Microprocessor and Computer Architecture Laboratory
COURSE CODE :UE19CS256
4th Semester, Academic Year 2020-21

Date:08/02/2021

Name: Priya Mohata	SRN:PES2UG19CS301	Section:E

Week#3 **Program Number: 1**

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

I. ARM Assembly Code for the program.

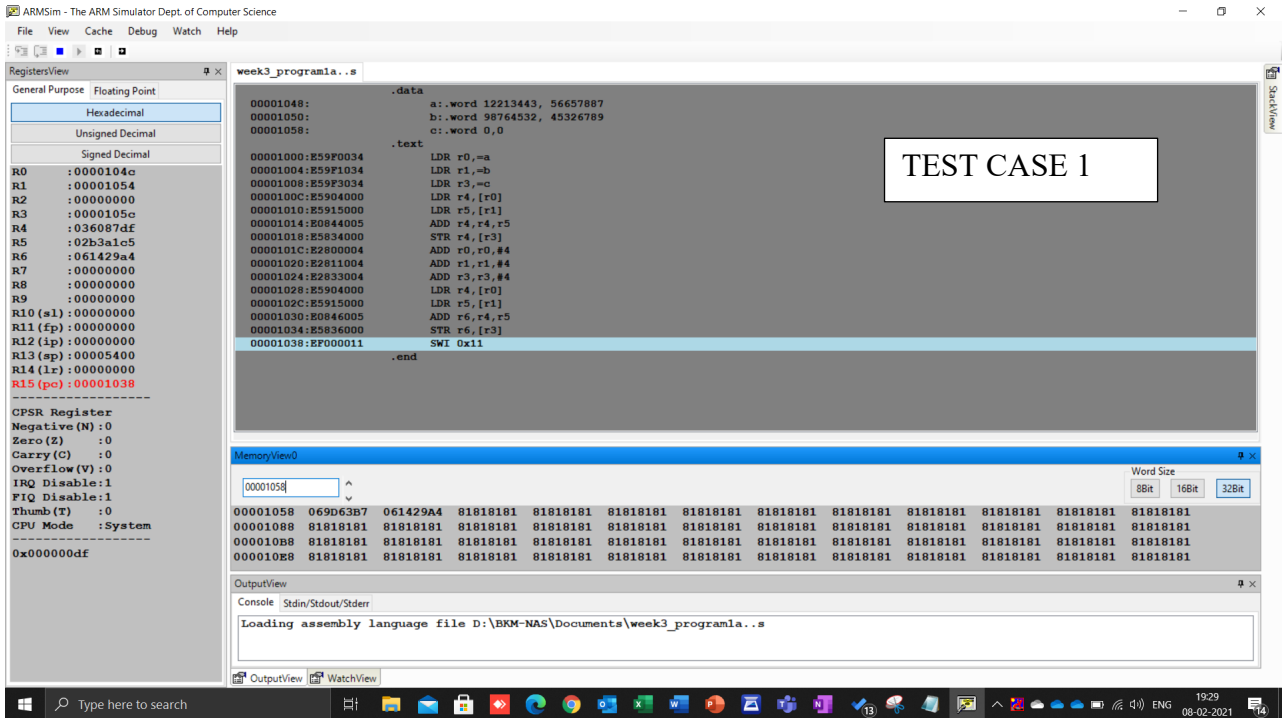
```
.data
        a:.word 12213443,
        b:.word 98764532,
        c:.word 0,0
.text
        LDR r0,=a
        LDR r1,=b
        LDR r3,=c
        LDR r4,[r0]
        LDR r5,[r1]
        ADD r4,r4,r5
        STR r4,[r3]
        ADD r0,r0,#4
        ADD r1,r1,#4
        ADD r3,r3,#4
        LDR r4,[r0]
        LDR r5,[r1]
        ADD r6,r4,r5
        STR r6,[r3]
        SWI 0x11
.end
```

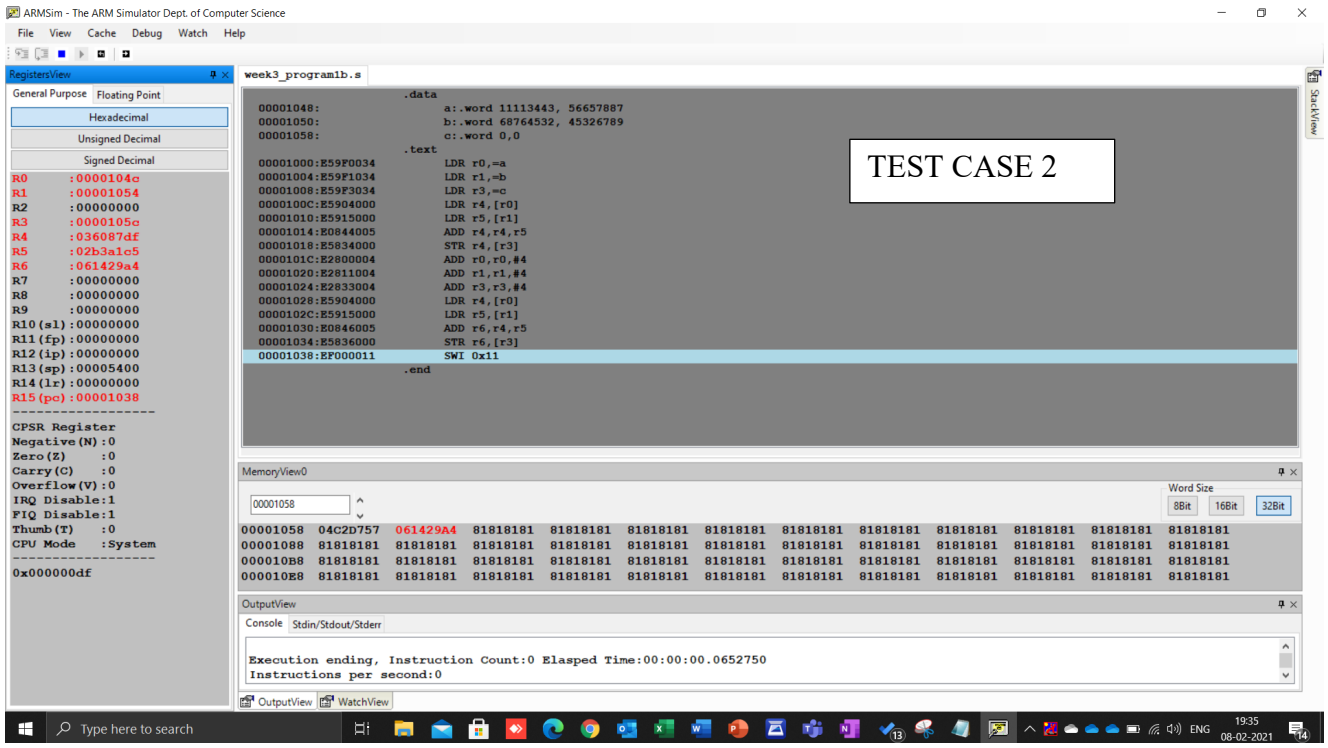
TEST CASE 1

```
.data
      a:.word 11113443, 56657887
      b:.word 68764532, 45326789
      c:.word 0,0
.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 0x11
.end
```

TEST CASE 2

II.Output Screen Shot





III.Output Table for the program

TEST CASE 1	a: .word 12213443, 56657887 b: .word 98764532, 45326789	
	Upper 32 bits	**Lower 32 bits**
a: .word	56657887 **(036087DF)**	12213443 **(00BA5CC3)**
b: .word	45326789 **(02B3A1C5)**	98764532 **(05E306F4)**
c: .word	101984676 **(061429A4)**	110977975 **(069D63B7)**

