Microprocessor and Computer Architecture

UE22CS251B

4th Semester, Academic Year 2023-24

Date:

Name: Amogh P	SRN:PES2UG22CS062	Section: B

LAB #____2_____ Program Number: ____1___

Title of the Program

Write a program to add two numbers by reading them from memory and store the result back to the memory.

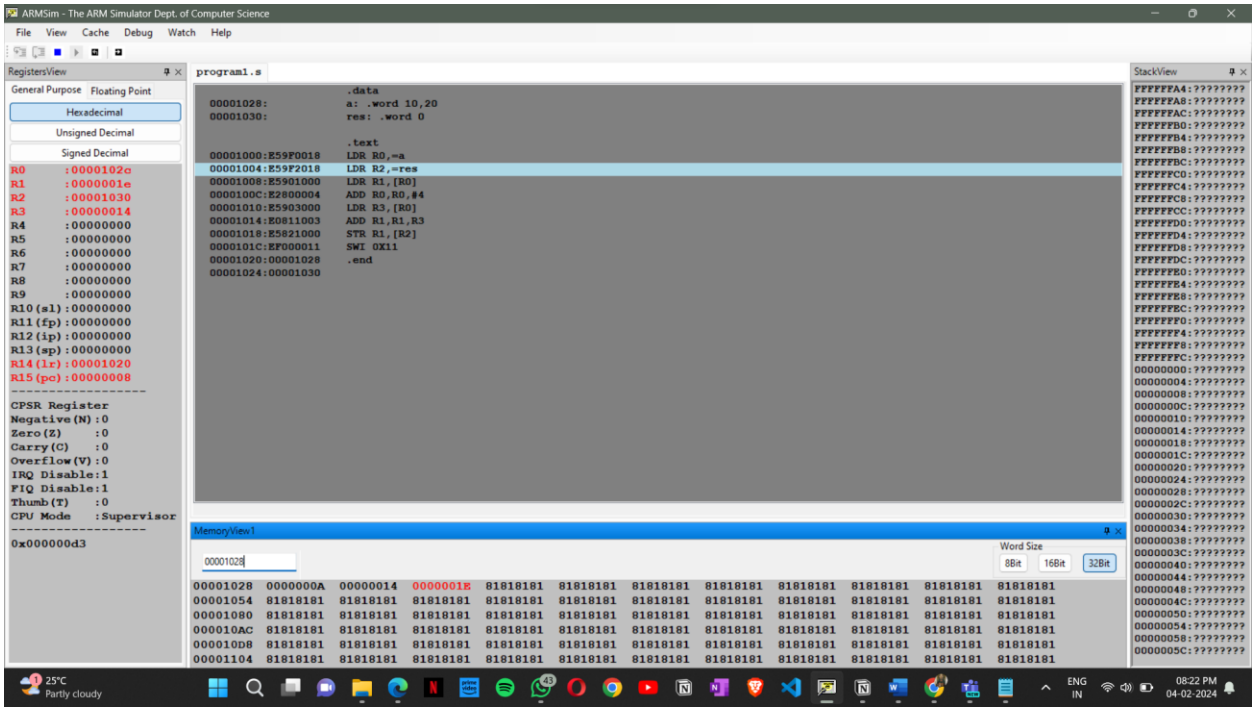
1. ARM Assembly Code

CODE:

```
.data
a: .word 10,20
res: .word 0
.text
LDR R0,=a
LDR R2,=res
LDR R1,[R0]
ADD R0,R0,#4
LDR R3,[R0]
ADD R1,R1,R3
STR R1,[R2]
SWI 0X11
.end
```

I. Output Screen Shots

(One Screenshot including Register Window,Memory Window and Code Window)



Microprocessor and Computer Architecture

UE22CS251B

4th Semester, Academic Year 2023-24

Date:

Name: Amogh P	SRN:PES2UG22CS062	Section : B

LAB #____2_____ Program Number: ____2___

Title of the Program

Write a program to Check if a given set of numbers are even or odd . Then store even and odd numbers in two different memory locations

