Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date:

Name: A Narendiran	SRN: PES1UG19CS001	Section A

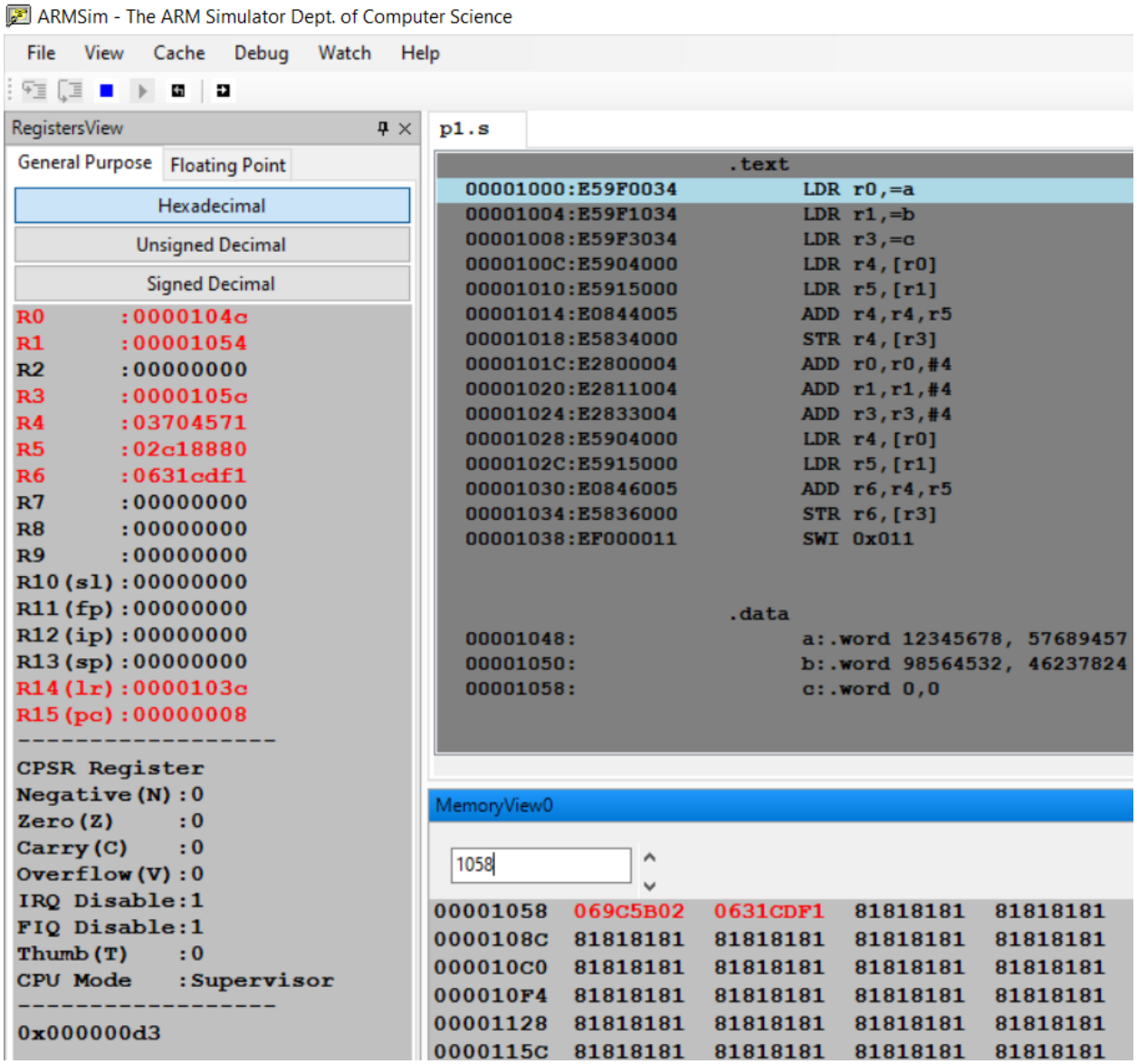
Week#_____3_____ Program Number: ___1___

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

ARM Assembly Code for the program

```
text
      LDR r0,=a
      LDR r1,=b
      LDR r3,=c
      LDR r4,[r0]
      LDR r5,[r1]
      ADD r4,r4,r5
      STR r4,[r3]
      ADD r0,r0,#4
      ADD r1,r1,#4
      ADD r3,r3,#4
      LDR r4,[r0]
      LDR r5,[r1]
      ADD r6,r4,r5
      STR r6,[r3]
      SWI 0x011
.data
      a:.word 12345678, 57689457
      b:.word 98564532, 46237824
      c:.word 0,0
```

Output Screen Shot



Week#____3_____ Program Number: ___2___

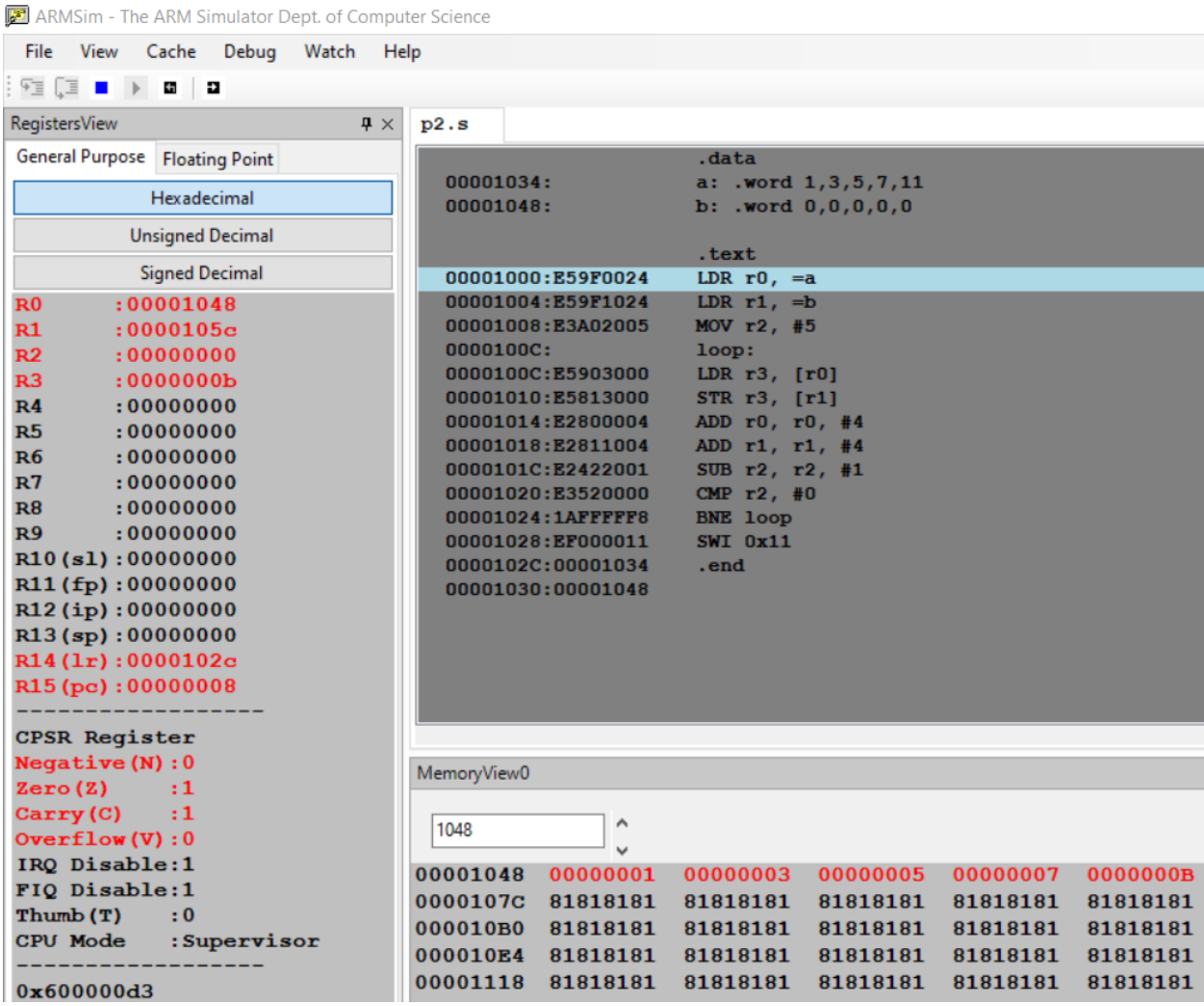
2.Write an ALP to copy n numbers from Memory Location A to Memory Location B

ARM Assembly Code for the program

```
.data
a: .word 1,3,5,7,11
b: .word 0,0,0,0,0

.text
LDR r0, =a
LDR r1, =b
MOV r2, #5
loop:
LDR r3, [r0]
STR r3, [r1]
ADD r0, r0, #4
ADD r1, r1, #4
SUB r2, r2, #1
CMP r2, #0
BNE loop
SWI 0x11
.end
```

Output Screen Shot



Week#____3_____ Program Number: __3____

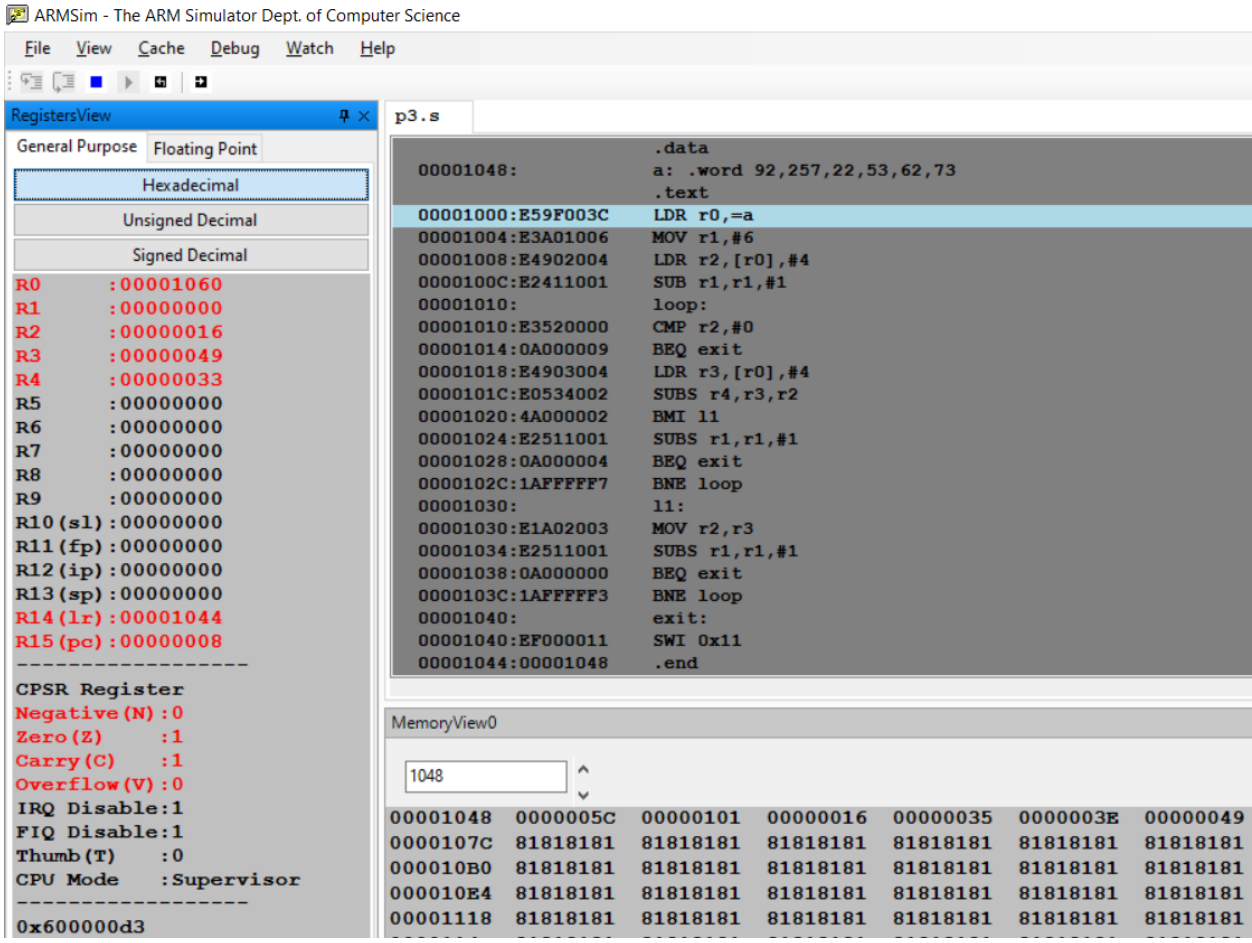
3. Write an ALP to find smallest number in an array of n - 32 bit numbers

ARM Assembly Code for the program

```
.data
a: .word 92,257,22,53,62,73
.text
LDR r0,=a
MOV r1,#6
LDR r2,[r0],#4
SUB r1,r1,#1
```

```
loop:
CMP r2,#0
BEQ exit
LDR r3,[r0],#4
SUBS r4,r3,r2
BMI l1
SUBS r1,r1,#1
BEQ exit
BNE loop
l1:
MOV r2,r3
SUBS r1,r1,#1
BEQ exit
BNE loop
exit:
SWI 0x11
.end
```

Output Screen Shot



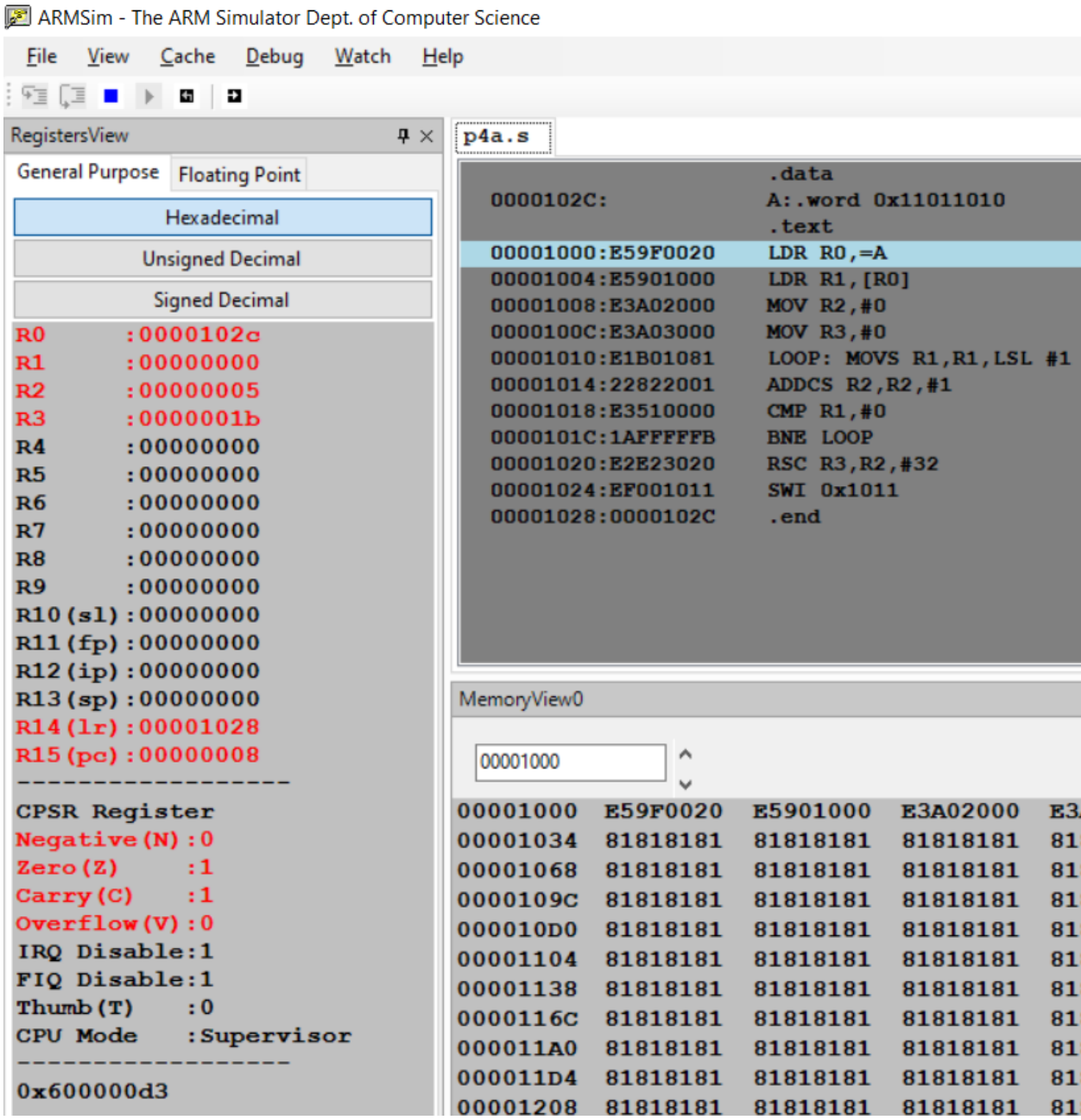
Week#____3_____ Program Number: __4a__

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

ARM Assembly Code for the program

```
.data
A:.word 0x11011010
.text
LDR R0,=A
LDR R1,[R0]
MOV R2,#0
MOV R3,#0
LOOP: MOVS R1,R1,LSL #1
ADDCS R2,R2,#1
CMP R1,#0
BNE LOOP
RSC R3,R2,#32
SWI 0x1011
.end
```