TEST CASE 2	a: .word 11113443, 56657887 b: .word 68764532, 45326789	
	Upper 32 bits	**Lower 32 bits**
a: .word	56657887 **(036087DF)**	12213443 **(00A993E3)**
b: .word	45326789 **(02B3A1C5)**	98764532 **(04194374)**
c: .word	101984676 **(061429A4)**	110977975 **(04C2D757)**

Week#3 Program Number: 2

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

I. ARM Assembly Code for the program.

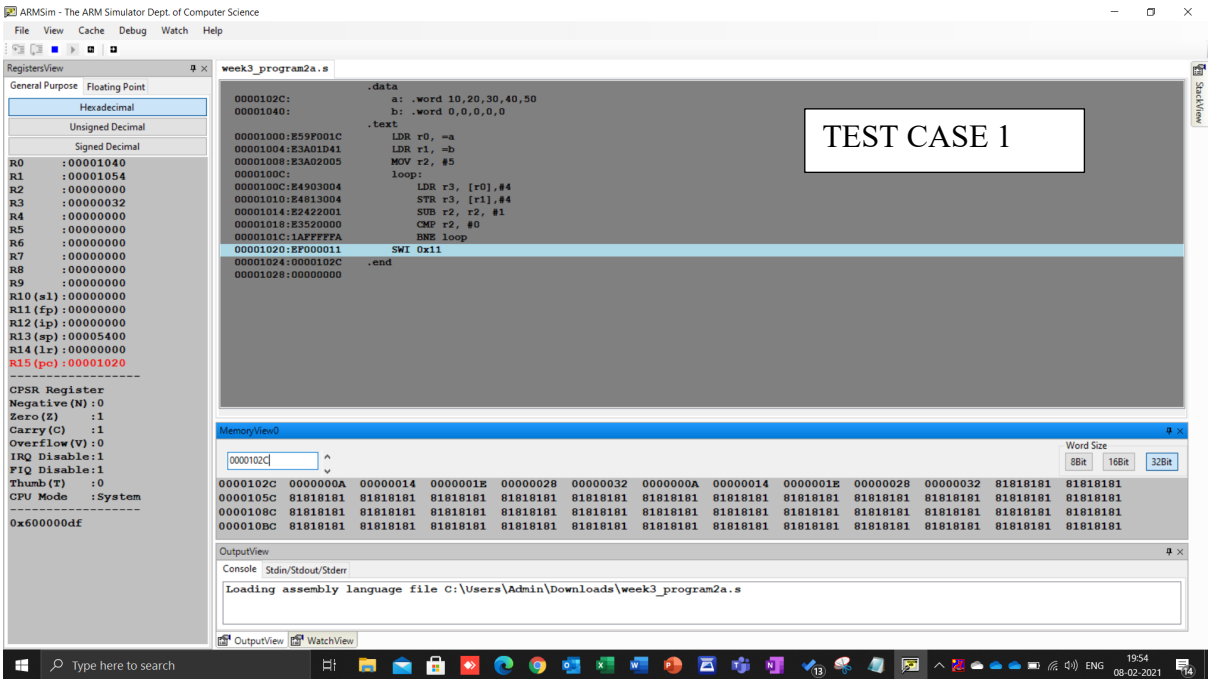
```
.data
    a: .word 10,20,30,40,50
    b: .word 0,0,0,0,0
.text
    LDR r0, =a
    LDR r1, =b
    MOV r2, #5
    loop:
        LDR r3, [r0],#4
        STR r3, [r1],#4
        SUB r2, r2, #1
        CMP r2, #0
        BNE loop
    SWI 0x11
.end
```

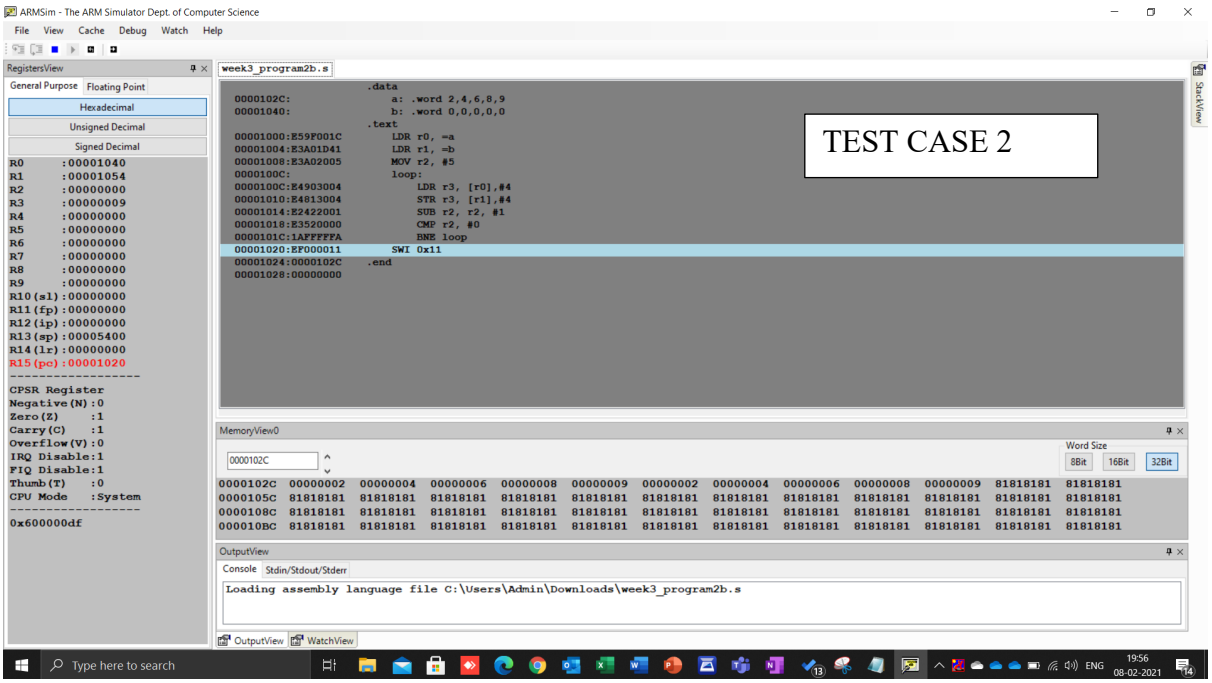
TEST CASE 1

```
.data
    a: .word 2,4,6,8,9
    b: .word 0,0,0,0,0
.text
    LDR r0, =a
    LDR r1, =b
    MOV r2, #5
    loop:
        LDR r3, [r0],#4
        STR r3, [r1],#4
        SUB r2, r2, #1
        CMP r2, #0
        BNE loop
    SWI 0x11
.end
```

TEST CASE 2

II. Output Screen Shot





III. Output Table for the program

.data a: .word 10, 20, 30, 40,50 b: .word 0, 0, 0, 0,0	TEST CASE 1
1st Iteration	a: .word 0A, 14, 1E, 28,32 b: .word 0A, 0, 0, 0,0
2nd Iteration	a: .word 0A, 14, 1E, 28,32 b: .word 0A, 14, 0, 0,0
3rd Iteration	a: .word 0A, 14, 1E, 28,32 b: .word 0A, 14, 1E, 0,0
4th Iteration	a: .word 0A, 14, 1E, 28,32 b: .word 0A, 14, 1E, 28,0
5th Iteration	a: .word 0A, 14, 1E, 28,32 b: .word 0A, 14, 1E, 28,32
.data a: .word 2, 4,6,8,9 b: .word 0, 0, 0, 0,0	TEST CASE 2
1st Iteration	a: .word 02, 04, 06, 08,09 b: .word 02, 0, 0, 0,0
2nd Iteration	a: .word 02, 04, 06, 08,09 b: .word 02, 04, 0, 0,0

