

Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

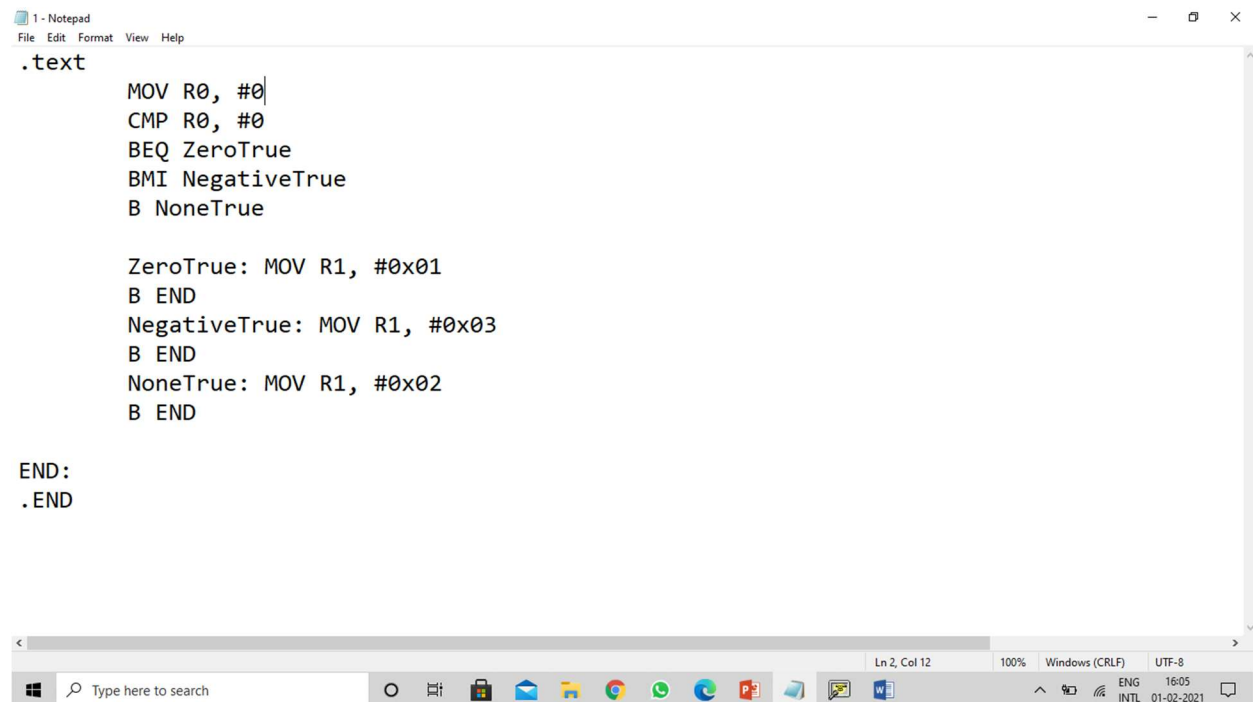
Date:28-01-21

| | | |
|---------------------|-------------------|--------------|
| Name: Achyut Jagini | SRN:PES2UG19CS013 | Section A |
|---------------------|-------------------|--------------|

Week# 2 Program Number: _____

Title of the Program

5.

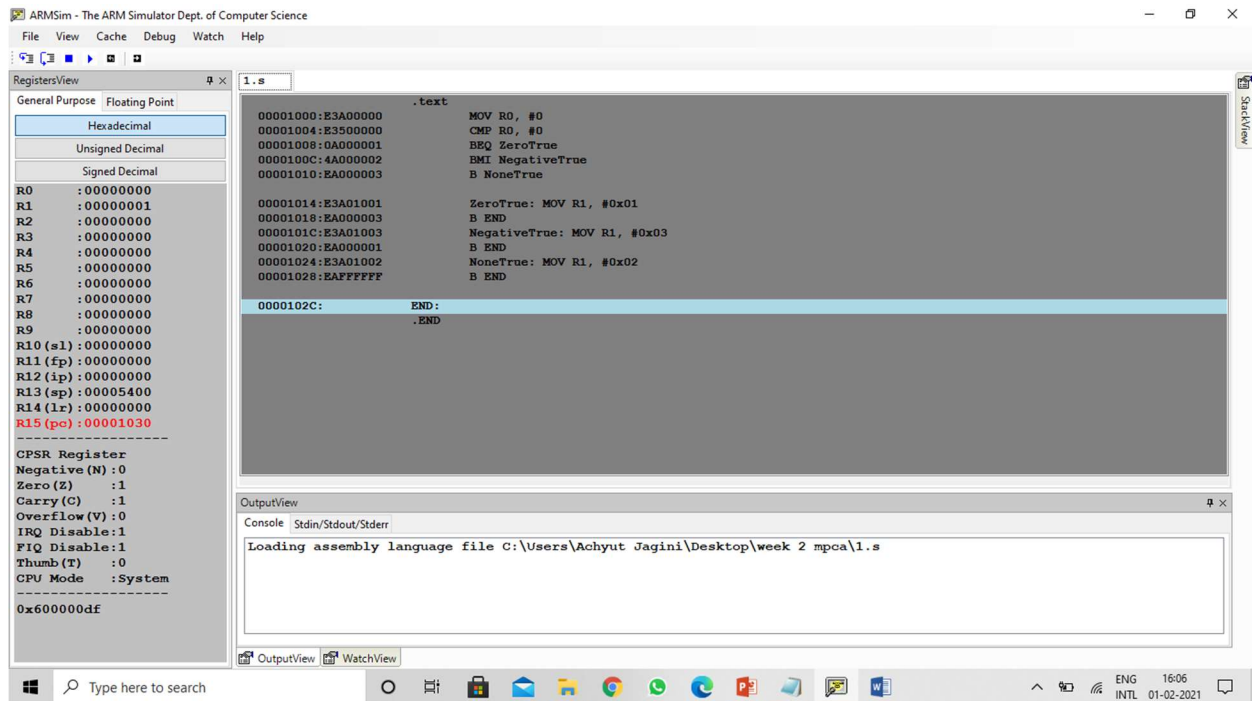


```
1 - Notepad
File Edit Format View Help
.text
    MOV R0, #0
    CMP R0, #0
    BEQ ZeroTrue
    BMI NegativeTrue
    B NoneTrue

ZeroTrue: MOV R1, #0x01
B END
NegativeTrue: MOV R1, #0x03
B END
NoneTrue: MOV R1, #0x02
B END

END:
.END
```

Ln 2, Col 12 100% Windows (CRLF) UTF-8
Type here to search 16:05 01-02-2021



6. Write an ALP to compare the value of R0 and R1, add if R0 = R1, else subtract (Program shown in class)

2 - Notepad

File Edit Format View Help

```
.text

MOV R0, #90
MOV R1, #45

CMP R0, R1

BEQ L1

SUB R2, R1, R0
SWI 0x1011

L1: ADD R2, R1, R0
SWI 0x1011

.END
```

Ln 9, Col 18 100% Windows (CRLF) UTF-8

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 : 0000005a
R1 : 0000002d
R2 : ffffffff
R3 : 00000000
R4 : 00000000
R5 : 00000000
R6 : 00000000
R7 : 00000000
R8 : 00000000
R9 : 00000000
R10 (s1) : 00000000
R11 (fp) : 00000000
R12 (ip) : 00000000
R13 (sp) : 00000000
R14 (lr) : 00001018
R15 (pc) : 00000008

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 : Supervisor

0x200000d3

2.s

```
.text
00001000:E3A0005A MOV R0, #90
00001004:E3A0102D MOV R1, #45
00001008:E1500001 CMP R0, R1
0000100C:0A000001 BEQ L1
00001010:E0412000 SUB R2, R1, R0
00001014:EF001011 SWI 0x1011
00001018:E0812000 L1: ADD R2, R1, R0
0000101C:EF001011 SWI 0x1011
.END
```

OutputView

Console Stdin/Stdout/Stderr

Loading assembly language file C:\Users\Achyut Jagini\Desktop\week 2 mpca\2.s
PC out of valid memory range, address:00000008

ENG INTL 16:12 01-02-2021

7. 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. (Program shown in class)

```

3 - Notepad
File Edit Format View Help
.text
    MOV R0, #7
    MOV R1, R0 @ n

    RE: SUB R0, R0, #1 @ n-1
        MUL R2, R1, R0 @ R2 = n * n-1
        MOV R1, R2
        CMP R0, #1
        BNE RE
        SWI 0X1011

.END

```

ARMSim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView

General Purpose Floating Point

| Register | Hexadecimal | Unsigned Decimal | Signed Decimal |
|----------|-------------|------------------|----------------|
| R0 | :00000003 | | |
| R1 | :000009d8 | | |
| R2 | :000009d8 | | |
| R3 | :00000000 | | |
| R4 | :00000000 | | |
| R5 | :00000000 | | |
| R6 | :00000000 | | |
| R7 | :00000000 | | |
| R8 | :00000000 | | |
| R9 | :00000000 | | |
| R10 (s1) | :00000000 | | |
| R11 (fp) | :00000000 | | |
| R12 (ip) | :00000000 | | |
| R13 (sp) | :00005400 | | |
| R14 (lr) | :00000000 | | |
| R15 (pc) | :00001014 | | |

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

0x200000df

3.S

```

.text
00001000:E3A00007    MOV R0, #7
00001004:E1A01000    MOV R1, R0 @ n

00001008:E2400001    RE: SUB R0, R0, #1 @ n-1
0000100C:E0020091    MUL R2, R1, R0 @ R2 = n * n-1
00001010:E1A01002    MOV R1, R2
00001014:E3500001    CMP R0, #1
00001018:1AFFFFFA    BNE RE
0000101C:EF001011    SWI 0X1011

.END

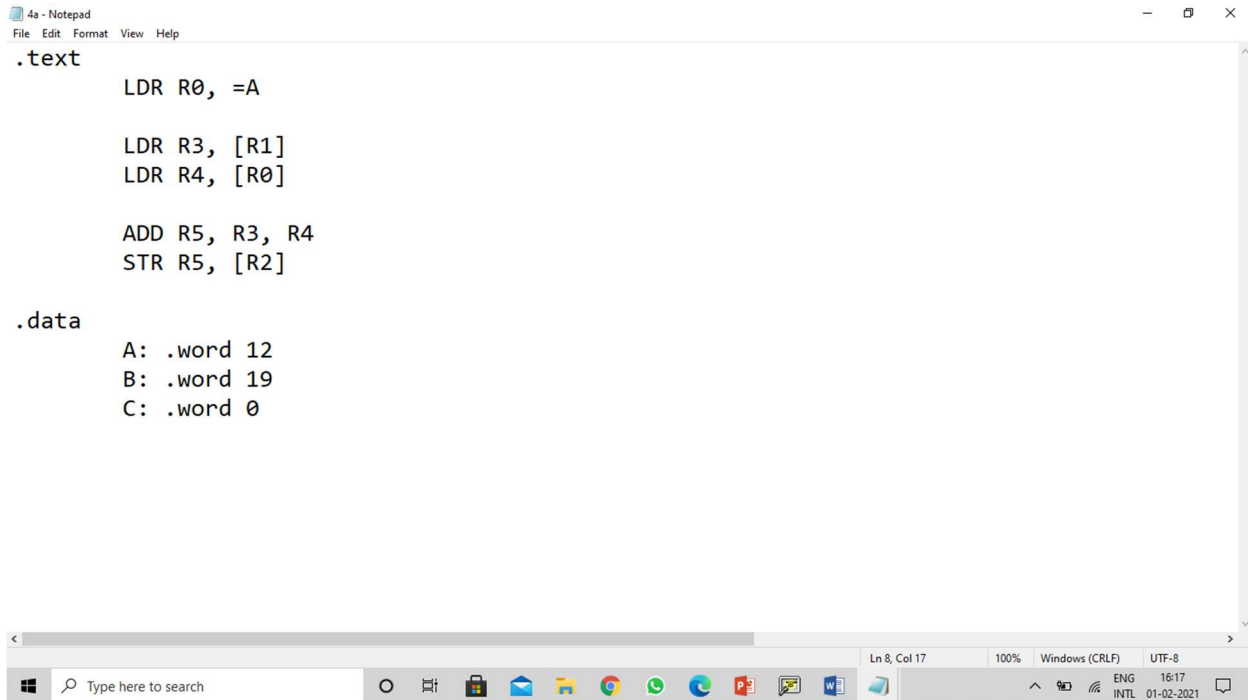
```