Output Screen Shot



Week#____3_____ Program Number: ___4b___

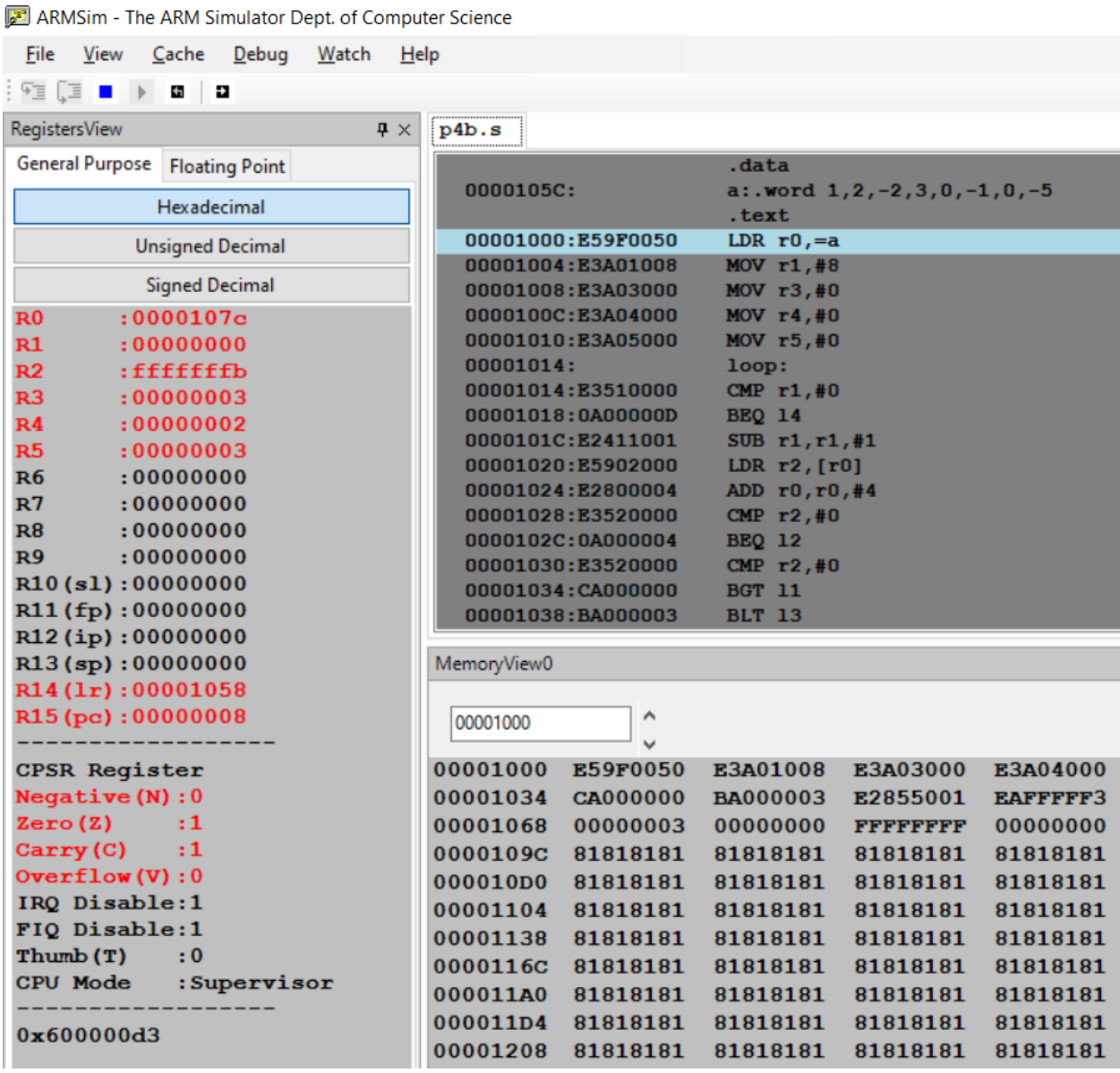
4 b) Write an ALP to find the number of zeroes, positive and negative numbers in a given array.

ARM Assembly Code for the program

```
.data
a:.word 1,2,-2,3,0,-1,0,-5
.text
```

```
LDR r0,=a
MOV r1,#8
MOV r3,#0
MOV r4,#0
MOV r5,#0
loop:
CMP r1,#0
BEQ l4
SUB r1,r1,#1
LDR r2,[r0]
ADD r0,r0,#4
CMP r2,#0
BEQ l2
CMP r2,#0
BGT l1
BLT l3
l1:
ADD r5,r5,#1
B loop
l2:
ADD r4,r4,#1
B loop
l3:
ADD r3,r3,#1
B loop
l4:
SWI 0x11
.end
```

Output Screen Shot



Week#____3_____ Program Number: __5__

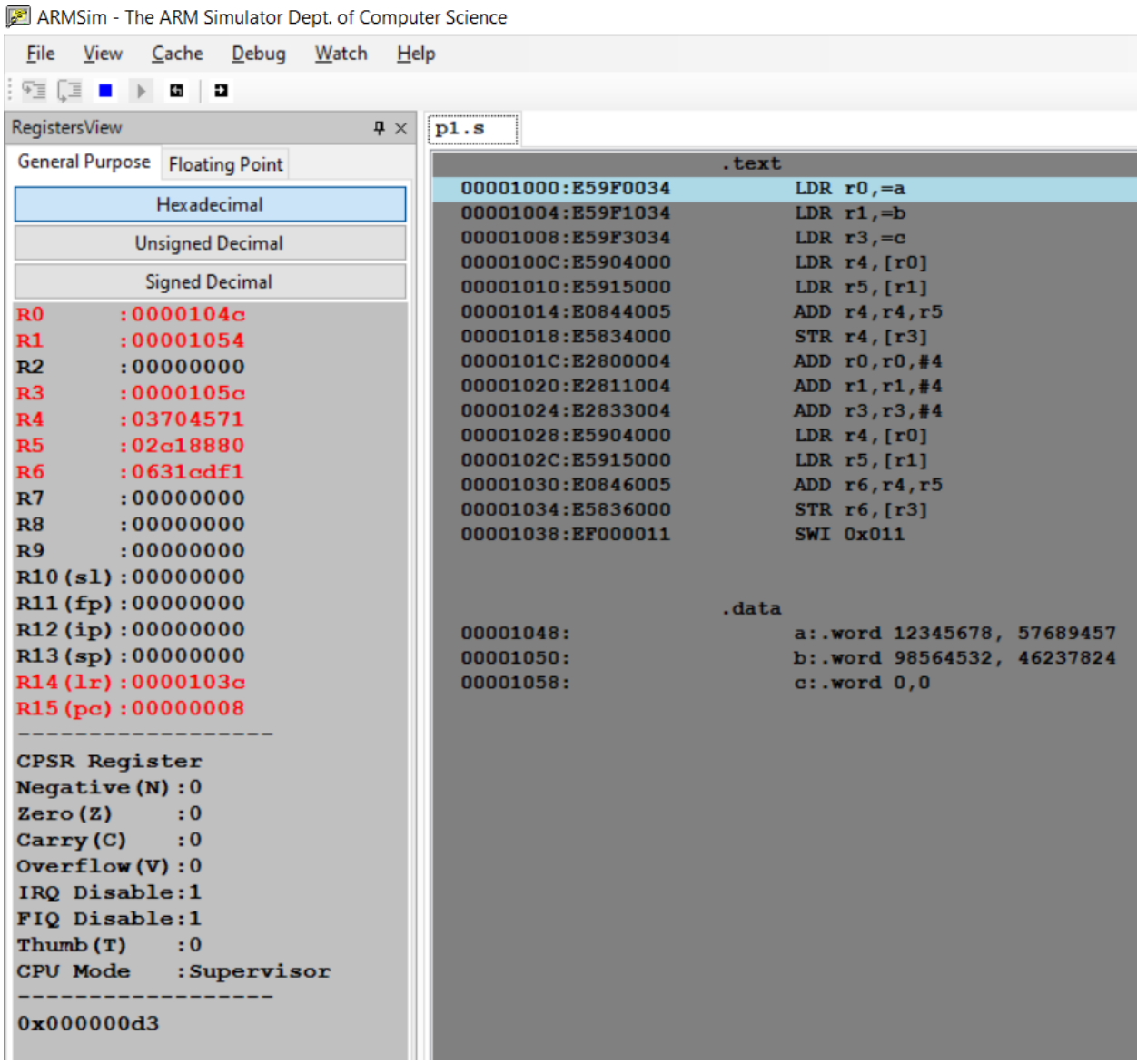
5. Write an ALP to check whether a given number is present in array using Linear Search (Without SWI 0x02), if found move +1 to R6 and key position to R7 else move -1 to R6 (if number not found)

ARM Assembly Code for the program

```
.data
a: .WORD 3,9,10,12,16,38,40,58,65,79
.text
LDR r0, =a
```

```
MOV r1, #10
MOV r4,#0 ;to count the position
Loop:
LDR r2, [r0]
CMP r2, r1
BEQ l1
ADD r4, r4, #1
ADD r0, r0, #4
SUBS r1, r1, #1
BNE Loop
MOV R3, #-1 ;if not found
SWI 0x011
l1:
MOV R3, #1 ;if found
SWI 0x011
.end
```

Output Screen Shot



Week#____3_____ Program Number: __6__

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

ARM Assembly Code for the program

```
.text
MOV r0, #10
MOV r1, #0
MOV r2, #1
LDR r3, =a
STR r1, [r3], #4
STR r2 ,[r3], #4
CMP r0, #1
BNE Loop
Loop:
ADD r5, r1, r2
MOV r1, r2
MOV r2, r5
STR r5, [r3], #4
SUBS r0, r0, #1
CMP r0, #1
BEQ exit
BL Loop
exit:
SWI 0x011
.data
a: .WORD
```

Output Screen Shot