3rd Iteration	a: .word 02, 04, 06, 08,09 b: .word 02, 04, 06, 0,0
4th Iteration	a: .word 02, 04, 06, 08,09 b: .word 02, 04, 06, 08,00
5th Iteration	a: .word 02, 04, 06, 08,09 b: .word 02, 04, 06, 08,09

Week#3 ProgramNumber:3

Title: Write an ALP to find smallest number in an array of n 32 bit numbers

I. ARM Assembly Code for the program.

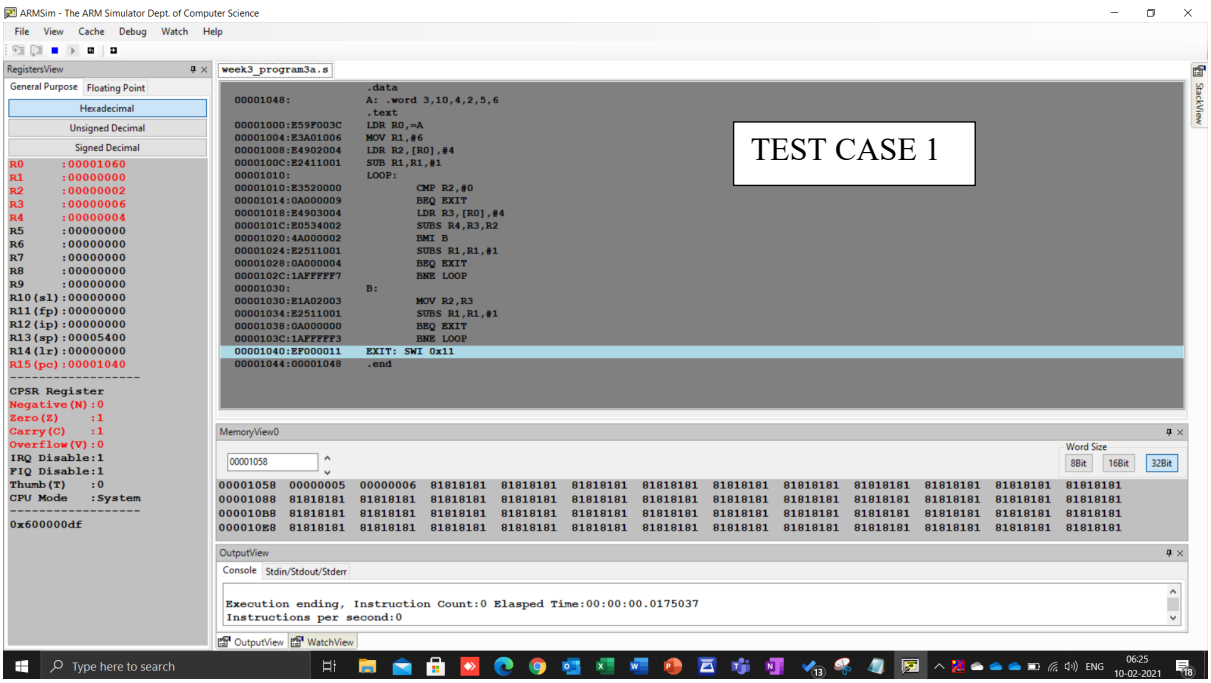
```
.data
A: .word 3,10,4,2,5,6
.text
LDR R0,=A
MOV R1,#6
LDR R2,[R0],#4
SUB R1,R1,#1
LOOP:
        CMP R2,#0
        BEQ EXIT
        LDR R3,[R0],#4
        SUBS R4,R3,R2
        BMI B
        SUBS R1,R1,#1
        BEQ EXIT
        BNE LOOP
B:
        MOV R2,R3
        SUBS R1,R1,#1
        BEQ EXIT
        BNE LOOP
EXIT: SWI 0x11
.end
```

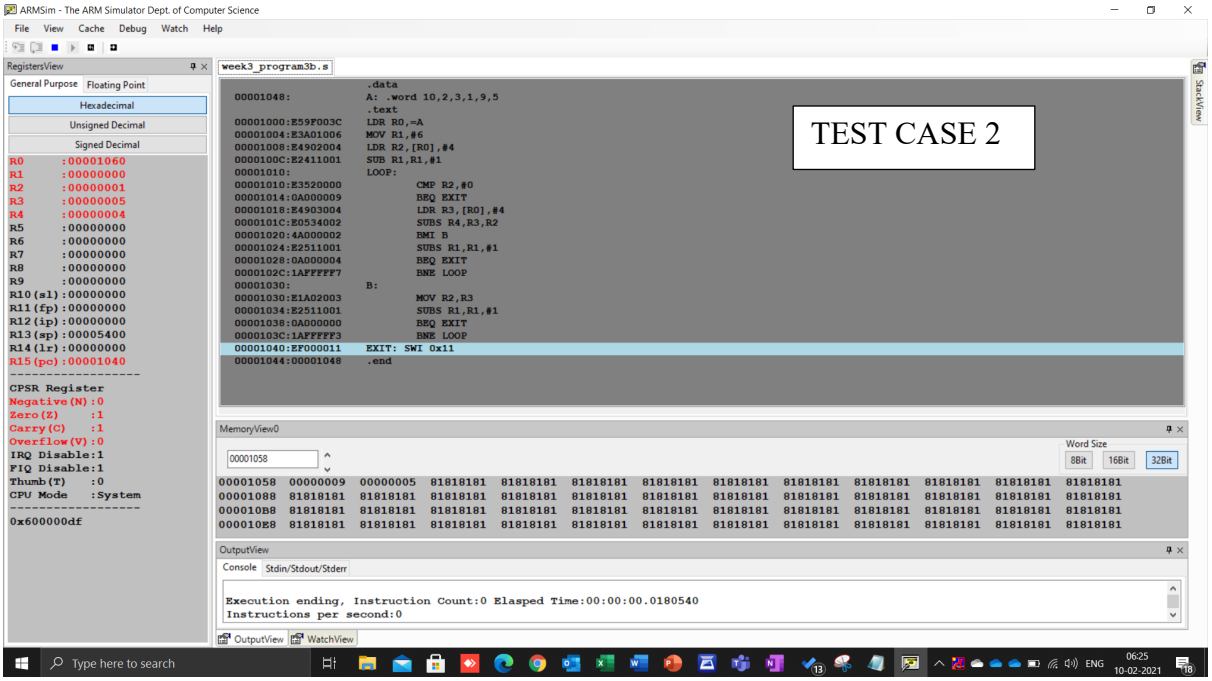
TEST CASE 1

```
.data
A: .word 10,2,3,1,9,5
.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 B
        SUBS R1,R1,#1
        BEQ EXIT
        BNE LOOP
B:
        MOV R2,R3
        SUBS R1,R1,#1
        BEQ EXIT
        BNE LOOP
EXIT: SWI 0x11
.end
```

