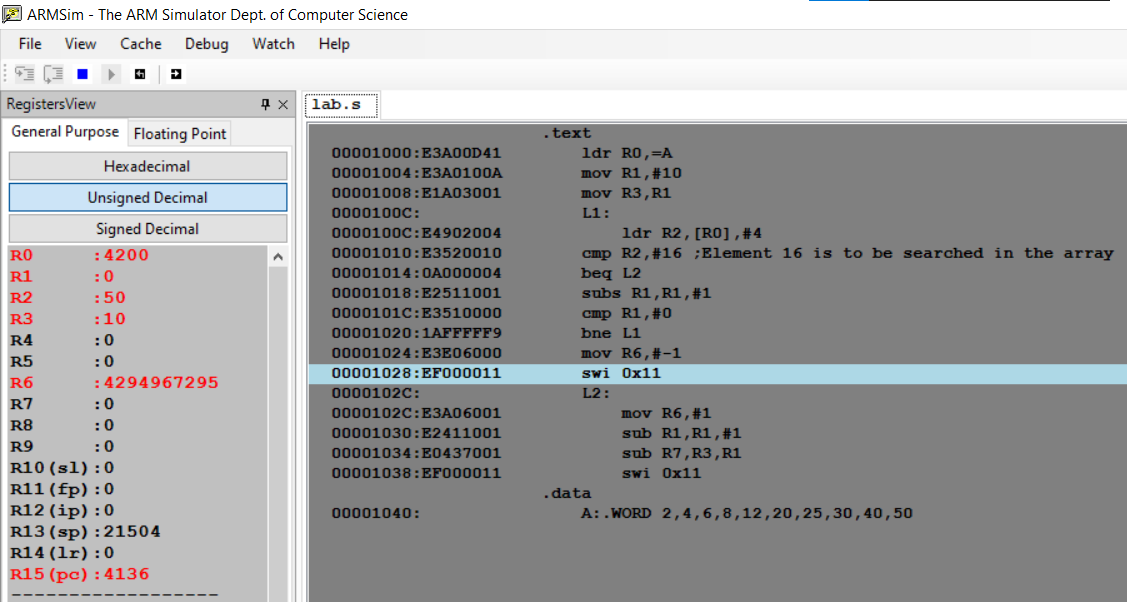
.data

    A:.WORD 2,4,6,8,12,20,25,30,40,50

Case1: Element found



Case2: Element NOT found



**Microprocessor and Computer Architecture Laboratory**

**UE19CS256**

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

Date: 12/02/2021

|  |  |  |
| --- | --- | --- |
| Name: Pranav R. Hegde | SRN: PES1UG19CS343 | Section: F |

Week#\_\_\_\_3\_\_\_\_\_\_\_ Program Number: \_\_\_6\_\_\_

Title of the Program

1. ARM Assembly Code for each program
2. Output Screen Shot

Problem statement:

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

Code:

.text

    ldr R1,=A

    mov R2,#0

    str R2,[R1]

    add R1,R1,#4

    mov R3,#1

    str R3,[R1]

    mov R5,#8 ; 8 fibonacci numbers will be stored after 0 and 1(So,total=10) in the array

    L1:

        add R4,R2,R3

        add R1,R1,#4

        str R4,[R1]

        mov R2,R3

        mov R3,R4

        subs R5,R5,#1

        mov R4,#0

        bne L1

    swi 0x11

.data

    A:.WORD

Output:

