Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 02/02/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	Section : A

Week#____2_______Program Number: ____1___

Title of the Program

Based on the value of the number in R0, Write an ALP to store 1 in R1 if R0 is zero, Store 2 in R1 if R0 is positive, Store 3 in R1 if R0 is negative.

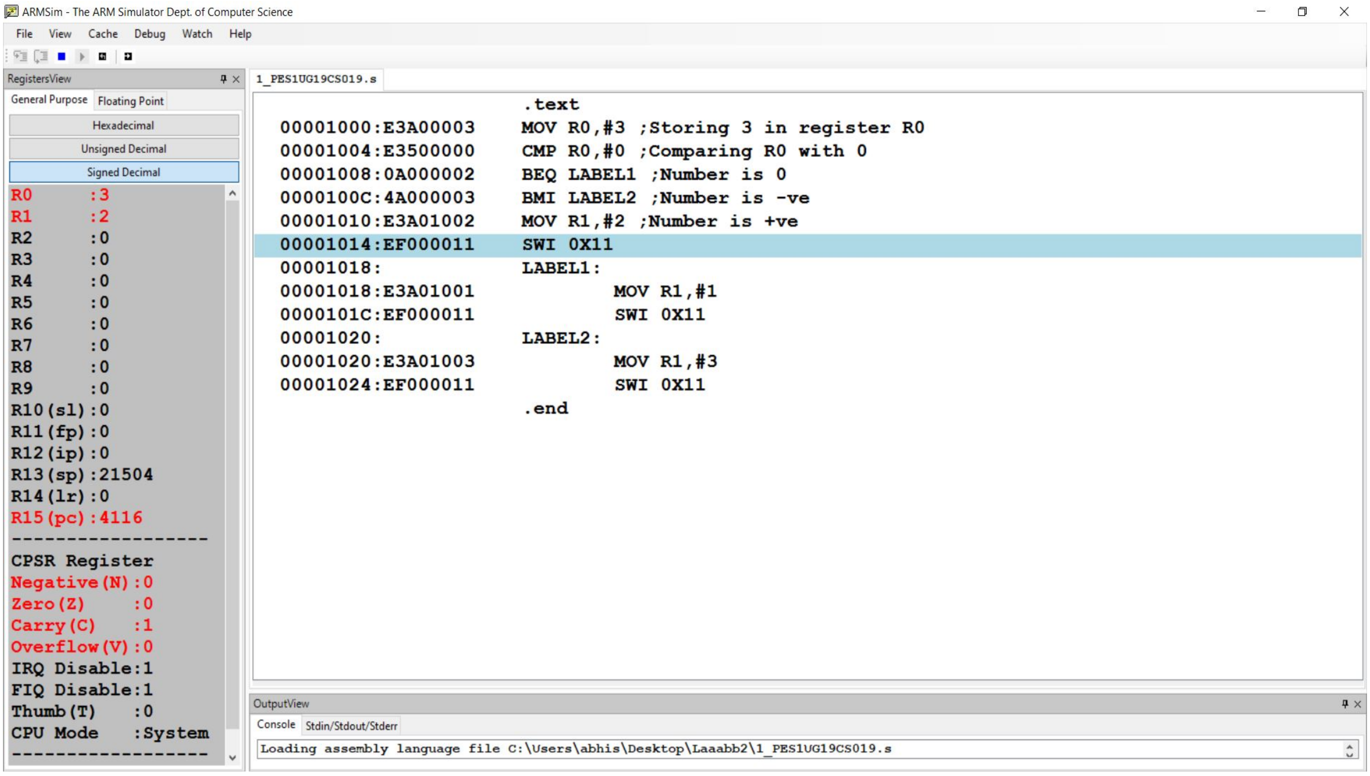
I. ARM Assembly Code for each program

```
.text

MOV R0,#3 ;Storing 3 in register R0

CMP R0,#0 ;Comparing R0 with 0

BEQ LABEL1 ;Number is 0

BMI LABEL2 ;Number is -ve

MOV R1,#2 ;Number is +ve

SWI 0X11
```

```
LABEL1:
     MOV R1,#1
     SWI 0X11
LABEL2:
     MOV R1,#3
     SWI 0X11
.end
```

II.Final Output Screen Shot

Test case 1 : Storing 3 in register R0

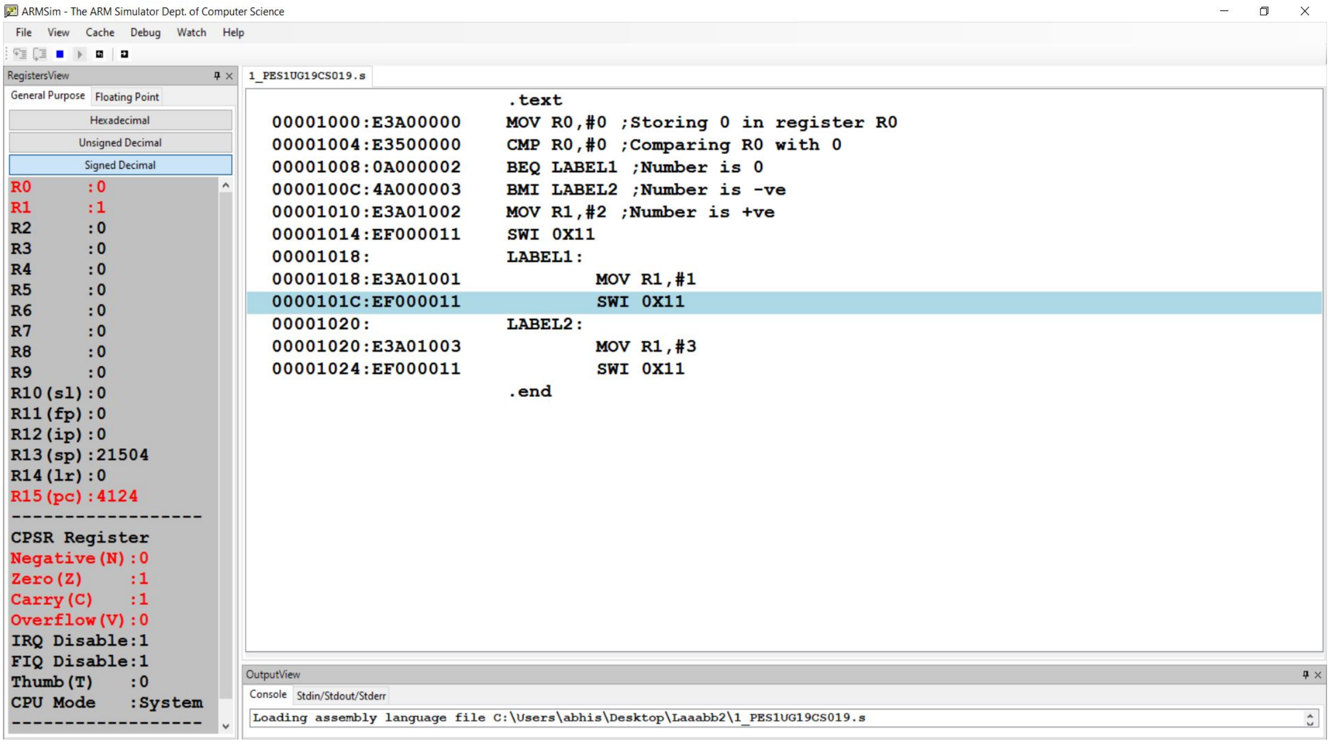


```
R0      :3
R1      :2
R2      :0
R3      :0
R4      :0
R5      :0
R6      :0
R7      :0
R8      :0
R9      :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4116
------------------
CPSR Register
Negative(N):0
Zero(Z)    :0
Carry(C)   :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
```

```
                      .text
00001000:E3A00003     MOV R0,#3  ;Storing 3 in register R0
00001004:E3500000     CMP R0,#0  ;Comparing R0 with 0
00001008:0A000002     BEQ LABEL1  ;Number is 0
0000100C:4A000003     BMI LABEL2  ;Number is -ve
00001010:E3A01002     MOV R1,#2  ;Number is +ve
00001014:EF000011     SWI 0X11
00001018:             LABEL1:
00001018:E3A01001          MOV R1,#1
0000101C:EF000011          SWI 0X11
00001020:             LABEL2:
00001020:E3A01003          MOV R1,#3
00001024:EF000011          SWI 0X11
                      .end
```

```
Loading assembly language file C:\Users\abhis\Desktop\Laaabb2\1_PES1UG19CS019.s
```