TEST CASE 2

II. Output Screen Shot





III. Output Table for the program

a: .word 3,10,4,2, 5, 6	TEST CASE 1
1st Iteration	R2=3,R3=10 (R3>R2)
2nd Iteration	R2=3, R3=4 (R3>R2)
3rd Iteration	R2=3, R3=2 (R3<R2)
4th Iteration	R2=2, R3=5 (R3>R2)
5th Iteration	R2=2, R3=6 (R3>R2)
Smallest number is present in R2	

a: .word 10,2,3,1,9,5	TEST CASE 2
1st Iteration	R2=10,R3=2 (R3<R2)
2nd Iteration	R2=2, R3=3 (R3>R2)
3rd Iteration	R2=2, R3=1 (R3<R2)
4th Iteration	R2=1, R3=9 (R3>R2)
5th Iteration	R2=1, R3=5 (R3>R2)
Smallest number is present in R2	

Week#3 Program Number: 4a

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

I. ARM Assembly Code for the program.

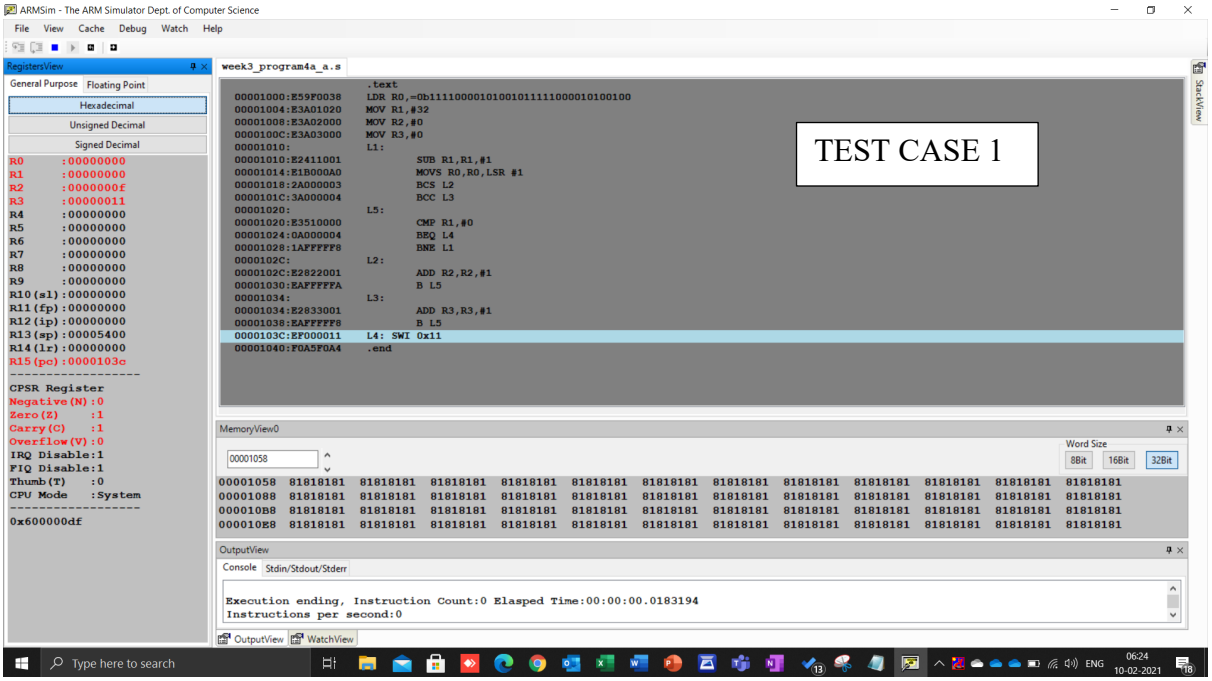
```
.text
LDR R0,=0b11110000101001011111000010100100
MOV R1,#32
MOV R2,#0
MOV R3,#0
L1:
        SUB R1,R1,#1
        MOVS R0,R0,LSR #1
        BCS L2
        BCC L3
L5:
        CMP R1,#0
        BEQ L4
        BNE L1
L2:
        ADD R2,R2,#1
        B L5
L3:
        ADD R3,R3,#1
        B L5
L4: SWI 0x11
.end
```

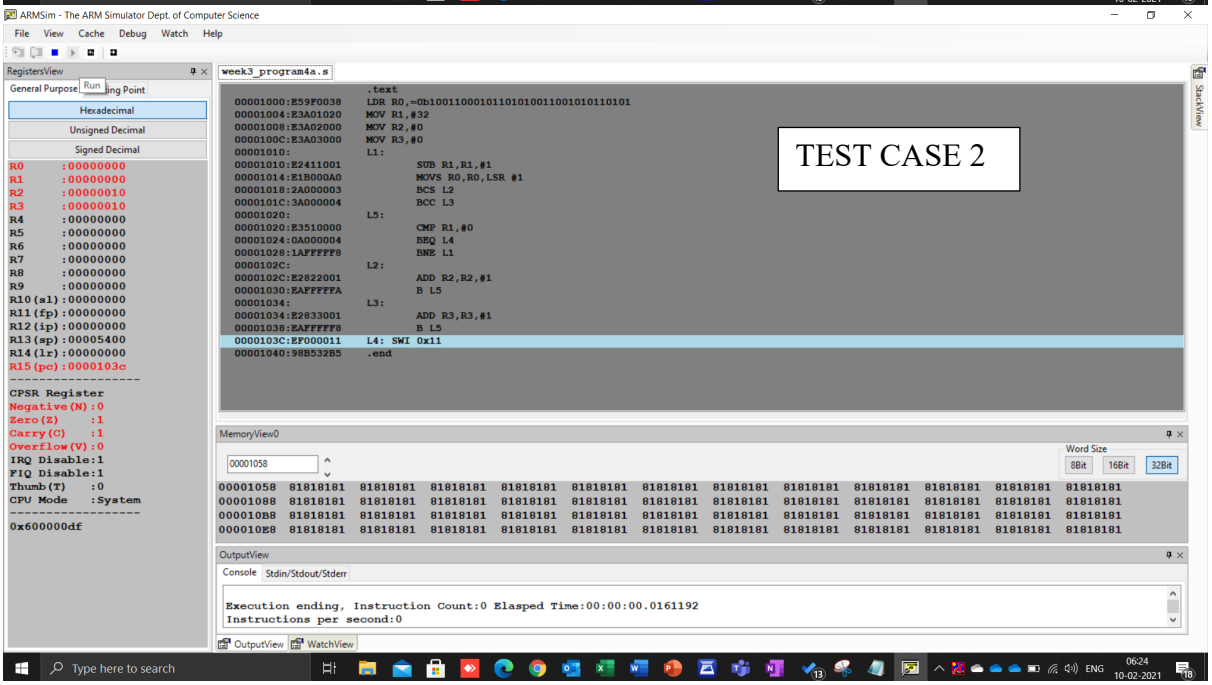
TEST CASE 1

```
.text
LDR R0,=0b10011000101101010011001010110101
MOV R1,#32
MOV R2,#0
MOV R3,#0
L1:
        SUB R1,R1,#1
        MOVS R0,R0,LSR #1
        BCS L2
        BCC L3
L5:
        CMP R1,#0
        BEQ L4
        BNE L1
L2:
        ADD R2,R2,#1
        B L5
L3:
        ADD R3,R3,#1
        B L5
L4: SWI 0x11
.end
```