I. ARM Assembly Code

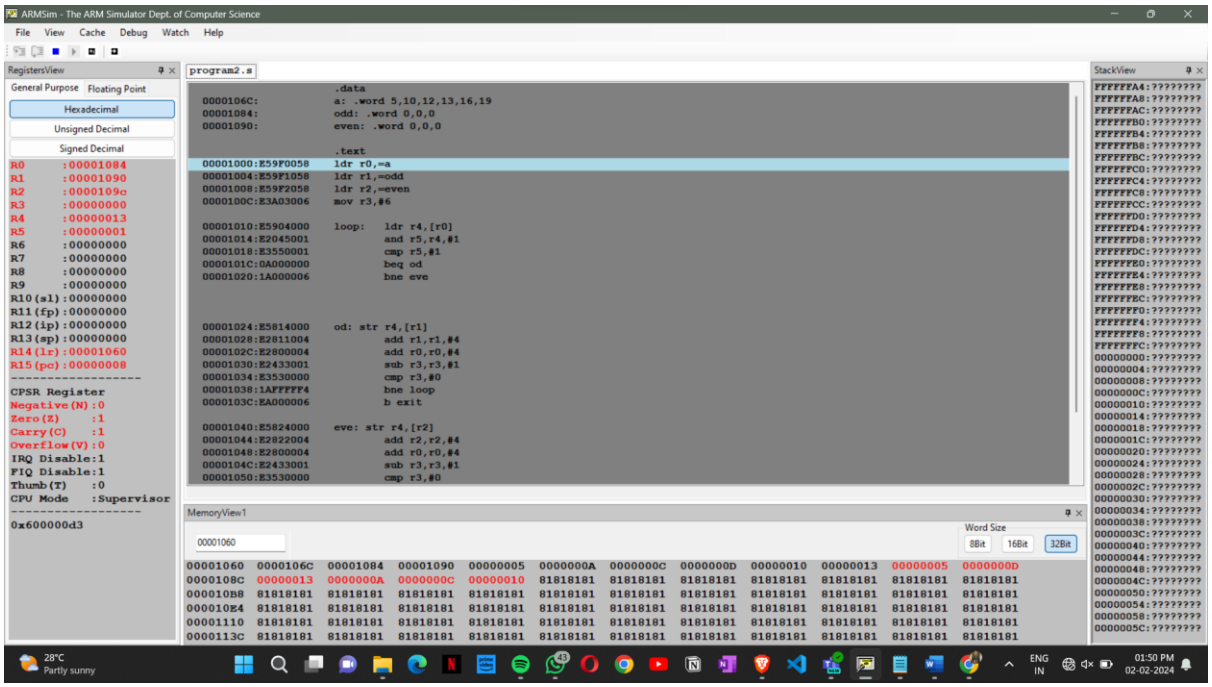
CODE:

```
.data
a: .word 5,10,12,13,16,19
odd: .word 0,0,0
even: .word 0,0,0
.text
ldr r0,=a
ldr r1,=odd
ldr r2,=even
mov r3,#6

loop:   ldr r4,[r0]
        and r5,r4,#1
```

```
        cmp r5,#1
        beq od
        bne eve

od: str r4,[r1]
        add r1,r1,#4
        add r0,r0,#4
        sub r3,r3,#1
        cmp r3,#0
        bne loop
        b exit

eve: str r4,[r2]
        add r2,r2,#4
        add r0,r0,#4
        sub r3,r3,#1
        cmp r3,#0
        bne loop
        b exit

exit:
        SWI 0X11
        .end
```

II. Output Screen Shots

(One Screenshot including Register Window,Memory Window and Code Window)



Microprocessor and Computer Architecture

UE22CS251B

4th Semester, Academic Year 2023-24

Date:

Name: Amogh P	SRN: PES2UG22CS062	Section :B

LAB #____2_____ Program Number: ____3___

Title of the Program

Write a program to Classify the given set of numbers as positive, negative or zero and also store them in different memory locations.