Test case 2 : Storing 0 in register R0

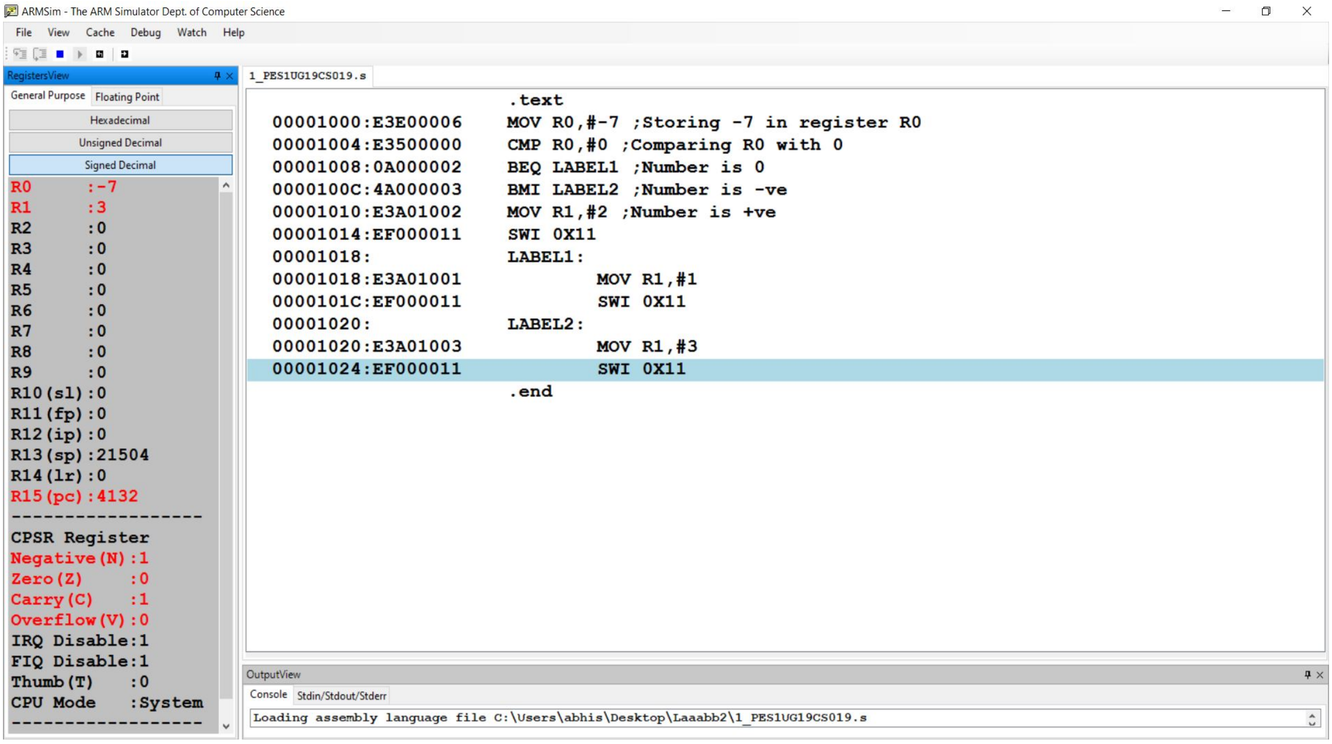


```
R0       :0
R1       :1
R2       :0
R3       :0
R4       :0
R5       :0
R6       :0
R7       :0
R8       :0
R9       :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4124
------------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
------------------
```

```
                      .text
00001000:E3A00000     MOV R0,#0 ;Storing 0 in register R0
00001004:E3500000     CMP R0,#0 ;Comparing R0 with 0
00001008:0A000002     BEQ LABEL1 ;Number is 0
0000100C:4A000003     BMI LABEL2 ;Number is -ve
00001010:E3A01002     MOV R1,#2 ;Number is +ve
00001014:EF000011     SWI 0X11
00001018:             LABEL1:
00001018:E3A01001             MOV R1,#1
0000101C:EF000011             SWI 0X11
00001020:             LABEL2:
00001020:E3A01003             MOV R1,#3
00001024:EF000011             SWI 0X11
                      .end
```

```
Loading assembly language file C:\Users\abhis\Desktop\Laaabb2\1_PES1UG19CS019.s
```

Test case 3 : Storing -7 in register R0



```
R0       :-7
R1       :3
R2       :0
R3       :0
R4       :0
R5       :0
R6       :0
R7       :0
R8       :0
R9       :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4132
------------------
CPSR Register
Negative(N):1
Zero(Z)     :0
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
------------------
```

```
                      .text
00001000:E3E00006     MOV R0,#-7 ;Storing -7 in register R0
00001004:E3500000     CMP R0,#0 ;Comparing R0 with 0
00001008:0A000002     BEQ LABEL1 ;Number is 0
0000100C:4A000003     BMI LABEL2 ;Number is -ve
00001010:E3A01002     MOV R1,#2 ;Number is +ve
00001014:EF000011     SWI 0X11
00001018:             LABEL1:
00001018:E3A01001             MOV R1,#1
0000101C:EF000011             SWI 0X11
00001020:             LABEL2:
00001020:E3A01003             MOV R1,#3
00001024:EF000011             SWI 0X11
                      .end
```

```
Loading assembly language file C:\Users\abhis\Desktop\Laaabb2\1_PES1UG19CS019.s
```

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 02/02/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	Section : A

Week#____2_______Program Number: ____2___

Title of the Program

Write an ALP to compare the value of R0 and R1, add if R0 = R1, else subtract

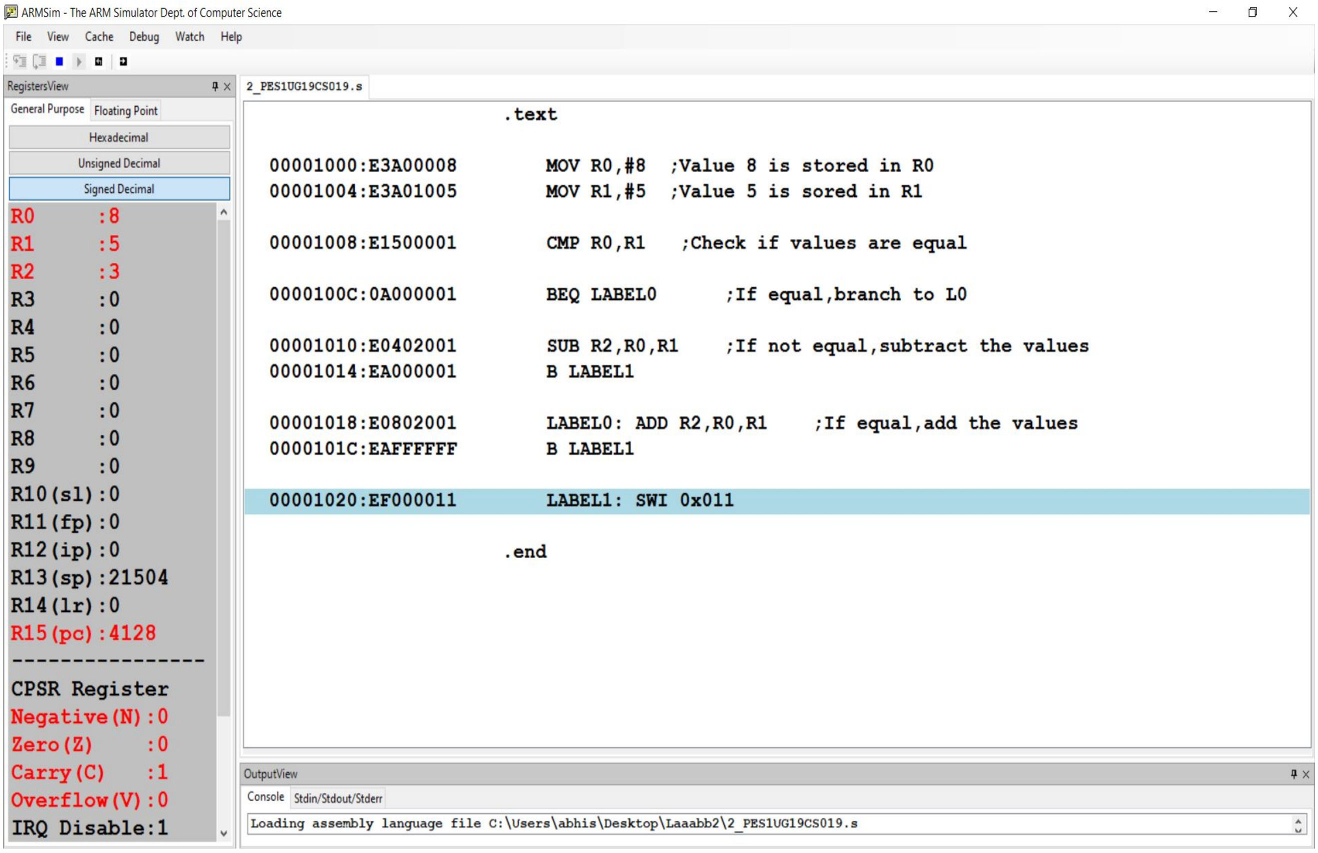
I. ARM Assembly Code for each program

```
.text

MOV R0,#8  ;Value 8 is stored in R0

MOV R1,#5  ;Value 5 is stored in R1

CMP R0,R1   ;Check if values are equal

BEQ LABEL0  ;If equal,branch to L0

SUB R2,R0,R1  ;If not equal,subtract the values

B LABEL1

LABEL0: ADD R2,R0,R1  ;If equal,add the values
```

```
B LABEL1

LABEL1: SWI 0x011

.end
```