TEST CASE 2

II. Output Screen Shot


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000000
R2 :0000000f
R3 :00000011
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):0000103c
CPSR Register
Negative(N):0
Zero(Z) :1
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x600000df
week3_program4a_a.s
.text
00001000:E59F0038 LDR R0,=0b11110000101001011111000010100100
00001004:E3A01020 MOV R1,#32
00001008:E3A02000 MOV R2,#0
0000100C:E3A03000 MOV R3,#0
00001010: L1:
00001010:E2411001 SUB R1,R1,#1
00001014:E1B000A0 MOVS R0,R0,LSR #1
00001018:2A000003 BCS L2
0000101C:3A000004 BCC L3
00001020: L5:
00001020:E3510000 CMP R1,#0
00001024:0A000004 BEQ L4
00001028:1AFFFFF8 BNE L1
0000102C: L2:
0000102C:E2822001 ADD R2,R2,#1
00001030:EAFFFFFA B L5
00001034: L3:
00001034:E2833001 ADD R3,R3,#1
00001038:EAFFFFF8 B L5
0000103C:EF000011 L4: SWI 0x11
00001040:F0A5F0A4 .end
TEST CASE 1
MemoryView0
00001058
Word Size
8Bit 16Bit 32Bit
00001058 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001088 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010B8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010E8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
OutputView
Console Stdin/Stdout/Stderr
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0183194
Instructions per second:0
OutputView WatchView
Type here to search
ENG 06:24 10-02-2021


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
Run
General Purpose ing Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00000000
R1 :00000000
R2 :00000010
R3 :00000010
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):0000103c
CPSR Register
Negative(N):0
Zero(Z) :1
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x600000df
week3_program4a.s
.text
00001000:E59F0038 LDR R0,=0b10011000101101010011001010110101
00001004:E3A01020 MOV R1,#32
00001008:E3A02000 MOV R2,#0
0000100C:E3A03000 MOV R3,#0
00001010: L1:
00001010:E2411001 SUB R1,R1,#1
00001014:E1B000A0 MOVS R0,R0,LSR #1
00001018:2A000003 BCS L2
0000101C:3A000004 BCC L3
00001020: L5:
00001020:E3510000 CMP R1,#0
00001024:0A000004 BEQ L4
00001028:1AFFFFF8 BNE L1
0000102C: L2:
0000102C:E2822001 ADD R2,R2,#1
00001030:EAFFFFFA B L5
00001034: L3:
00001034:E2833001 ADD R3,R3,#1
00001038:EAFFFFF8 B L5
0000103C:EF000011 L4: SWI 0x11
00001040:98B532B5 .end
TEST CASE 2
MemoryView0
00001058
Word Size
8Bit 16Bit 32Bit
00001058 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001088 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010B8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010E8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
OutputView
Console Stdin/Stdout/Stderr
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0161192
Instructions per second:0
OutputView WatchView
Type here to search
ENG 06:24 10-02-2021

III.Output Table for the program

r0, =0b11110000101001011111000010100100		
r1	32	
r2	After execution	15 (=0F in hex)
r3	After execution	17 (=11 in hex)

r0, =0b10011000101101010011001010110101	
After execution r2	16 (=10 in hex)
After execution r3	16 (=10 in hex)

Week#3 Program Number: 4b

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

I. ARM Assembly Code for the program.

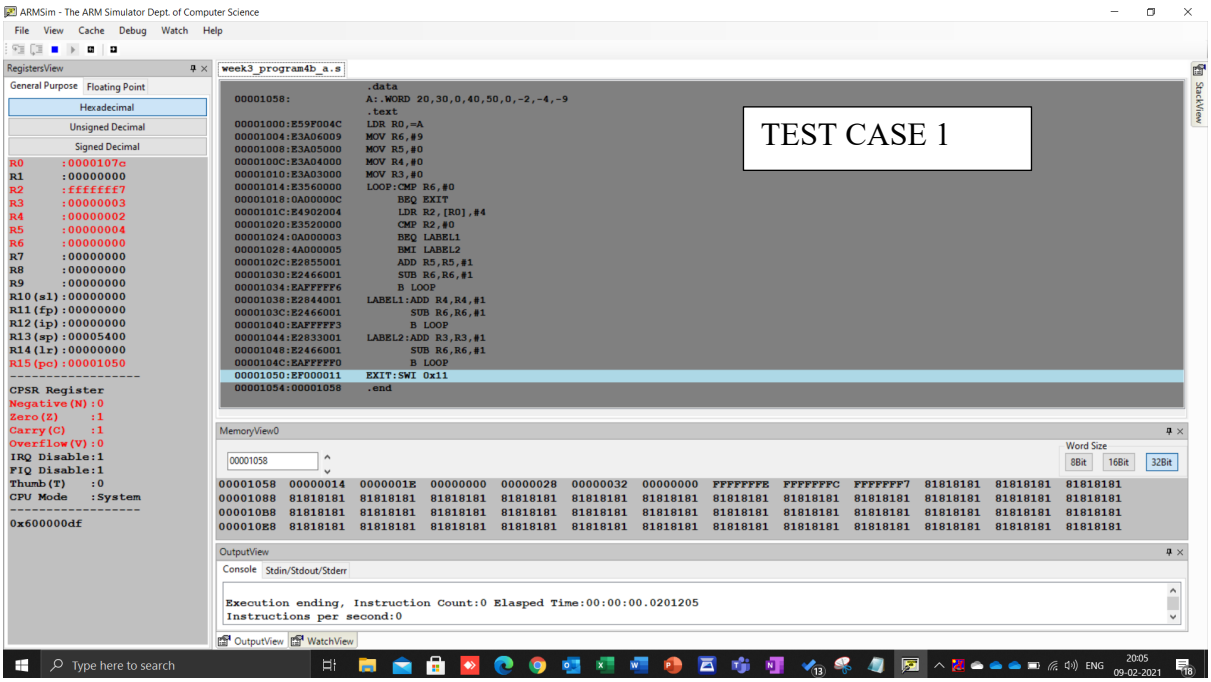
```
.data
A:.WORD 20,30,0,40,50,0,-2,-4,-9
.text
LDR R0,=A
MOV R6,#9
MOV R5,#0
MOV R4,#0
MOV R3,#0
LOOP:CMP R6,#0
     BEQ EXIT
     LDR R2,[R0],#4
     CMP R2,#0
     BEQ LABEL1
     BMI LABEL2
     ADD R5,R5,#1
     SUB R6,R6,#1
     B LOOP
LABEL1:ADD R4,R4,#1
       SUB R6,R6,#1
       B LOOP
LABEL2:ADD R3,R3,#1
       SUB R6,R6,#1
       B LOOP
EXIT:SWI 0x11
.end
```