I. ARM Assembly Code

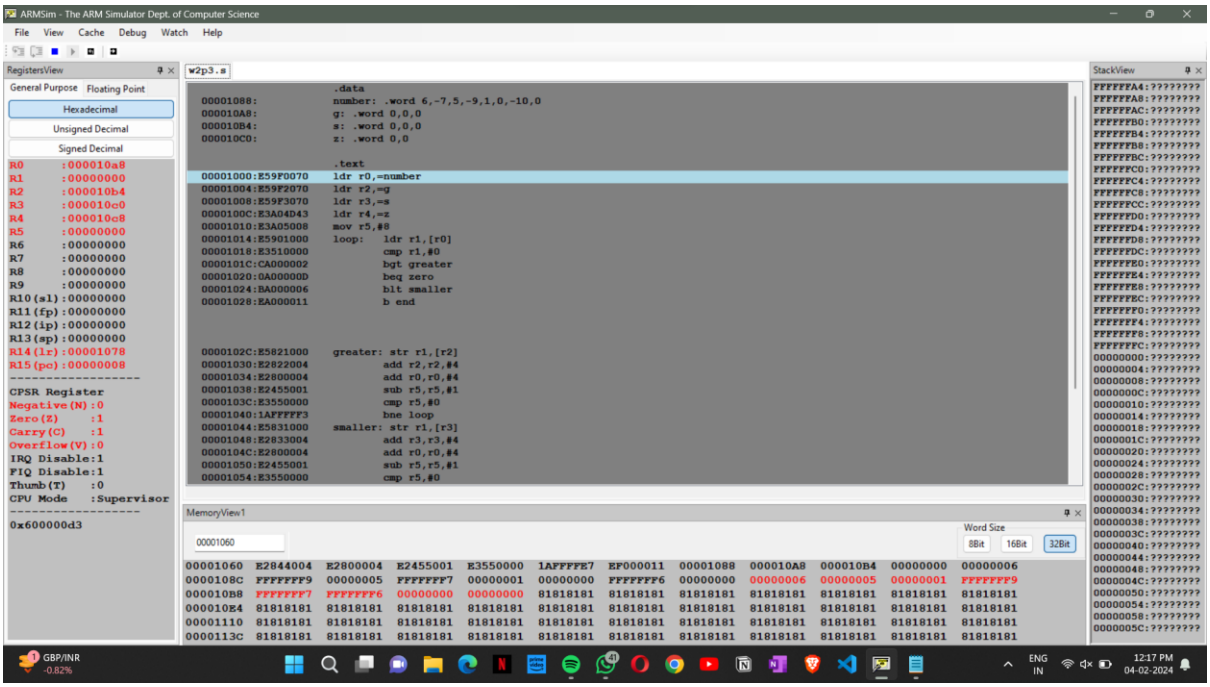
CODE:

```
.data
number: .word 6,-7,5,-9,1,0,-10,0
g: .word 0,0,0
s: .word 0,0,0
z: .word 0,0


.text
ldr r0,=number
ldr r2,=g
ldr r3,=s
ldr r4,=z
mov r5,#8
loop:   ldr r1,[r0]
```

```
        cmp r1,#0
        bgt greater
        beq zero
        blt smaller
        b end

greater: str r1,[r2]
        add r2,r2,#4
        add r0,r0,#4
        sub r5,r5,#1
        cmp r5,#0
        bne loop
smaller: str r1,[r3]
        add r3,r3,#4
        add r0,r0,#4
        sub r5,r5,#1
        cmp r5,#0
        bne loop
zero: str r1,[r4]
        add r4,r4,#4
        add r0,r0,#4
        sub r5,r5,#1
        cmp r5,#0
        bne loop
end: SWI 0X11
        .end
```

II. Output Screen Shots

(One Screenshot including Register Window,Memory Window and Code Window)



Microprocessor and Computer Architecture

UE22CS251B

4th Semester, Academic Year 2023-24

Date:

Name: Amogh P	SRN:PES2UG22CS062	Section :B

LAB #____2_____ Program Number: ____4___

Title of the Program

Write a program to find the largest number from a given set of numbers.(Unsorted Array)