II. Final Output Screen Shot

Test case 1 : Storing 8 in register R0 and 5 in register R1

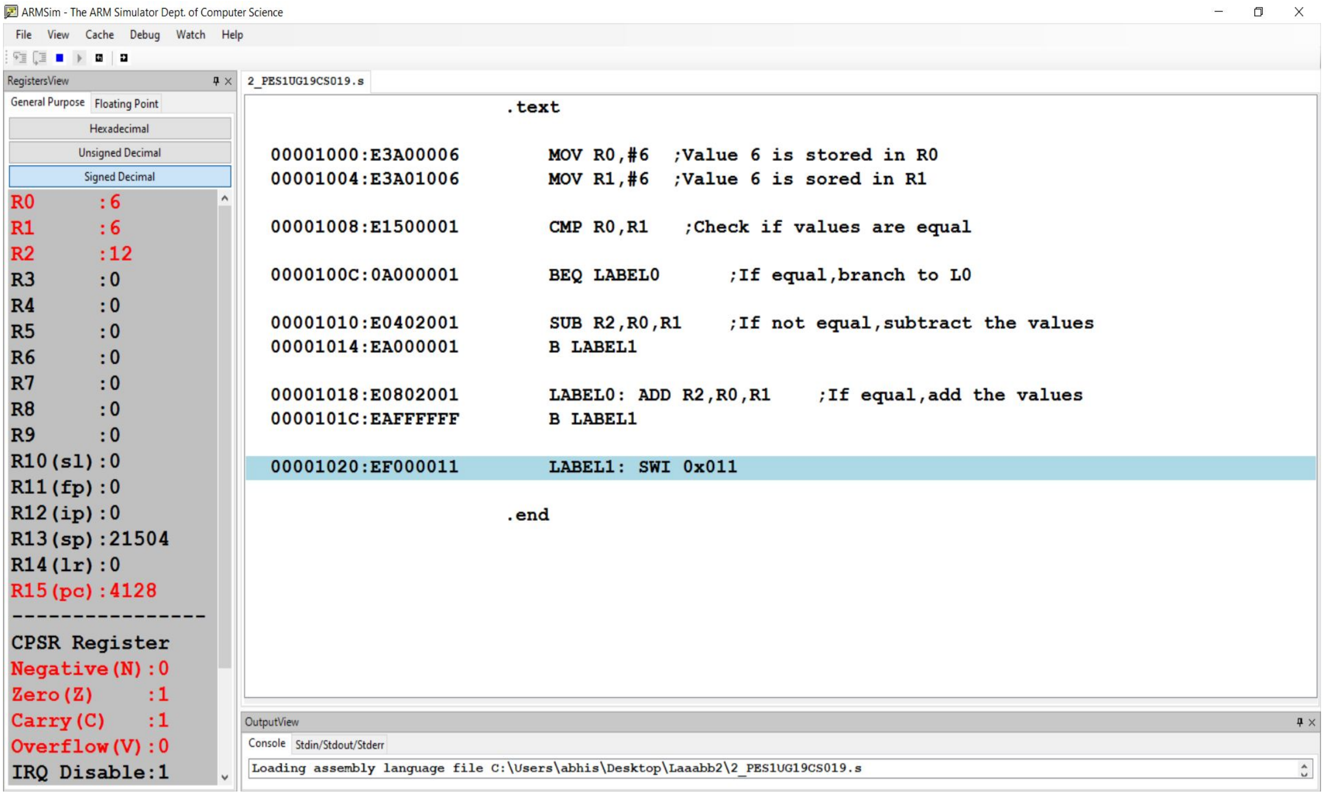

ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
2_PES1UG19CS019.s
OutputView
Console Stdin/Stdout/Stderr

```
                              .text

00001000:E3A00008       MOV R0,#8   ;Value 8 is stored in R0
00001004:E3A01005       MOV R1,#5   ;Value 5 is sored in R1

00001008:E1500001       CMP R0,R1    ;Check if values are equal

0000100C:0A000001       BEQ LABEL0      ;If equal,branch to L0

00001010:E0402001       SUB R2,R0,R1    ;If not equal,subtract the values
00001014:EA000001       B LABEL1

00001018:E0802001       LABEL0: ADD R2,R0,R1    ;If equal,add the values
0000101C:EAFFFFFF       B LABEL1

00001020:EF000011       LABEL1: SWI 0x011

                              .end
```

```
R0      :8
R1      :5
R2      :3
R3      :0
R4      :0
R5      :0
R6      :0
R7      :0
R8      :0
R9      :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4128
----------------
CPSR Register
Negative(N):0
Zero(Z)     :0
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
```

```
Loading assembly language file C:\Users\abhis\Desktop\Laaabb2\2_PES1UG19CS019.s
```

Test case 2 : Storing 6 in register R0 and 6 in register R1



```
R0       :6
R1       :6
R2       :12
R3       :0
R4       :0
R5       :0
R6       :0
R7       :0
R8       :0
R9       :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4128
----------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
```

```
                        .text

00001000:E3A00006       MOV R0,#6   ;Value 6 is stored in R0
00001004:E3A01006       MOV R1,#6   ;Value 6 is sored in R1

00001008:E1500001       CMP R0,R1    ;Check if values are equal

0000100C:0A000001       BEQ LABEL0      ;If equal,branch to L0

00001010:E0402001       SUB R2,R0,R1    ;If not equal,subtract the values
00001014:EA000001       B LABEL1

00001018:E0802001       LABEL0: ADD R2,R0,R1    ;If equal,add the values
0000101C:EAFFFFFF       B LABEL1

00001020:EF000011       LABEL1: SWI 0x011

                        .end
```

```
Loading assembly language file C:\Users\abhis\Desktop\Laaabb2\2_PES1UG19CS019.s
```

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 02/02/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	Section : A

Week#____2_______Program Number: ____3___

Title of the Program

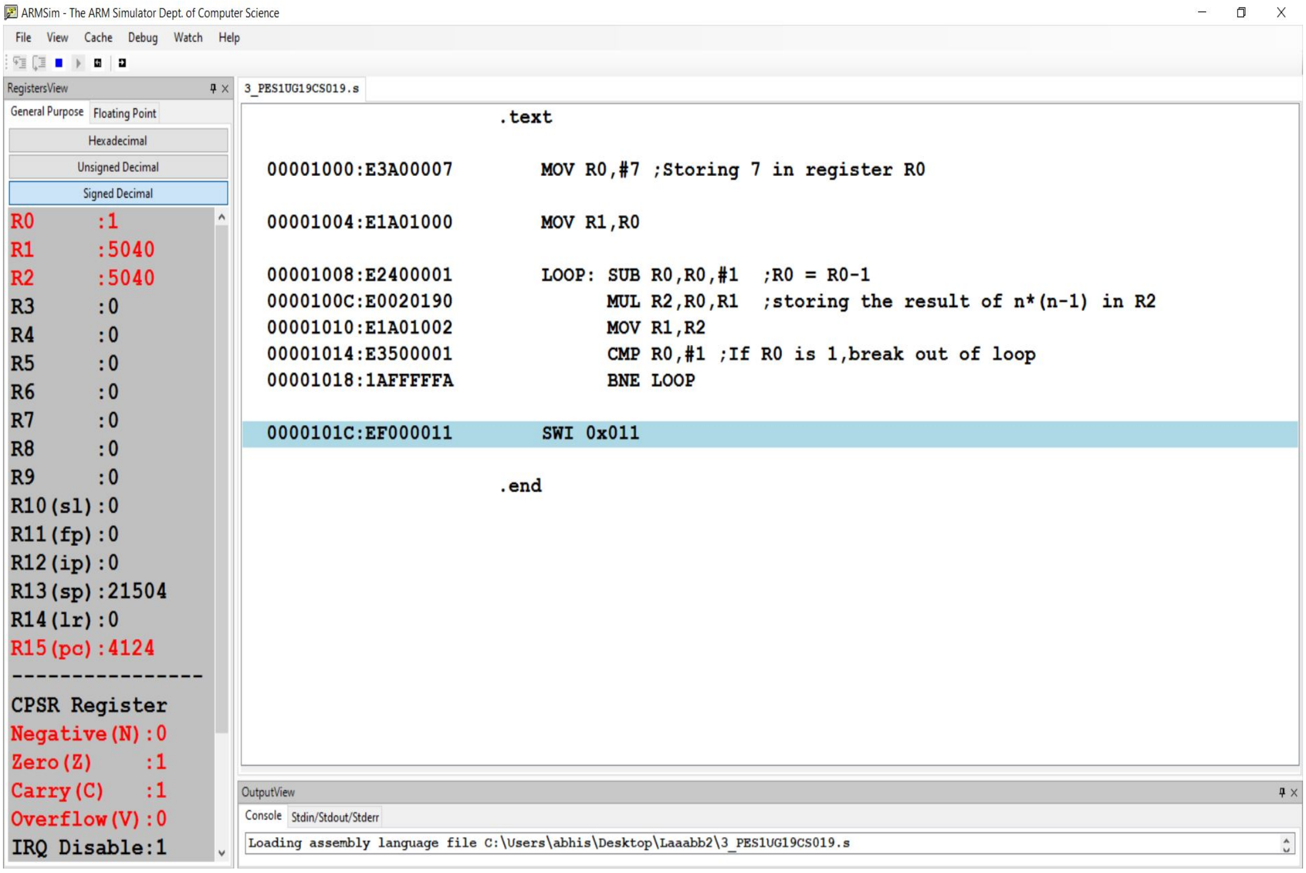
Write an ALP to find the factorial of a number stored in R0. Store the value in R1 (without using LDR and STR instructions). Use only registers.

I. ARM Assembly Code for each program

```
.text

MOV R0,#7 ;Storing 7 in register R0

MOV R1,R0

LOOP: SUB R0,R0,#1  ;R0 = R0-1

MUL R2,R0,R1 ;storing the result of n*(n-1)
in R2

MOV R1,R2

CMP R0,#1 ;If R0 is 1,break out of loop

BNE LOOP

SWI 0x011

.end
```

II.Final Output Screen Shot

Test case : 7 in register R0 (7! = 5040)



```
R0       :1
R1       :5040
R2       :5040
R3       :0
R4       :0
R5       :0
R6       :0
R7       :0
R8       :0
R9       :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4124
-----------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
```

```
                          .text

00001000:E3A00007        MOV R0,#7 ;Storing 7 in register R0

00001004:E1A01000        MOV R1,R0

00001008:E2400001        LOOP: SUB R0,R0,#1   ;R0 = R0-1
0000100C:E0020190              MUL R2,R0,R1   ;storing the result of n*(n-1) in R2
00001010:E1A01002              MOV R1,R2
00001014:E3500001              CMP R0,#1 ;If R0 is 1,break out of loop
00001018:1AFFFFFA              BNE LOOP

0000101C:EF000011        SWI 0x011

                          .end
```

```
Loading assembly language file C:\Users\abhis\Desktop\Laaabb2\3_PES1UG19CS019.s
```

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 02/02/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	Section : A

Week#____2_______Program Number: ____4___

Title of the Program

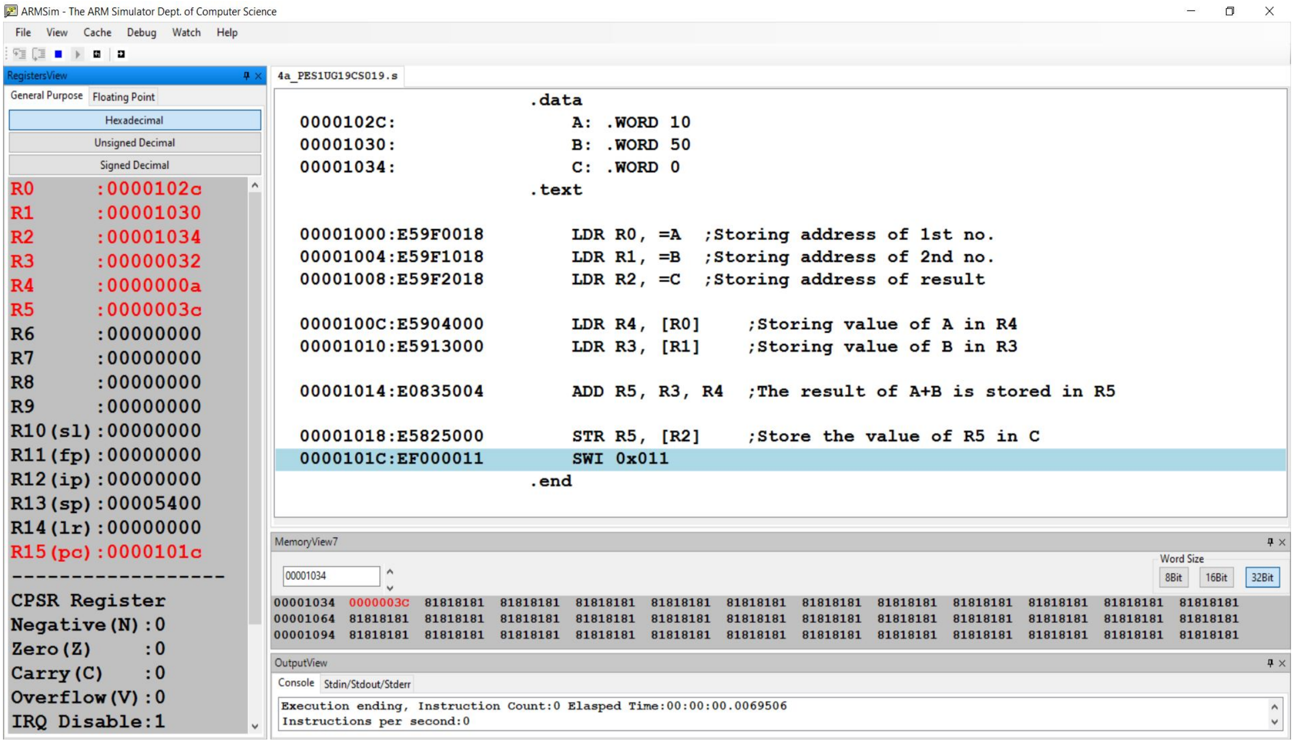
a) Write an ALP to add two 32 bit numbers loaded from memory and store the result in memory.