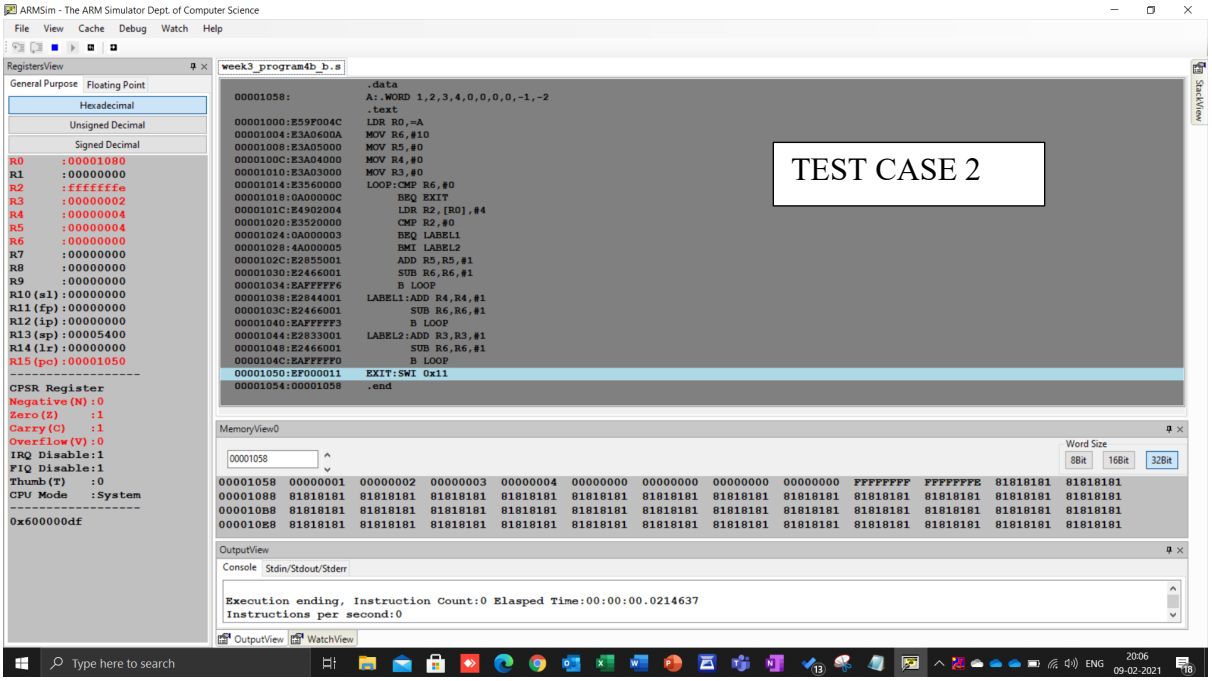
TEST CASE 1

```
.data
A:.WORD 1,2,3,4,0,0,0,0,-1,-2
.text
LDR R0,=A
MOV R6,#10
MOV R5,#0
MOV R4,#0
MOV R3,#0
LOOP:CMP R6,#0
     BEQ EXIT
     LDR R2,[R0],#4
     CMP R2,#0
     BEQ LABEL1
     BMI LABEL2
     ADD R5,R5,#1
     SUB R6,R6,#1
     B LOOP
LABEL1:ADD R4,R4,#1
       SUB R6,R6,#1
       B LOOP
LABEL2:ADD R3,R3,#1
       SUB R6,R6,#1
       B LOOP
EXIT:SWI 0x11
.end
```

TEST CASE 2

II. Output Screen Shot





III. Output Table for the program

a:.word 20,30,0,40,50,0,-2,-4,-9	TEST CASE 1
R3	3
R4	2
R5	4

a:.word 1,2,3,4,0,0,0,0,-1,-2	TEST CASE 2
R3	2
R4	4
R5	4

Week#3 **Program Number:5**

Title: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)

I. ARM Assembly Code for the program.

```
.data
a: .word 10,20,30,40,50
.text
LDR r0, =a
MOV r3, #5
loop:
MOV r2,#30
LDR r1, [r0]
ADD r0, r0, #4
CMP r1, r2
BEQ l1
SUBS r3, r3, #1
BNE loop
MOV r6, #-1
SWI 0x11
l1:
RSB r7,r3,#6
SWI 0x11
.end
```

TEST CASE 1

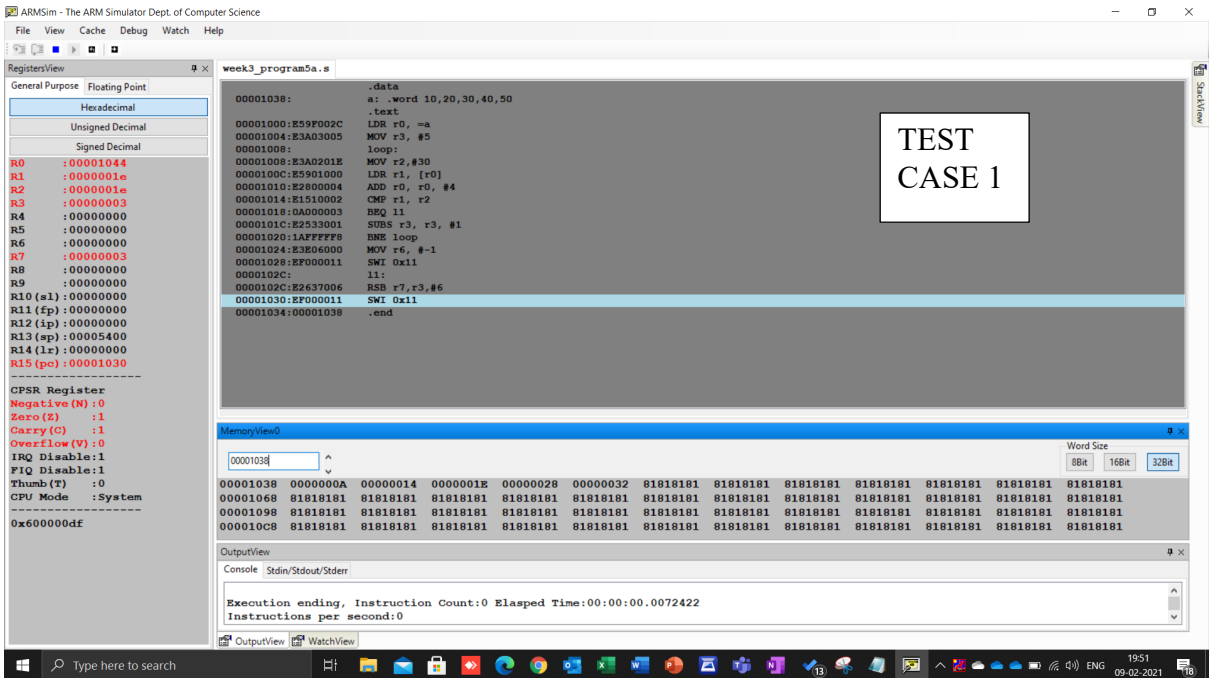
```
.data
a: .word 10,20,30,40,50
.text
LDR r0, =a
MOV r3, #5
loop:
MOV r2,#20
LDR r1, [r0]
ADD r0, r0, #4
CMP r1, r2
BEQ l1
SUBS r3, r3, #1
BNE loop
MOV r6, #-1
SWI 0x11
l1:
RSB r7,r3,#6
SWI 0x11
.end
```