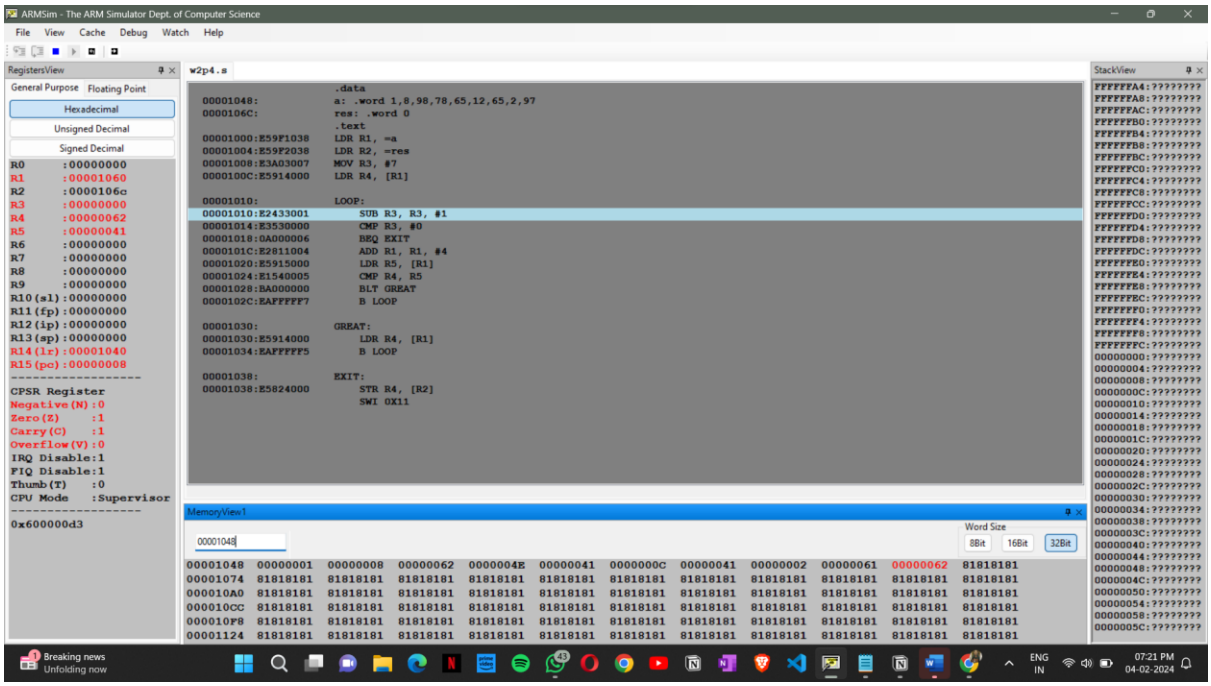
I. ARM Assembly Code

CODE:

```
.data
a: .word 1,8,98,78,65,12,65,2,97
res: .word 0
.text
LDR R1, =a
LDR R2, =res
MOV R3, #7
LDR R4, [R1]

LOOP:
   SUB R3, R3, #1
   CMP R3, #0
   BEQ EXIT
   ADD R1, R1, #4
   LDR R5, [R1]
   CMP R4, R5
   BLT GREAT
   B LOOP

GREAT:
   LDR R4, [R1]
   B LOOP

EXIT:
   STR R4, [R2]
   SWI 0X11
```

II. Output Screen Shots

(One Screenshot including Register Window,Memory Window and Code Window)


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 :00001060
R2 :0000106c
R3 :00000000
R4 :00000062
R5 :00000041
R6 :00000000
R7 :00000000
R8 :00000000
R9 :00000000
R10(sl) :00000000
R11(fp) :00000000
R12(ip) :00000000
R13(sp) :00000000
R14(lr) :00001040
R15(pc) :00000008
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 :Supervisor
0x600000d3
w2p4.s
.data
00001048: a: .word 1,8,98,78,65,12,65,2,97
0000106C: res: .word 0
.text
00001000:E59F1038 LDR R1, =a
00001004:E59F2038 LDR R2, =res
00001008:E3A03007 MOV R3, #7
0000100C:E5914000 LDR R4, [R1]
00001010: LOOP:
00001010:E2433001 SUB R3, R3, #1
00001014:E3530000 CMP R3, #0
00001018:0A000006 BEQ EXIT
0000101C:E2811004 ADD R1, R1, #4
00001020:E5915000 LDR R5, [R1]
00001024:E1540005 CMP R4, R5
00001028:BA000000 BLT GREAT
0000102C:EAFFFFF7 B LOOP
00001030: GREAT:
00001030:E5914000 LDR R4, [R1]
00001034:EAFFFFF5 B LOOP
00001038: EXIT:
00001038:E5824000 STR R4, [R2]
SWI 0X11
MemoryView1
00001048
Word Size
8Bit 16Bit 32Bit
00001048 00000001 00000008 00000062 0000004E 00000041 0000000C 00000041 00000002 00000061 00000062 81818181
00001074 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010A0 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010CC 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010F8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001124 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
StackView
FFFFFFA4:????????
FFFFFFA8:????????
FFFFFFAC:????????
FFFFFFB0:????????
FFFFFFB4:????????
FFFFFFB8:????????
FFFFFFBC:????????
FFFFFFC0:????????
FFFFFFC4:????????
FFFFFFC8:????????
FFFFFFCC:????????
FFFFFFD0:????????
FFFFFFD4:????????
FFFFFFD8:????????
FFFFFFDC:????????
FFFFFFE0:????????
FFFFFFE4:????????
FFFFFFE8:????????
FFFFFFEC:????????
FFFFFFF0:????????
FFFFFFF4:????????
FFFFFFF8:????????
FFFFFFFC:????????
00000000:????????
00000004:????????
00000008:????????
0000000C:????????
00000010:????????
00000014:????????
00000018:????????
0000001C:????????
00000020:????????
00000024:????????
00000028:????????
0000002C:????????
00000030:????????
00000034:????????
00000038:????????
0000003C:????????
00000040:????????
00000044:????????
00000048:????????
0000004C:????????
00000050:????????
00000054:????????
00000058:????????
0000005C:????????
Breaking news
Unfolding now
ENG
IN
07:21 PM
04-02-2024