I. ARM Assembly Code for each program

```
.data

    A:  .WORD 10

    B:  .WORD 50

    C:  .WORD 0

.text

    LDR R0, =A  ;Storing address of 1st no.

    LDR R1, =B  ;Storing address of 2nd no.

    LDR R2, =C  ;Storing address of result

    LDR R4, [R0]    ;Storing value of A in R4

    LDR R3, [R1]    ;Storing value of B in R3

    ADD R5, R3, R4  ;The result of A+B is
stored in R5

    STR R5, [R2]    ;Store the value of R5 in C

    SWI 0x011

.end
```

II. Final Output Screen Shot

Test case : A =10 and B=50



```
R0       :0000102c
R1       :00001030
R2       :00001034
R3       :00000032
R4       :0000000a
R5       :0000003c
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):0000101c
------------------
CPSR Register
Negative(N):0
Zero(Z)     :0
Carry(C)    :0
Overflow(V):0
IRQ Disable:1
```

```
                               .data
    0000102C:                      A: .WORD 10
    00001030:                      B: .WORD 50
    00001034:                      C: .WORD 0
                               .text

    00001000:E59F0018          LDR R0, =A  ;Storing address of 1st no.
    00001004:E59F1018          LDR R1, =B  ;Storing address of 2nd no.
    00001008:E59F2018          LDR R2, =C  ;Storing address of result

    0000100C:E5904000          LDR R4, [R0]    ;Storing value of A in R4
    00001010:E5913000          LDR R3, [R1]    ;Storing value of B in R3

    00001014:E0835004          ADD R5, R3, R4  ;The result of A+B is stored in R5

    00001018:E5825000          STR R5, [R2]    ;Store the value of R5 in C
    0000101C:EF000011          SWI 0x011
                               .end
```

```
00001034
00001034 0000003C 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001064 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001094 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
```

```
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0069506
Instructions per second:0
```

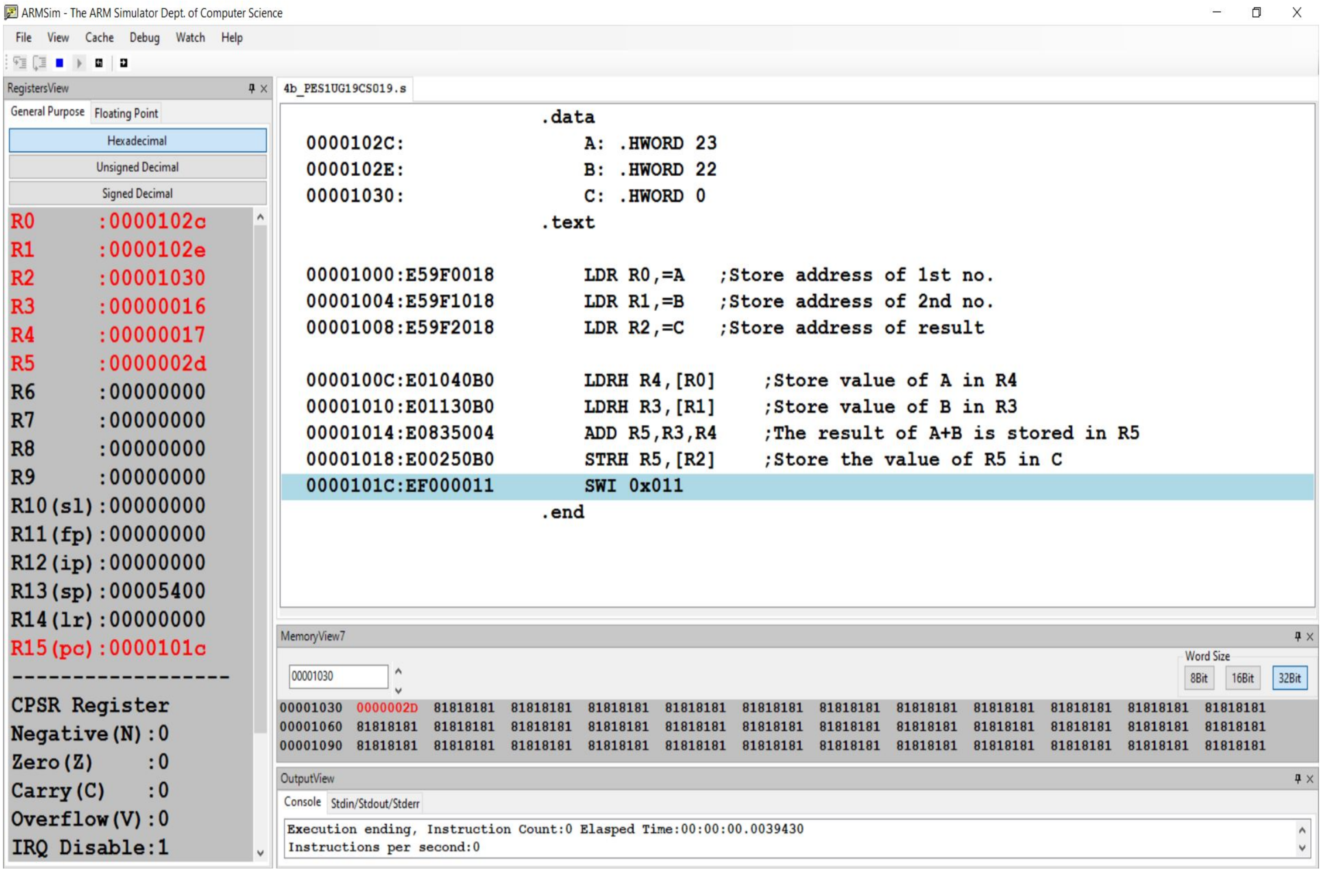
b) Write an ALP to add two 16 bit numbers loaded from memory and store the result in memory.

I. ARM Assembly Code for each program

```
.data

    A:  .HWORD 23

    B:  .HWORD 22

    C:  .HWORD 0

.text

    LDR R0,=A    ;Store address of 1st no.

    LDR R1,=B    ;Store address of 2nd no.

    LDR R2,=C    ;Store address of result

    LDRH R4,[R0]     ;Store value of A in R4
    LDRH R3,[R1]     ;Store value of B in R3

   ADD R5,R3,R4     ;The result of A+B is stored
in R5

    STRH R5,[R2]     ;Store the value of R5 in C

    SWI 0x011

.end
```

II.Final Output Screen Shot

Test case : A =23 and B=22



```
R0        :0000102c
R1        :0000102e
R2        :00001030
R3        :00000016
R4        :00000017
R5        :0000002d
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):0000101c
------------------
CPSR Register
Negative(N):0
Zero(Z)    :0
Carry(C)   :0
Overflow(V):0
IRQ Disable:1
```

```
                      .data
0000102C:                   A: .HWORD 23
0000102E:                   B: .HWORD 22
00001030:                   C: .HWORD 0
                      .text

00001000:E59F0018           LDR R0,=A     ;Store address of 1st no.
00001004:E59F1018           LDR R1,=B     ;Store address of 2nd no.
00001008:E59F2018           LDR R2,=C     ;Store address of result

0000100C:E01040B0           LDRH R4,[R0]     ;Store value of A in R4
00001010:E01130B0           LDRH R3,[R1]     ;Store value of B in R3
00001014:E0835004           ADD R5,R3,R4     ;The result of A+B is stored in R5
00001018:E00250B0           STRH R5,[R2]     ;Store the value of R5 in C
0000101C:EF000011           SWI 0x011
                      .end
```

```
00001030
00001030  0000002D  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001060  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001090  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0039430
Instructions per second:0
```

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 02/02/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	Section : A

Week#____2_______Program Number: ____5___

Title of the Program

a) Write an ALP to find GCD of two numbers (without using LDR and STR instructions). Both numbers are in registers. Use only registers.