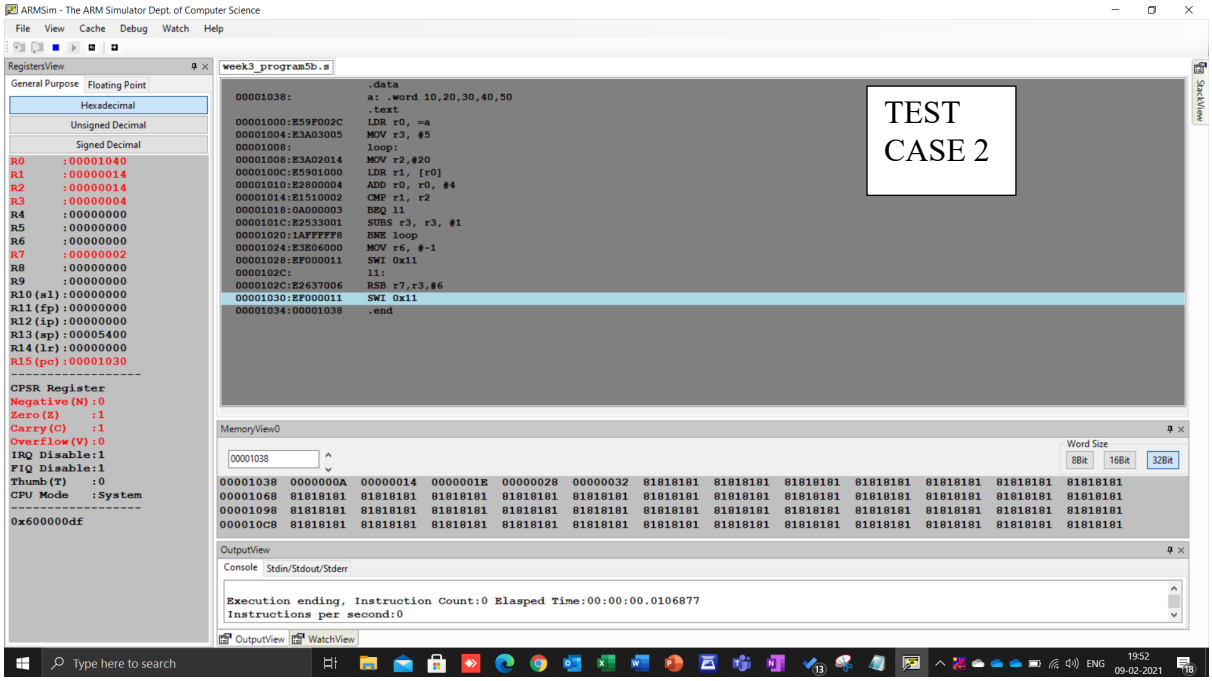
TEST CASE 2

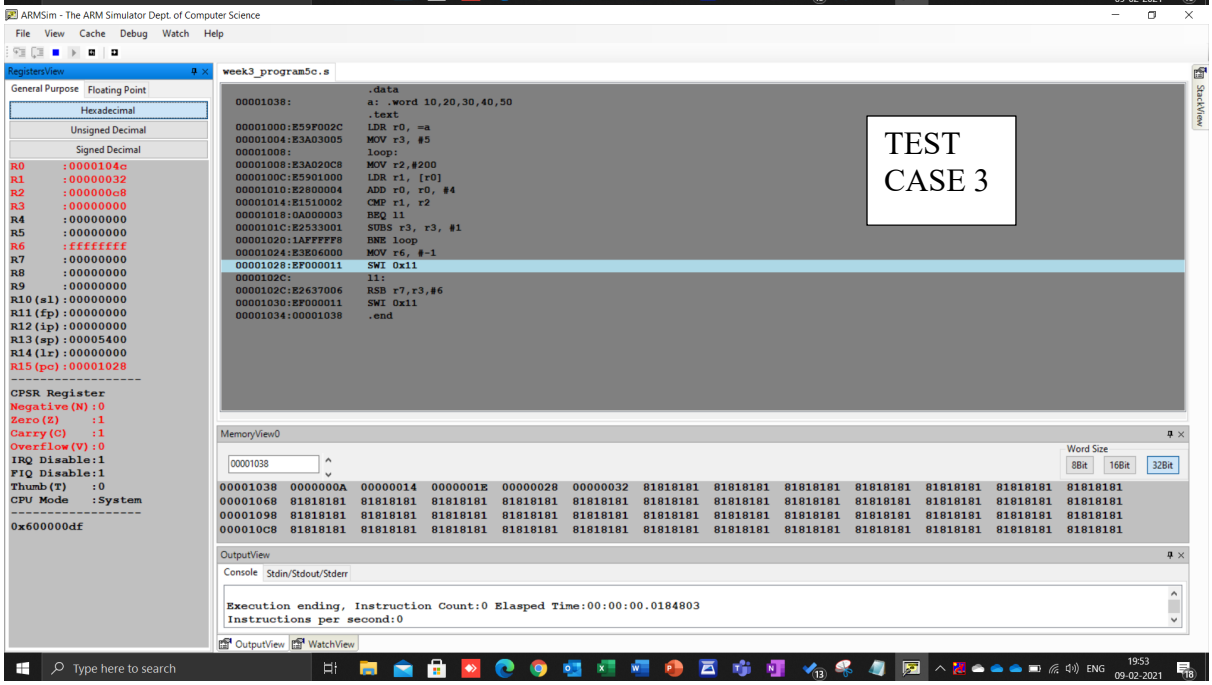
```
.data
a: .word 10,20,30,40,50
.text
LDR r0, =a
MOV r3, #5
loop:
MOV r2,#200
LDR r1, [r0]
ADD r0, r0, #4
CMP r1, r2
BEQ l1
SUBS r3, r3, #1
BNE loop
MOV r6, #-1
SWI 0x11
l1:
RSB r7,r3,#6
SWI 0x11
.end
```

TEST CASE 3

II. Output Screen Shot




ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :00001040
R1 :00000014
R2 :00000014
R3 :00000004
R4 :00000000
R5 :00000000
R6 :00000000
R7 :00000002
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):00001030
CPSR Register
Negative(N):0
Zero(Z) :1
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x600000df
week3_program5b.s
.data
00001038: a: .word 10,20,30,40,50
.text
00001000:E59F002C LDR r0, =a
00001004:E3A03005 MOV r3, #5
00001008: loop:
00001008:E3A02014 MOV r2,#20
0000100C:E5901000 LDR r1, [r0]
00001010:E2800004 ADD r0, r0, #4
00001014:E1510002 CMP r1, r2
00001018:0A000003 BEQ l1
0000101C:E2533001 SUBS r3, r3, #1
00001020:1AFFFFF8 BNE loop
00001024:E3E06000 MOV r6, #-1
00001028:EF000011 SWI 0x11
0000102C: l1:
0000102C:E2637006 RSB r7,r3,#6
00001030:EF000011 SWI 0x11
00001034:00001038 .end
TEST CASE 2
MemoryView0
00001038
Word Size
8Bit 16Bit 32Bit
00001038 0000000A 00000014 0000001E 00000028 00000032 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001068 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001098 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010C8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
OutputView
Console Stdin/Stdout/Stderr
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0106877
Instructions per second:0
OutputView WatchView
Type here to search
ENG 19:52 09-02-2021


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :0000104c
R1 :00000032
R2 :000000c8
R3 :00000000
R4 :00000000
R5 :00000000
R6 :ffffffff
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):00001028
CPSR Register
Negative(N):0
Zero(Z) :1
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System
0x600000df
week3_program5c.s
.data
00001038: a: .word 10,20,30,40,50
.text
00001000:E59F002C LDR r0, =a
00001004:E3A03005 MOV r3, #5
00001008: loop:
00001008:E3A020C8 MOV r2,#200
0000100C:E5901000 LDR r1, [r0]
00001010:E2800004 ADD r0, r0, #4
00001014:E1510002 CMP r1, r2
00001018:0A000003 BEQ l1
0000101C:E2533001 SUBS r3, r3, #1
00001020:1AFFFFF8 BNE loop
00001024:E3E06000 MOV r6, #-1
00001028:EF000011 SWI 0x11
0000102C: l1:
0000102C:E2637006 RSB r7,r3,#6
00001030:EF000011 SWI 0x11
00001034:00001038 .end
TEST CASE 3
MemoryView0
00001038
Word Size
8Bit 16Bit 32Bit
00001038 0000000A 00000014 0000001E 00000028 00000032 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001068 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001098 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010C8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
OutputView
Console Stdin/Stdout/Stderr
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0184803
Instructions per second:0
OutputView WatchView
Type here to search
ENG 19:53 09-02-2021