Microprocessor and Computer Architecture

UE22CS251B

4th Semester, Academic Year 2023-24

Date:

Name: Amogh P	SRN:PES2UG22CS062	Section : B

LAB #____2_____ Assignment Question 1

Title of the Program

Write a program to add array of ten 8-bit numbers taking data from memory location stored as byte data

(Hint: use .byte to store the data instead of .word

Use LDRB instruction for data transfer, Increment the index register by 1 to go to next element of array)

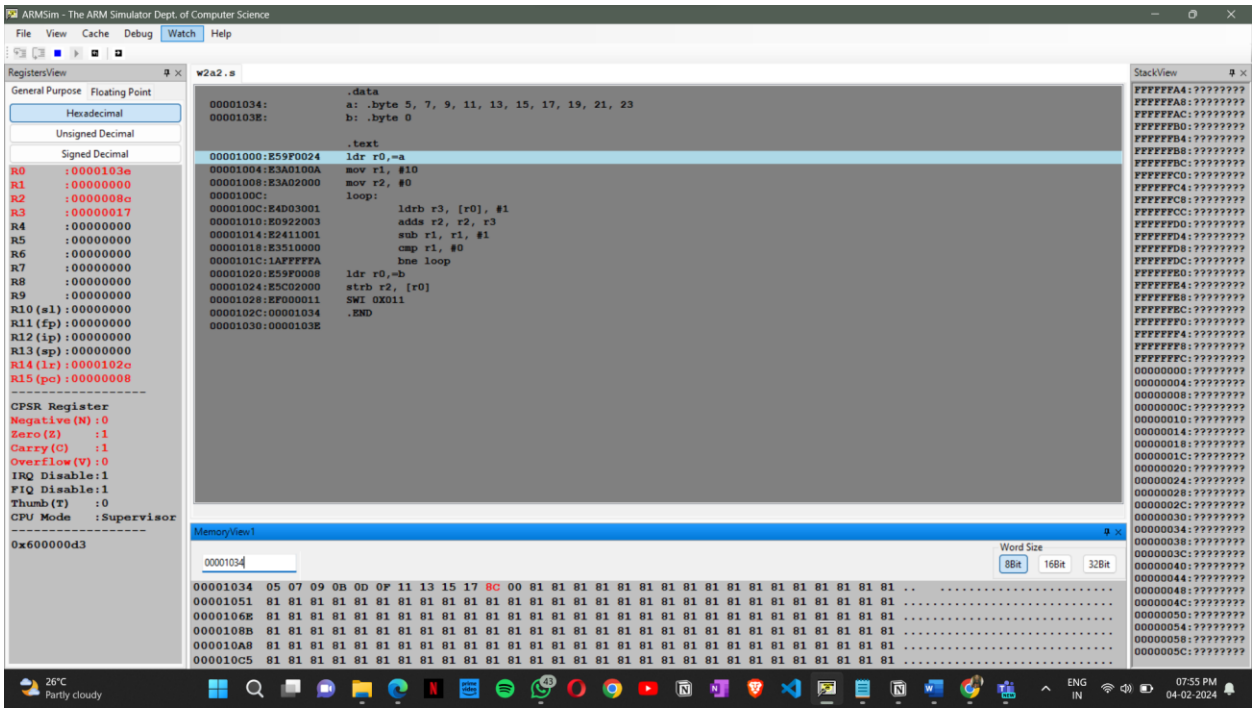
I. ARM Assembly Code

```
CODE:

.data

a: .byte 5, 7, 9, 11, 13, 15, 17, 19, 21, 23

b: .byte 0


.text

ldr r0,=a

mov r1, #10

mov r2, #0

loop:
```

```
        ldrb r3, [r0], #1
        adds r2, r2, r3
        sub r1, r1, #1
        cmp r1, #0
        bne loop
    ldr r0,=b
    strb r2, [r0]
    SWI 0X011
    .END
```

II. Output Screen Shots

(One Screenshot including Register Window,Memory Window and Code Window)



Microprocessor and Computer Architecture

UE22CS251B

4th Semester, Academic Year 2023-24

Date:

Name: Amogh P	SRN:PES2UG22CS062	Section: B

LAB #____2_____ Assignment Question 2

Title of the Program

Generate Fibonacci Series and store them in an array / memory location.