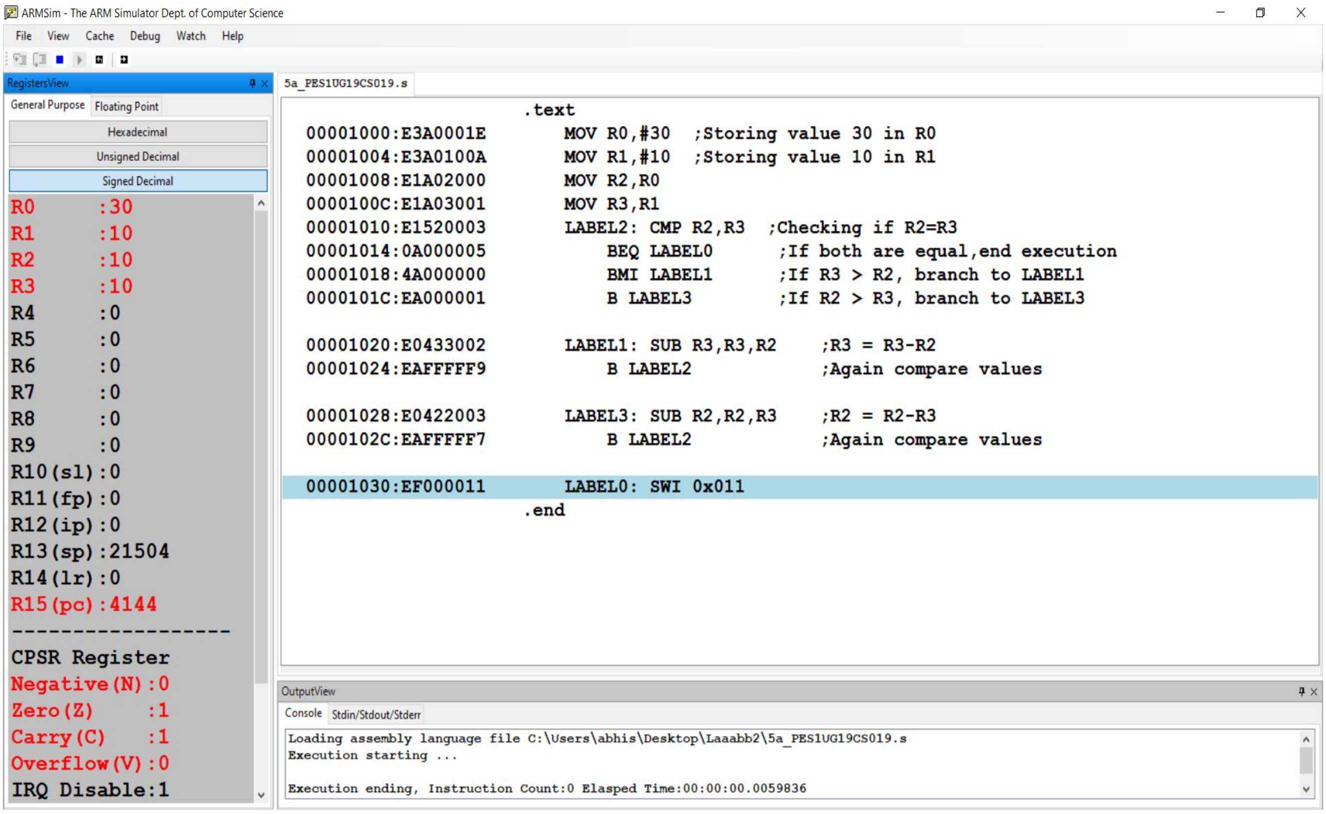
I. ARM Assembly Code for each program

```
.text

 MOV R0,#30  ;Storing value 30 in R0

 MOV R1,#10  ;Storing value 10 in R1

 MOV R2,R0

 MOV R3,R1

 LABEL2: CMP R2,R3  ;Checking if R2=R3

 BEQ LABEL0 ;If both are equal,end execution
```

```
  BMI LABEL1          ;If R3 > R2, branch to LABEL1
  B LABEL3            ;If R2 > R3, branch to LABEL3
  LABEL1: SUB R3,R3,R2     ;R3 = R3-R2
     B LABEL2             ;Again compare values
  LABEL3: SUB R2,R2,R3     ;R2 = R2-R3
     B LABEL2             ;Again compare values
  LABEL0: SWI 0x011
.end
```

II. Final Output Screen Shot

Test case : 30 in R0 and 10 in R1 (gcd of 10 & 30)



ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
5a_PES1UG19CS019.s
OutputView
Console Stdin/Stdout/Stderr

```
R0      :30
R1      :10
R2      :10
R3      :10
R4      :0
R5      :0
R6      :0
R7      :0
R8      :0
R9      :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4144
------------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
```

```
                        .text
00001000:E3A0001E           MOV R0,#30  ;Storing value 30 in R0
00001004:E3A0100A           MOV R1,#10  ;Storing value 10 in R1
00001008:E1A02000           MOV R2,R0
0000100C:E1A03001           MOV R3,R1
00001010:E1520003           LABEL2: CMP R2,R3  ;Checking if R2=R3
00001014:0A000005               BEQ LABEL0      ;If both are equal,end execution
00001018:4A000000               BMI LABEL1      ;If R3 > R2, branch to LABEL1
0000101C:EA000001               B LABEL3        ;If R2 > R3, branch to LABEL3

00001020:E0433002           LABEL1: SUB R3,R3,R2    ;R3 = R3-R2
00001024:EAFFFFF9               B LABEL2            ;Again compare values

00001028:E0422003           LABEL3: SUB R2,R2,R3    ;R2 = R2-R3
0000102C:EAFFFFF7               B LABEL2            ;Again compare values

00001030:EF000011           LABEL0: SWI 0x011
                        .end
```

```
Loading assembly language file C:\Users\abhis\Desktop\Laaabb2\5a_PES1UG19CS019.s
Execution starting ...

Execution ending, Instruction Count:0 Elasped Time:00:00:00.0059836
```

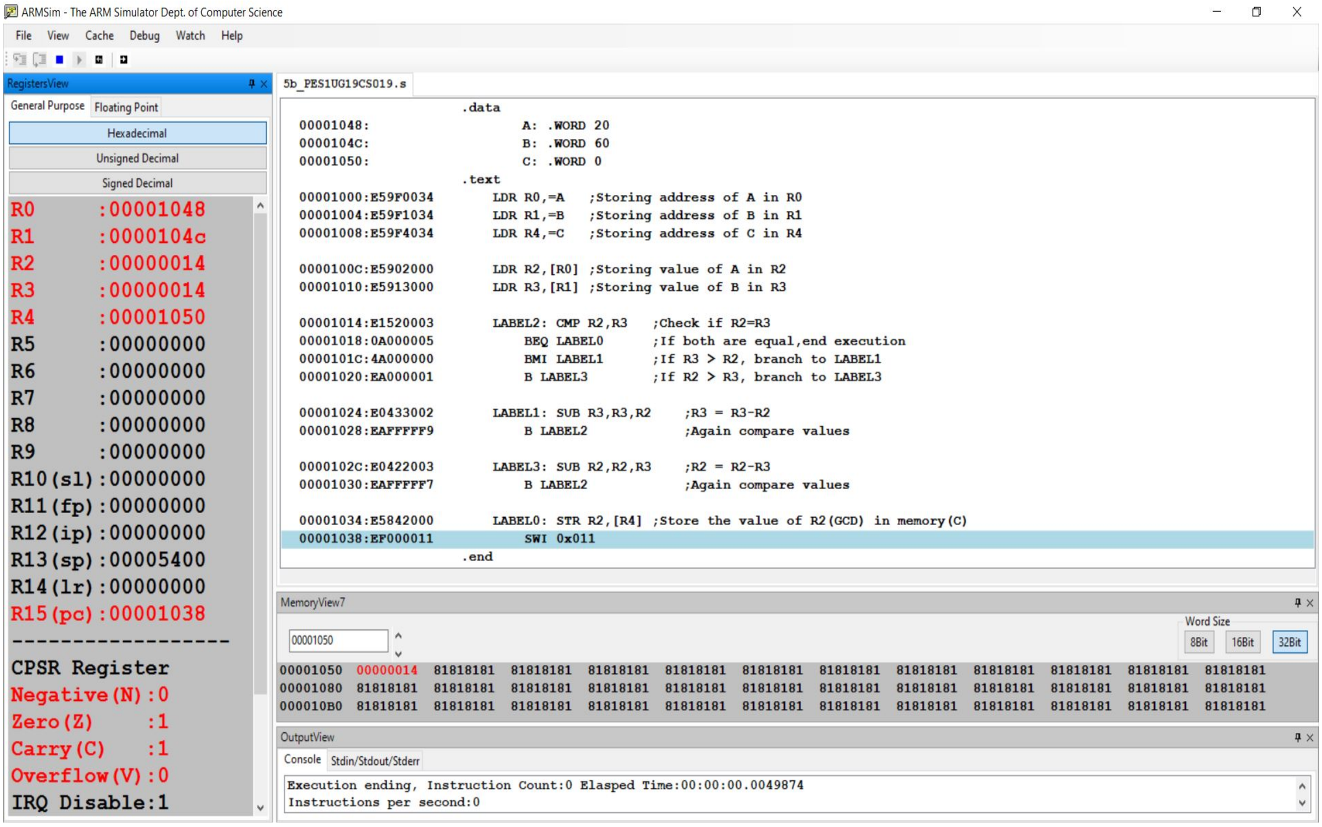
b) Write an ALP to find the GCD of given numbers (both numbers in memory) Store result in memory.

I. ARM Assembly Code for each program

```
.data
    A: .WORD 20
    B: .WORD 60
    C: .WORD 0
.text
    LDR R0,=A   ;Storing address of A in R0
    LDR R1,=B   ;Storing address of B in R1
    LDR R4,=C   ;Storing address of C in R4
    LDR R2,[R0] ;Storing value of A in R2
    LDR R3,[R1] ;Storing value of B in R3
    LABEL2: CMP R2,R3   ;Check if R2=R3
    BEQ LABEL0 ;If both are equal,end execution
    BMI LABEL1  ;If R3 > R2, branch to Label1
    B LABEL3     ;If R2 > R3, branch to Label3

    LABEL1: SUB R3,R3,R2    ;R3 = R3-R2
        B LABEL2   ;Again compare values
    LABEL3: SUB R2,R2,R3     ;R2 = R2-R3
        B LABEL2    ;Again compare values
    LABEL0:  STR  R2,[R4]  ;Store  the  value  of R2(GCD)  in memory(C)
        SWI 0x011
.end
```

II. Final Output Screen Shot

Test case 1 : 20 in R2 and 60 in R3 (A<B)