III. OUTPUT TABLE

TEST CASE 1:		HEX value
A:.WORD 10,20,30,40,50		
R2	KEY =30	1E
R3	COUNT =5	
R0	Address of A	00001038
R3	After Execution =3	Position of key element =3
TEST CASE 2:		HEX value
A:.WORD 10,20,30,40,50		
R2	KEY =20	1E
R3	COUNT =5	
R0	Address of A	00001038
R3	After Execution =2	Position of key element =2
TEST CASE 2:		HEX value
A:.WORD 10,20,30,40,50		
R2	KEY =20	1E
R3	COUNT =5	
R0	Address of A	00001038
R3	After Execution =2	Position of key element =2

Week#3 Program Number: 6

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

I.ARM ASSEMBLY CODE

```
.data
a: .word
.text
MOV r0,#6
LDR r1,=a
MOV r2,#0
MOV r3,#1
STR r2,[r1]
ADD r1, r1, #4
STR r3, [r1]

loop:
ADD r4, r2, r3
ADD r1, r1, #4
STR r4, [r1]
MOV r2, r3
MOV r3, r4
SUB r0, r0, #1
CMP r0,#0
BNE loop
.end
```

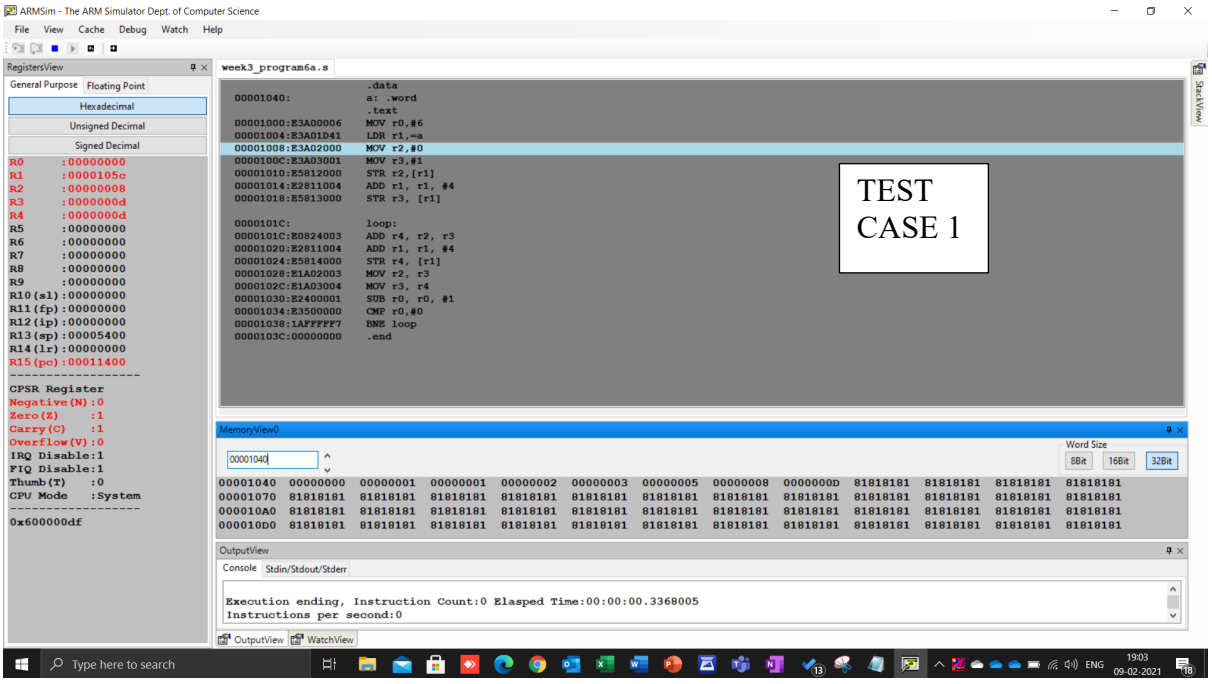
TEST CASE 1

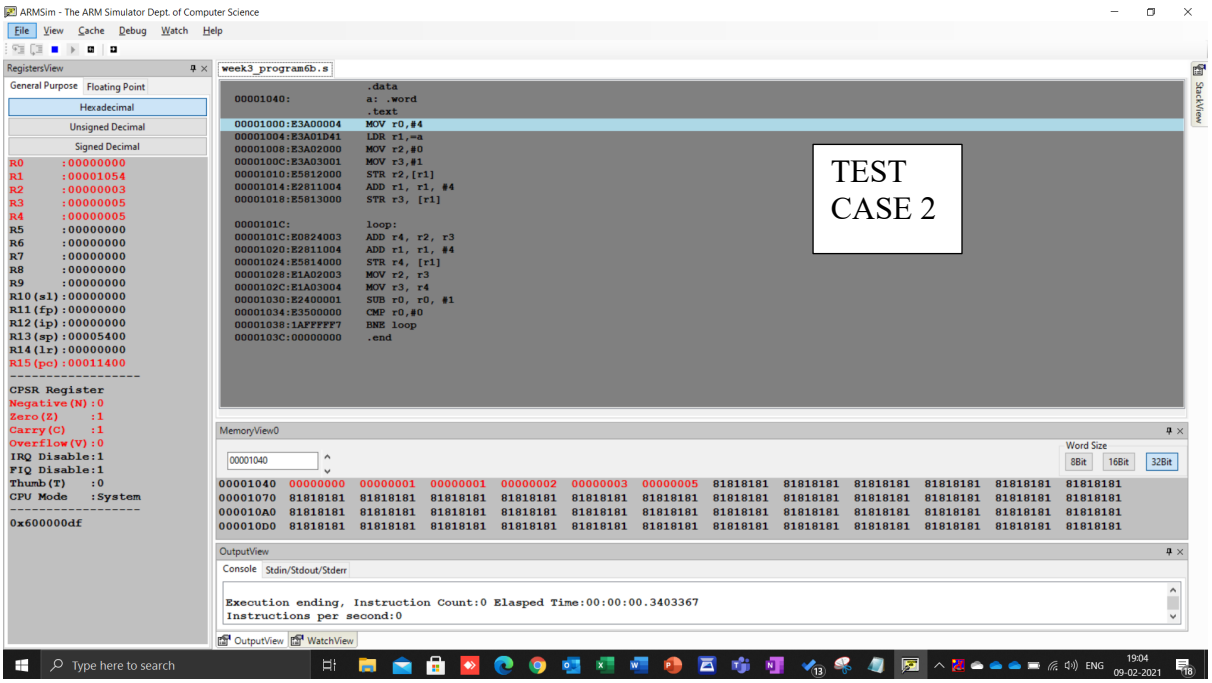
```
.data
a: .word
.text
MOV r0,#4
LDR r1,=a
MOV r2,#0
MOV r3,#1
STR r2,[r1]
ADD r1, r1, #4
STR r3, [r1]

loop:
ADD r4, r2, r3
ADD r1, r1, #4
STR r4, [r1]
MOV r2, r3
MOV r3, r4
SUB r0, r0, #1
CMP r0,#0
BNE loop
.end
```

TEST CASE 2

II.OUTPUT SCREENSHOT





III.OUTPUT TABLE

R0	**Fibonacci Count**	**6**
R1	**Address of A**	
R2	**Initially 0**	
R3	**Initially 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 =0D**

R0	**Fibonacci Count**	**4**
R1	**Address of A**	
R2	**Initially 0**	
R3	**Initially 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**

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: PRIYA MOHATA
Name: PRIYA MOHATA
SRN: PES2UG19CS301
Section: E
Date: 09/02/2021