I. ARM Assembly Code

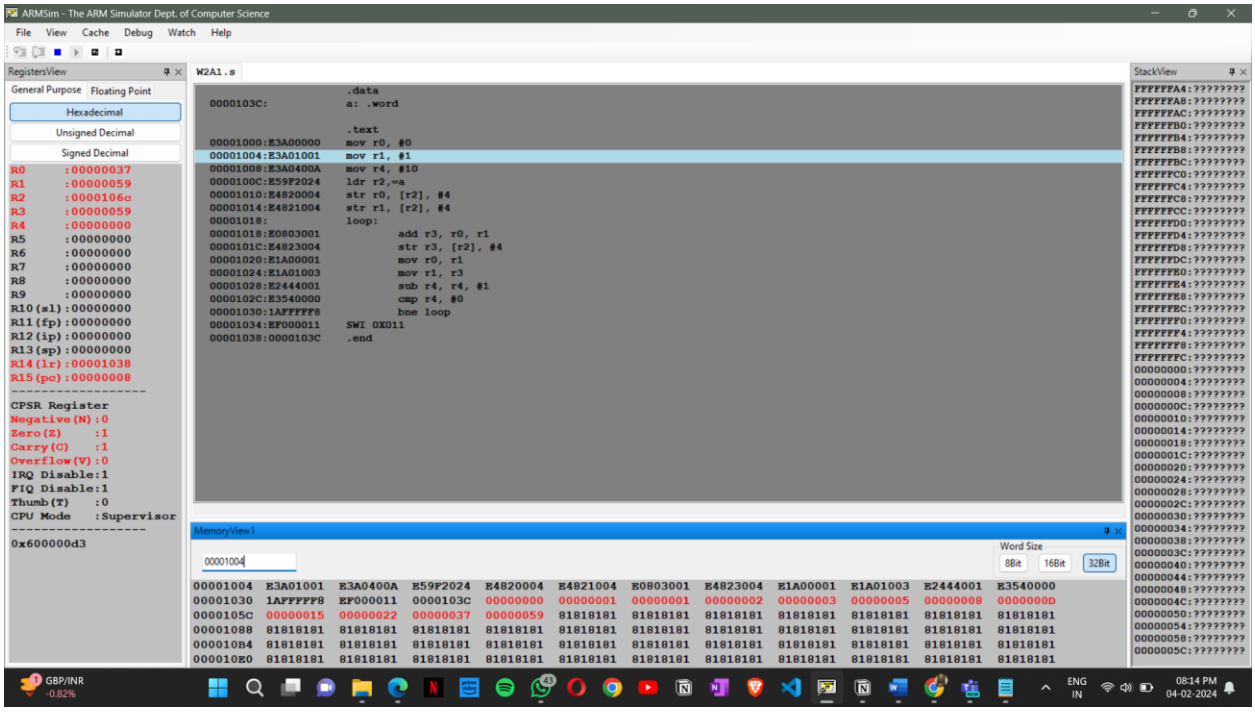
CODE:

```
.data
a: .word

.text
mov r0, #0
mov r1, #1
mov r4, #10
ldr r2,=a
str r0, [r2], #4
str r1, [r2], #4
loop:
        add r3, r0, r1
        str r3, [r2], #4
        mov r0, r1
        mov r1, r3
        sub r4, r4, #1
```

```
        cmp r4, #0
        bne loop
    SWI 0X011
    .end
```

II. Output Screen Shots

(One Screenshot including Register Window,Memory Window and Code Window)

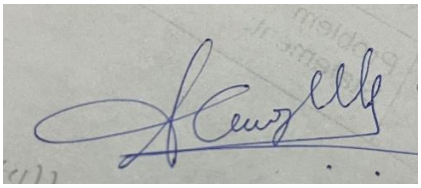


Disclaimer:

- 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:



Name : Amogh P

SRN: PES2UG22CS062
Section: B
Date:04/02/2024