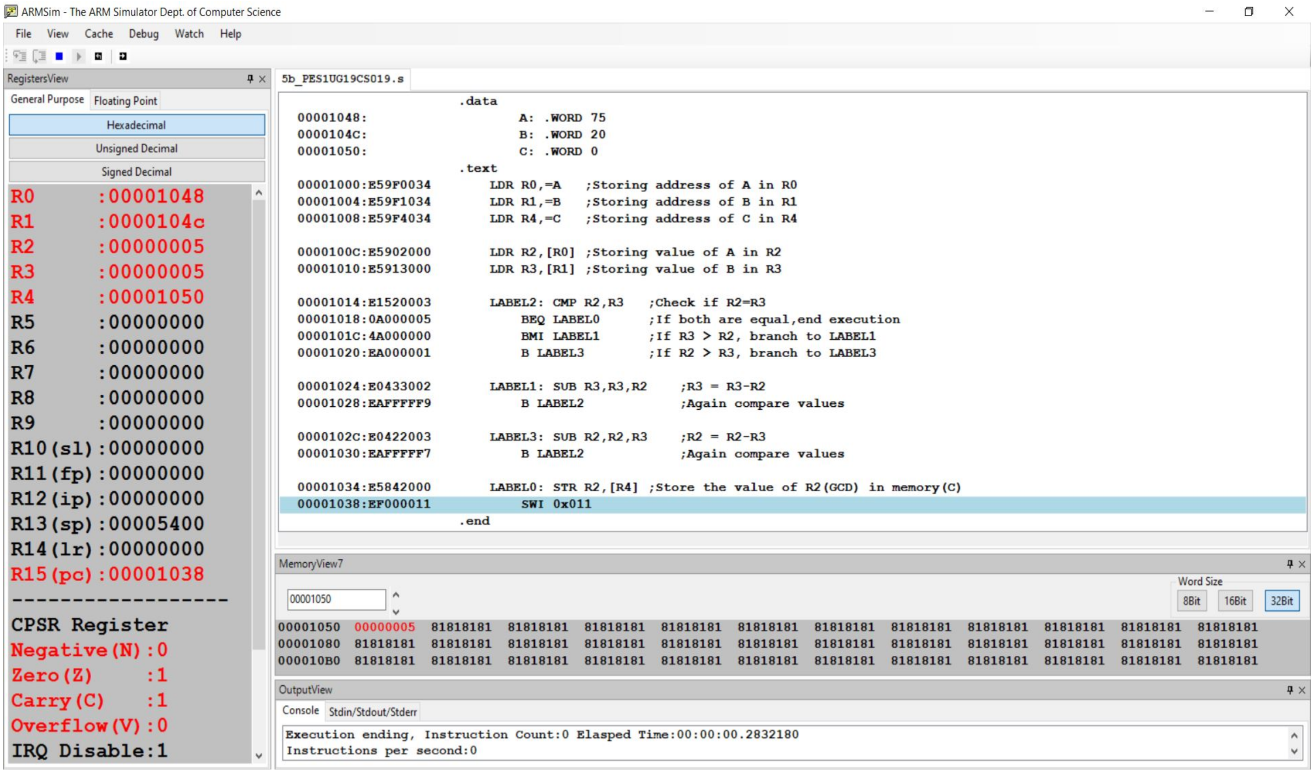
```
R0        :00001048
R1        :0000104c
R2        :00000014
R3        :00000014
R4        :00001050
R5        :00000000
R6        :00000000
R7        :00000000
R8        :00000000
R9        :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):00001038
------------------
CPSR Register
Negative(N):0
Zero(Z)      :1
Carry(C)     :1
Overflow(V):0
IRQ Disable:1
```

```
                        .data
00001048:                    A: .WORD 20
0000104C:                    B: .WORD 60
00001050:                    C: .WORD 0
                        .text
00001000:E59F0034        LDR R0,=A   ;Storing address of A in R0
00001004:E59F1034        LDR R1,=B   ;Storing address of B in R1
00001008:E59F4034        LDR R4,=C   ;Storing address of C in R4

0000100C:E5902000        LDR R2,[R0] ;Storing value of A in R2
00001010:E5913000        LDR R3,[R1] ;Storing value of B in R3

00001014:E1520003        LABEL2: CMP R2,R3   ;Check if R2=R3
00001018:0A000005            BEQ LABEL0      ;If both are equal,end execution
0000101C:4A000000            BMI LABEL1      ;If R3 > R2, branch to LABEL1
00001020:EA000001            B LABEL3        ;If R2 > R3, branch to LABEL3

00001024:E0433002        LABEL1: SUB R3,R3,R2    ;R3 = R3-R2
00001028:EAFFFFF9            B LABEL2            ;Again compare values

0000102C:E0422003        LABEL3: SUB R2,R2,R3    ;R2 = R2-R3
00001030:EAFFFFF7            B LABEL2            ;Again compare values

00001034:E5842000        LABEL0: STR R2,[R4] ;Store the value of R2(GCD) in memory(C)
00001038:EF000011            SWI 0x011
                        .end
```

```
00001050
00001050  00000014  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001080  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010B0  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0049874
Instructions per second:0
```

Test case 2 : 75 in R2 and 20 in R3 (A>B)



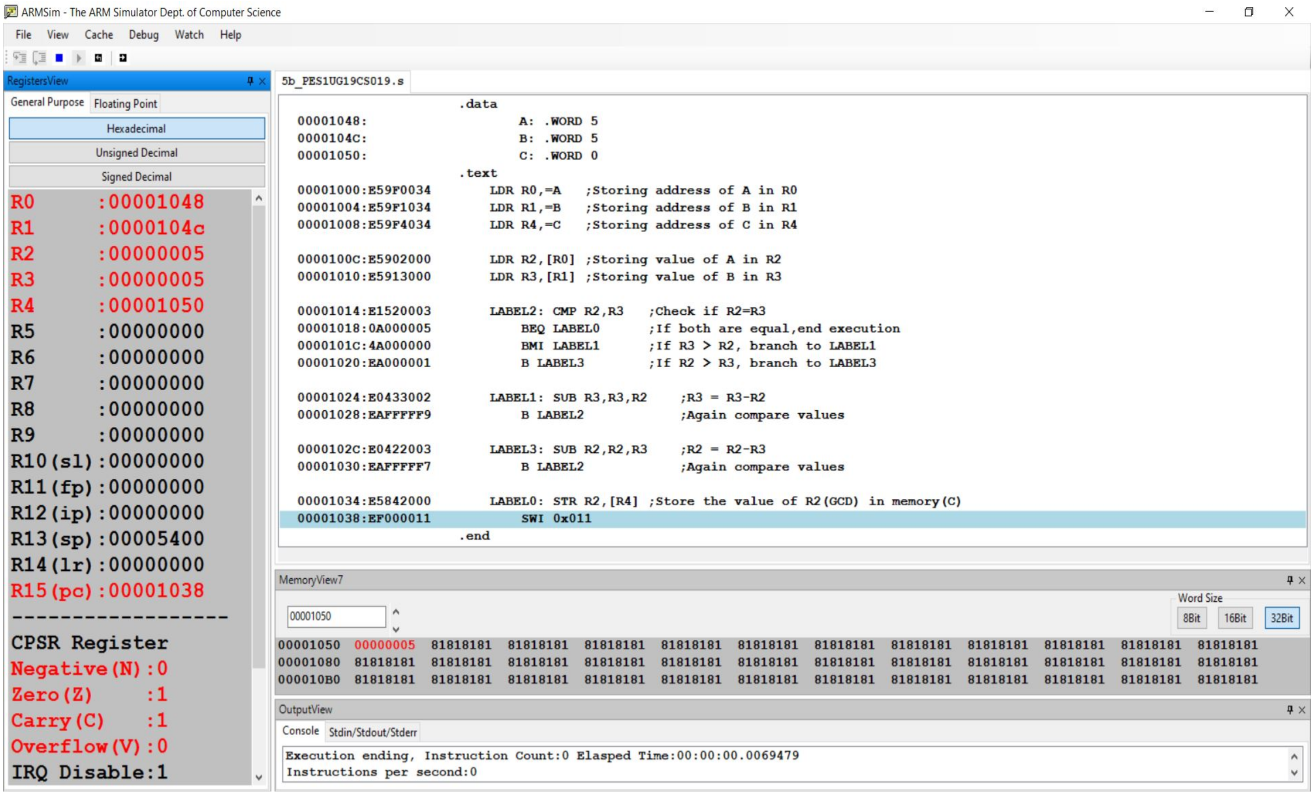
```
R0       :00001048
R1       :0000104c
R2       :00000005
R3       :00000005
R4       :00001050
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):00001038
------------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
```

```
                        .data
  00001048:                     A: .WORD 75
  0000104C:                     B: .WORD 20
  00001050:                     C: .WORD 0
                        .text
  00001000:E59F0034         LDR R0,=A    ;Storing address of A in R0
  00001004:E59F1034         LDR R1,=B    ;Storing address of B in R1
  00001008:E59F4034         LDR R4,=C    ;Storing address of C in R4

  0000100C:E5902000         LDR R2,[R0] ;Storing value of A in R2
  00001010:E5913000         LDR R3,[R1] ;Storing value of B in R3

  00001014:E1520003         LABEL2: CMP R2,R3    ;Check if R2=R3
  00001018:0A000005             BEQ LABEL0       ;If both are equal,end execution
  0000101C:4A000000             BMI LABEL1       ;If R3 > R2, branch to LABEL1
  00001020:EA000001             B LABEL3         ;If R2 > R3, branch to LABEL3

  00001024:E0433002         LABEL1: SUB R3,R3,R2     ;R3 = R3-R2
  00001028:EAFFFFF9             B LABEL2             ;Again compare values

  0000102C:E0422003         LABEL3: SUB R2,R2,R3     ;R2 = R2-R3
  00001030:EAFFFFF7             B LABEL2             ;Again compare values

  00001034:E5842000         LABEL0: STR R2,[R4] ;Store the value of R2(GCD) in memory(C)
  00001038:EF000011             SWI 0x011
                        .end
```

```
00001050
00001050  00000005  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001080  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010B0  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
Execution ending, Instruction Count:0 Elasped Time:00:00:00.2832180
Instructions per second:0
```

Test case 3 : 5 in R2 and 5 in R3 (A=B)



```
R0       :00001048
R1       :0000104c
R2       :00000005
R3       :00000005
R4       :00001050
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):00001038
------------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
```

```
                        .data
  00001048:                     A: .WORD 5
  0000104C:                     B: .WORD 5
  00001050:                     C: .WORD 0
                        .text
  00001000:E59F0034         LDR R0,=A    ;Storing address of A in R0
  00001004:E59F1034         LDR R1,=B    ;Storing address of B in R1
  00001008:E59F4034         LDR R4,=C    ;Storing address of C in R4

  0000100C:E5902000         LDR R2,[R0] ;Storing value of A in R2
  00001010:E5913000         LDR R3,[R1] ;Storing value of B in R3

  00001014:E1520003         LABEL2: CMP R2,R3    ;Check if R2=R3
  00001018:0A000005             BEQ LABEL0       ;If both are equal,end execution
  0000101C:4A000000             BMI LABEL1       ;If R3 > R2, branch to LABEL1
  00001020:EA000001             B LABEL3         ;If R2 > R3, branch to LABEL3

  00001024:E0433002         LABEL1: SUB R3,R3,R2     ;R3 = R3-R2
  00001028:EAFFFFF9             B LABEL2             ;Again compare values

  0000102C:E0422003         LABEL3: SUB R2,R2,R3     ;R2 = R2-R3
  00001030:EAFFFFF7             B LABEL2             ;Again compare values

  00001034:E5842000         LABEL0: STR R2,[R4] ;Store the value of R2(GCD) in memory(C)
  00001038:EF000011             SWI 0x011
                        .end
```

```
00001050
00001050  00000005  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001080  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010B0  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0069479
Instructions per second:0
```

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 02/02/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	Section : A

Week#____2_______Program Number: ____6___

Title of the Program

a) Write an ALP to add an array of ten 32 bit numbers from memory.

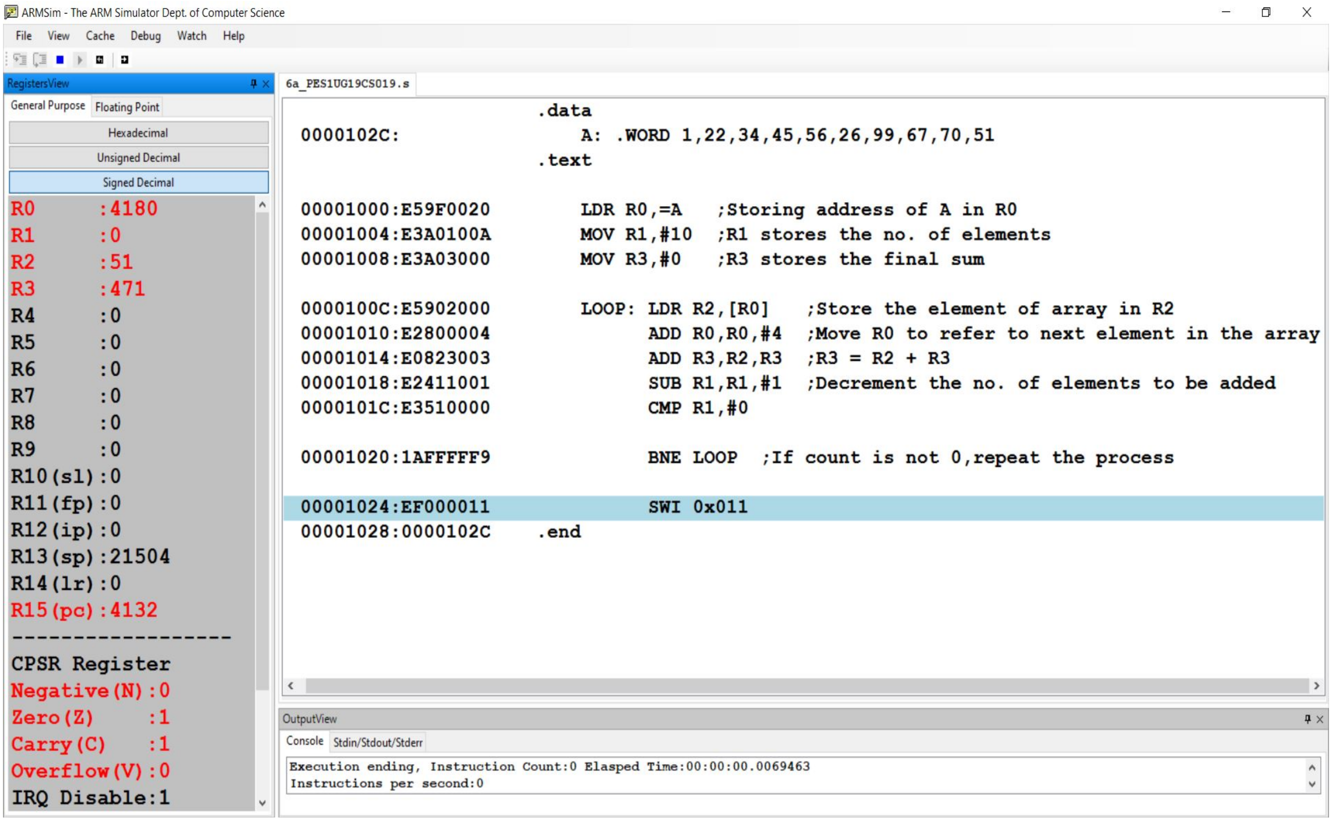
I. ARM Assembly Code for each program

```
.data
    A: .WORD 1,22,34,45,56,26,99,67,70,51
.text
    LDR R0,=A   ;Storing address of A in R0
    MOV R1,#10  ;R1 stores the no. of elements
    MOV R3,#0   ;R3 stores the final sum
```

```
  LOOP: LDR R2,[R0] ;Store the element of array in R2

    ADD  R0,R0,#4    ;Move  R0  to  refer  to  next element in the array

    ADD R3,R2,R3   ;R3 = R2 + R3

    SUB R1,R1,#1   ;Decrement the no. of elements to be added

    CMP R1,#0

    BNE  LOOP    ;If  count  is  not  0,repeat  the process

     SWI 0x011

.end
```

II.Final Output Screen Shot

Test case : Array of ten 32 bit numbers
1,22,34,45,56,26,99,67,70,51



```
R0       :4180
R1       :0
R2       :51
R3       :471
R4       :0
R5       :0
R6       :0
R7       :0
R8       :0
R9       :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4132
------------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
```

```
                        .data
0000102C:                  A:  .WORD 1,22,34,45,56,26,99,67,70,51
                        .text

00001000:E59F0020          LDR R0,=A    ;Storing address of A in R0
00001004:E3A0100A          MOV R1,#10   ;R1 stores the no. of elements
00001008:E3A03000          MOV R3,#0    ;R3 stores the final sum

0000100C:E5902000          LOOP: LDR R2,[R0]    ;Store the element of array in R2
00001010:E2800004                ADD R0,R0,#4   ;Move R0 to refer to next element in the array
00001014:E0823003                ADD R3,R2,R3   ;R3 = R2 + R3
00001018:E2411001                SUB R1,R1,#1   ;Decrement the no. of elements to be added
0000101C:E3510000                CMP R1,#0

00001020:1AFFFFF9                BNE LOOP   ;If count is not 0,repeat the process

00001024:EF000011                SWI 0x011
00001028:0000102C       .end
```

```
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0069463
Instructions per second:0
```

b) Write an ALP to add array of ten 8 bit numbers taking data from memory location stored as byte data (use .byte to store the data instead of .word)

I. ARM Assembly Code for each program

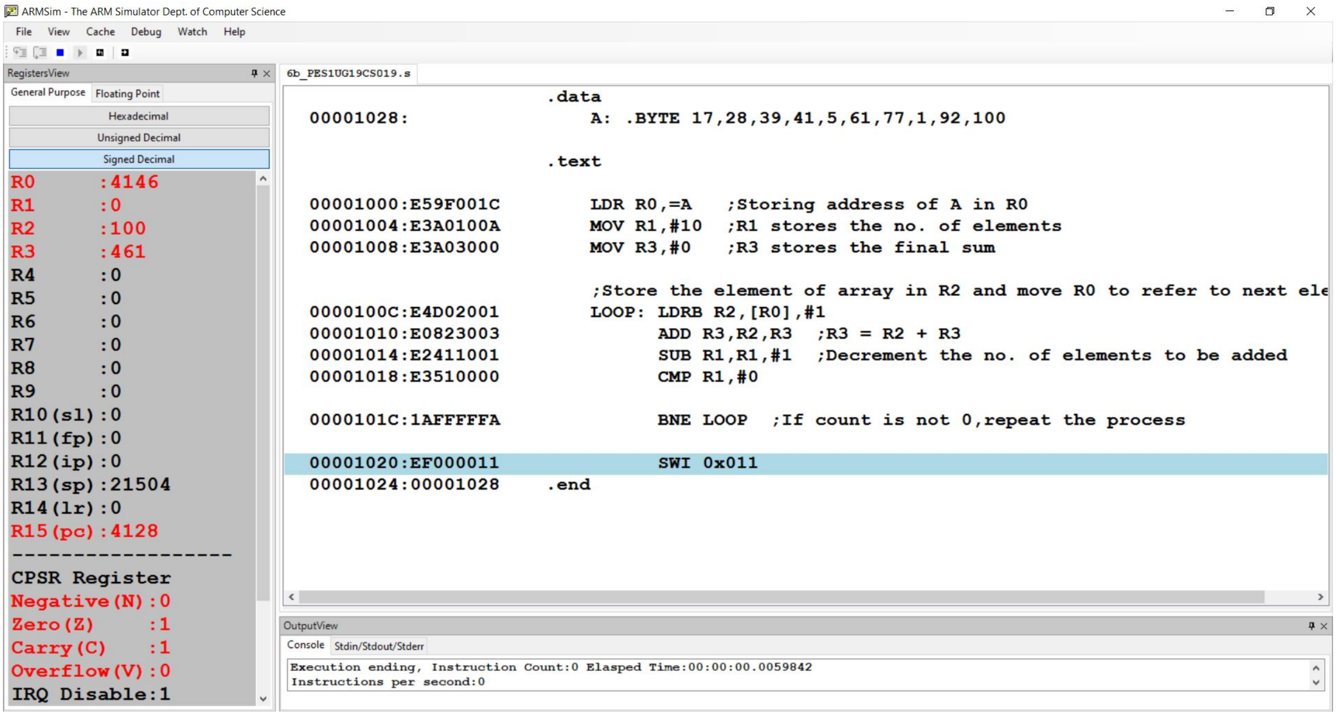
```
.data

    A:  .BYTE 17,28,39,41,5,61,77,1,92,100

.text

    LDR R0,=A    ;Storing address of A in R0

    MOV R1,#10   ;R1 stores the no. of elements

    MOV R3,#0    ;R3 stores the final sum
```

```
;Store the element of array in R2 and move R0
to refer to next element

LOOP: LDRB R2,[R0],#1

      ADD R3,R2,R3  ;R3 = R2 + R3

      SUB R1,R1,#1  ;Decrement the no. of
elements to be added

      CMP R1,#0

      BNE LOOP  ;If count is not 0,repeat the
process

      SWI 0x011

.end
```

II. Final Output Screen Shot

Test case : Array of ten 8 bit numbers

17,28,39,41,5,61,77,1,92,100



```
R0       :4146
R1       :0
R2       :100
R3       :461
R4       :0
R5       :0
R6       :0
R7       :0
R8       :0
R9       :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4128
-----------------
CPSR Register
Negative(N):0
Zero(Z)     :1
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
```

```
                          .data
00001028:                     A: .BYTE 17,28,39,41,5,61,77,1,92,100

                          .text

00001000:E59F001C             LDR R0,=A    ;Storing address of A in R0
00001004:E3A0100A             MOV R1,#10   ;R1 stores the no. of elements
00001008:E3A03000             MOV R3,#0    ;R3 stores the final sum

                              ;Store the element of array in R2 and move R0 to refer to next ele
0000100C:E4D02001             LOOP: LDRB R2,[R0],#1
00001010:E0823003                   ADD R3,R2,R3  ;R3 = R2 + R3
00001014:E2411001                   SUB R1,R1,#1  ;Decrement the no. of elements to be added
00001018:E3510000                   CMP R1,#0

0000101C:1AFFFFFA                   BNE LOOP  ;If count is not 0,repeat the process

00001020:EF000011                   SWI 0x011
00001024:00001028         .end
```

```
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0059842
Instructions per second:0
```

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 02/02/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	Section : A

Week#____2_______Program Number: ____7___

Title of the Program

Write an ALP to multiply using a barrel shifter.

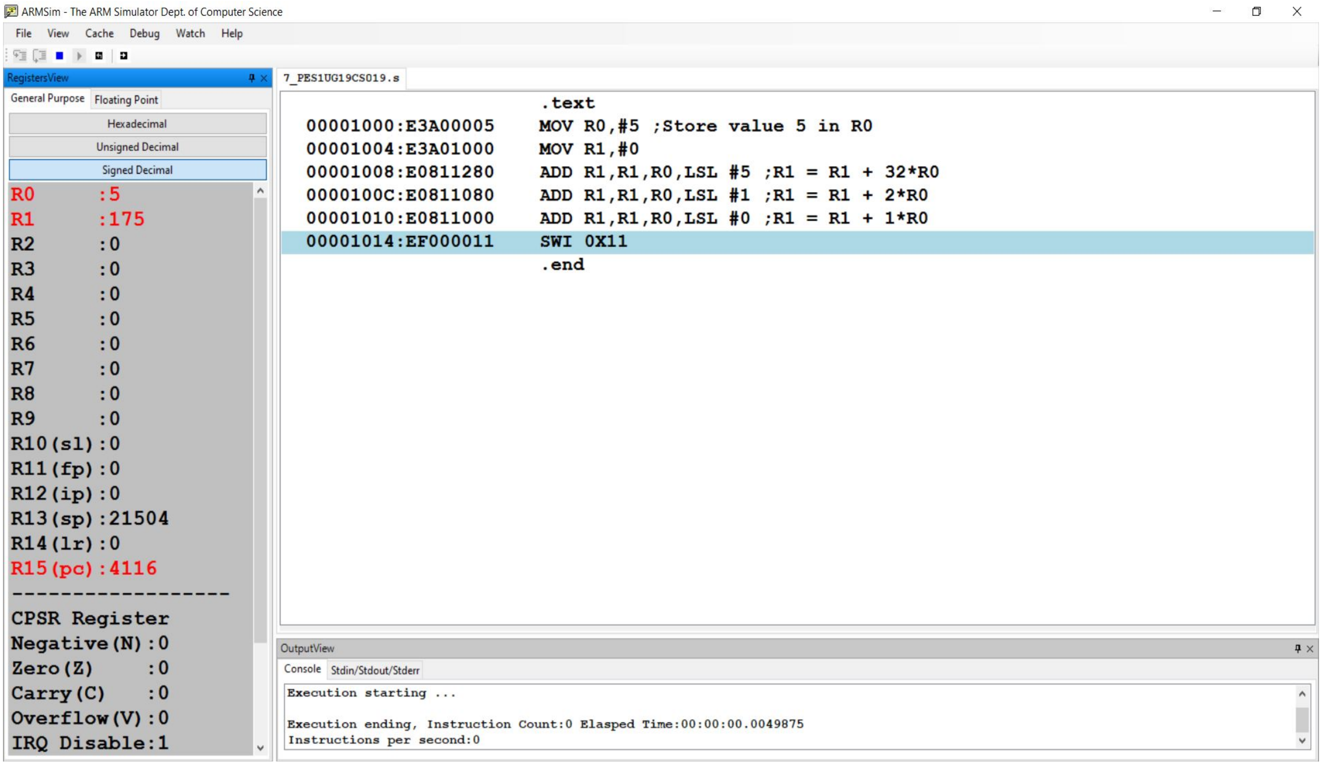
35*R0

I. ARM Assembly Code for each program

```
.text

MOV R0,#5 ;Store value 5 in R0

MOV R1,#0

ADD R1,R1,R0,LSL #5 ;R1 = R1 + 32*R0

ADD R1,R1,R0,LSL #1 ;R1 = R1 + 2*R0

ADD R1,R1,R0,LSL #0 ;R1 = R1 + 1*R0

SWI 0X11

.end
```

II. Final Output Screen Shot

Test case : 5 in R0 (35 * 5)



```
R0       :5
R1       :175
R2       :0
R3       :0
R4       :0
R5       :0
R6       :0
R7       :0
R8       :0
R9       :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4116
------------------
CPSR Register
Negative(N):0
Zero(Z)     :0
Carry(C)    :0
Overflow(V):0
IRQ Disable:1
```

```
                           .text
00001000:E3A00005          MOV R0,#5 ;Store value 5 in R0
00001004:E3A01000          MOV R1,#0
00001008:E0811280          ADD R1,R1,R0,LSL #5 ;R1 = R1 + 32*R0
0000100C:E0811080          ADD R1,R1,R0,LSL #1 ;R1 = R1 + 2*R0
00001010:E0811000          ADD R1,R1,R0,LSL #0 ;R1 = R1 + 1*R0
00001014:EF000011          SWI 0X11
                           .end
```

```
Execution starting ...

Execution ending, Instruction Count:0 Elasped Time:00:00:00.0049875
Instructions per second:0
```

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

Date: 02/02/2021

Name : Abhishek Aditya BS	SRN: PES1UG19CS019	Section : A

Week#____2_______Program Number: ____8___

Title of the Program

Write an ALP to evaluate the expression (A+B) + (3*B), where A and B are memory locations.

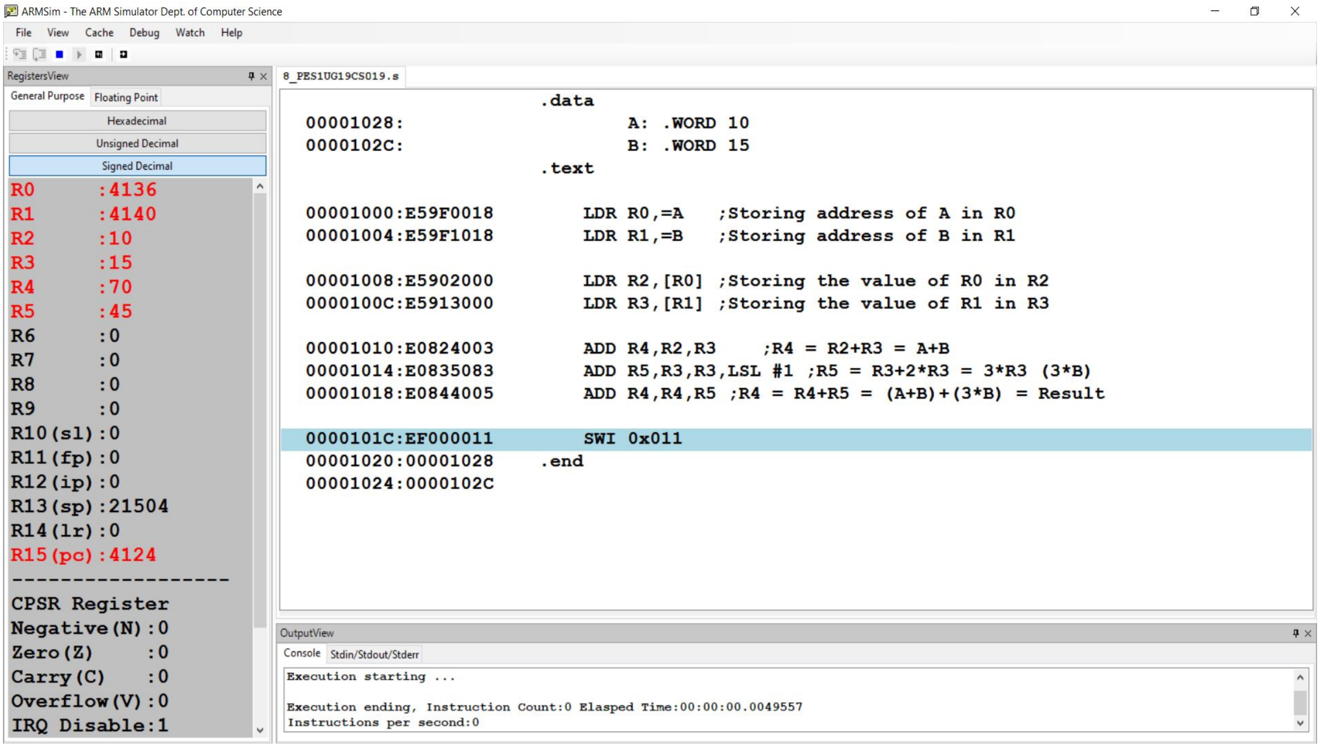
I. ARM Assembly Code for each program

```
.data

    A: .WORD 10

    B: .WORD 15

.text


    LDR R0,=A   ;Storing address of A in R0

    LDR R1,=B   ;Storing address of B in R1


    LDR R2,[R0] ;Storing the value of R0 in R2

    LDR R3,[R1] ;Storing the value of R1 in R3


    ADD R4,R2,R3    ;R4 = R2+R3 = A+B

    ADD R5,R3,R3,LSL #1 ;R5 = R3+2*R3 = 3*R3 (3*B)

   ADD R4,R4,R5 ;R4 = R4+R5 = (A+B)+(3*B) = Result


    SWI 0x011

.end
```

II. Final Output Screen Shot

Test case : A = 10 , B = 15

(A+B)+(3*B) = (10+15) + (3*15) = 70



```
R0        :4136
R1        :4140
R2        :10
R3        :15
R4        :70
R5        :45
R6        :0
R7        :0
R8        :0
R9        :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4124
------------------
CPSR Register
Negative(N):0
Zero(Z)    :0
Carry(C)   :0
Overflow(V):0
IRQ Disable:1
```

```
                              .data
00001028:                            A: .WORD 10
0000102C:                            B: .WORD 15
                              .text

00001000:E59F0018         LDR R0,=A    ;Storing address of A in R0
00001004:E59F1018         LDR R1,=B    ;Storing address of B in R1

00001008:E5902000         LDR R2,[R0]  ;Storing the value of R0 in R2
0000100C:E5913000         LDR R3,[R1]  ;Storing the value of R1 in R3

00001010:E0824003         ADD R4,R2,R3     ;R4 = R2+R3 = A+B
00001014:E0835083         ADD R5,R3,R3,LSL #1  ;R5 = R3+2*R3 = 3*R3  (3*B)
00001018:E0844005         ADD R4,R4,R5  ;R4 = R4+R5 = (A+B)+(3*B) = Result

0000101C:EF000011         SWI 0x011
00001020:00001028     .end
00001024:0000102C
```

```
Execution starting ...

Execution ending, Instruction Count:0 Elasped Time:00:00:00.0049557
Instructions per second:0
```

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 : Abhishek Aditya BS Date : 02/02/2021

Name : Abhishek Aditya BS

SRN : PES1UG19CS019

Section : A