OutputView

Console Stdin/Stdout/Stderr

Loading assembly language file C:\Users\Achyut Jagini\Desktop\Week 2\3\3.S

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



```
.text

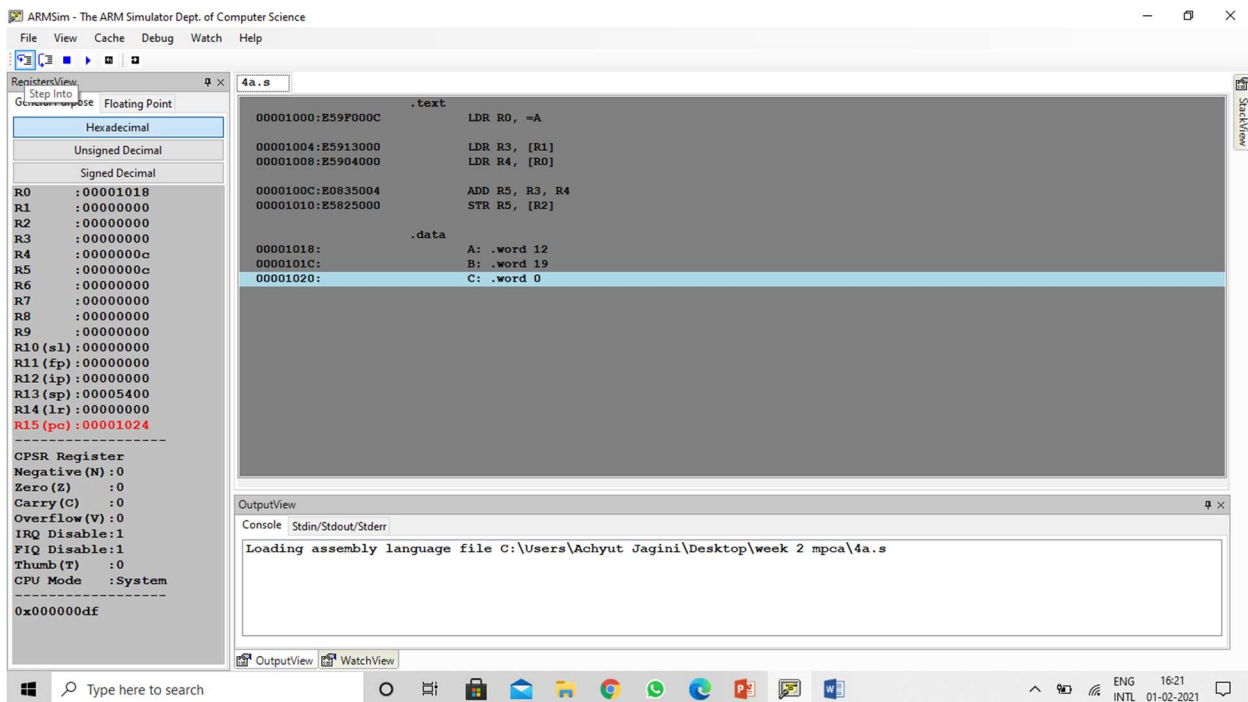
LDR R0, =A

LDR R3, [R1]
LDR R4, [R0]

ADD R5, R3, R4
STR R5, [R2]

.data

A: .word 12
B: .word 19
C: .word 0
```



ARMSim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView: Step Into Floating Point

Hexadecimal
Unsigned Decimal
Signed Decimal

R0 : 00001018
R1 : 00000000
R2 : 00000000
R3 : 00000000
R4 : 0000000c
R5 : 0000000c
R6 : 00000000
R7 : 00000000
R8 : 00000000
R9 : 00000000
R10 (s1) : 00000000
R11 (fp) : 00000000
R12 (ip) : 00000000
R13 (sp) : 00005400
R14 (lr) : 00000000
R15 (pc) : 00001024

CPSR Register
Negative (N) : 0
Zero (Z) : 0
Carry (C) : 0
Overflow (V) : 0
IRQ Disable : 1
FIQ Disable : 1
Thumb (T) : 0
CPU Mode : System
0x000000df

4a.s

```
.text
00001000:E59F000C      LDR R0, =A
00001004:E5913000      LDR R3, [R1]
00001008:E5904000      LDR R4, [R0]
0000100C:E0835004      ADD R5, R3, R4
00001010:E5825000      STR R5, [R2]
.data
00001018:                A: .word 12
0000101C:                B: .word 19
00001020:                C: .word 0
```

OutputView
Console Stdin/Stdout/Stderr
Loading assembly language file C:\Users\Achyut Jagini\Desktop\week 2 mpc\4a.s

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

```
4b - Notepad
File Edit Format View Help

.text

    LDR R0, =A
    LDR R1, =B
    LDR R2, =C

    LDRH R3, [R1]
    LDRH R4, [R0]

    ADD R5, R3, R4
    STR R5, [R2]

.data
A: .hword 12
B: .hword 19
C: .hword 0
```

ARMSim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView

Step Into Step Out Floating Point

Hexadecimal
Unsigned Decimal
Signed Decimal

R0 : 00001028
R1 : 0000102a
R2 : 0000102c
R3 : 00000013
R4 : 0000000c
R5 : 0000001f
R6 : 00000000
R7 : 00000000
R8 : 00000000
R9 : 00000000
R10 (s1) : 00000000
R11 (fp) : 00000000
R12 (ip) : 00000000
R13 (sp) : 00005400
R14 (lr) : 00000000
R15 (pc) : 00001020

CPSR Register
Negative (N) : 0
Zero (Z) : 0
Carry (C) : 0
Overflow (V) : 0
IRQ Disable : 1
FIQ Disable : 1
Thumb (T) : 0
CPU Mode : System
0x000000df

4b.s

```
.text
00001000:E59F0014    LDR R0, =A
00001004:E59F1014    LDR R1, =B
00001008:E59F2014    LDR R2, =C

0000100C:E01130B0    LDRH R3, [R1]
00001010:E01040B0    LDRH R4, [R0]

00001014:E0835004    ADD R5, R3, R4
00001018:E5825000    STR R5, [R2]

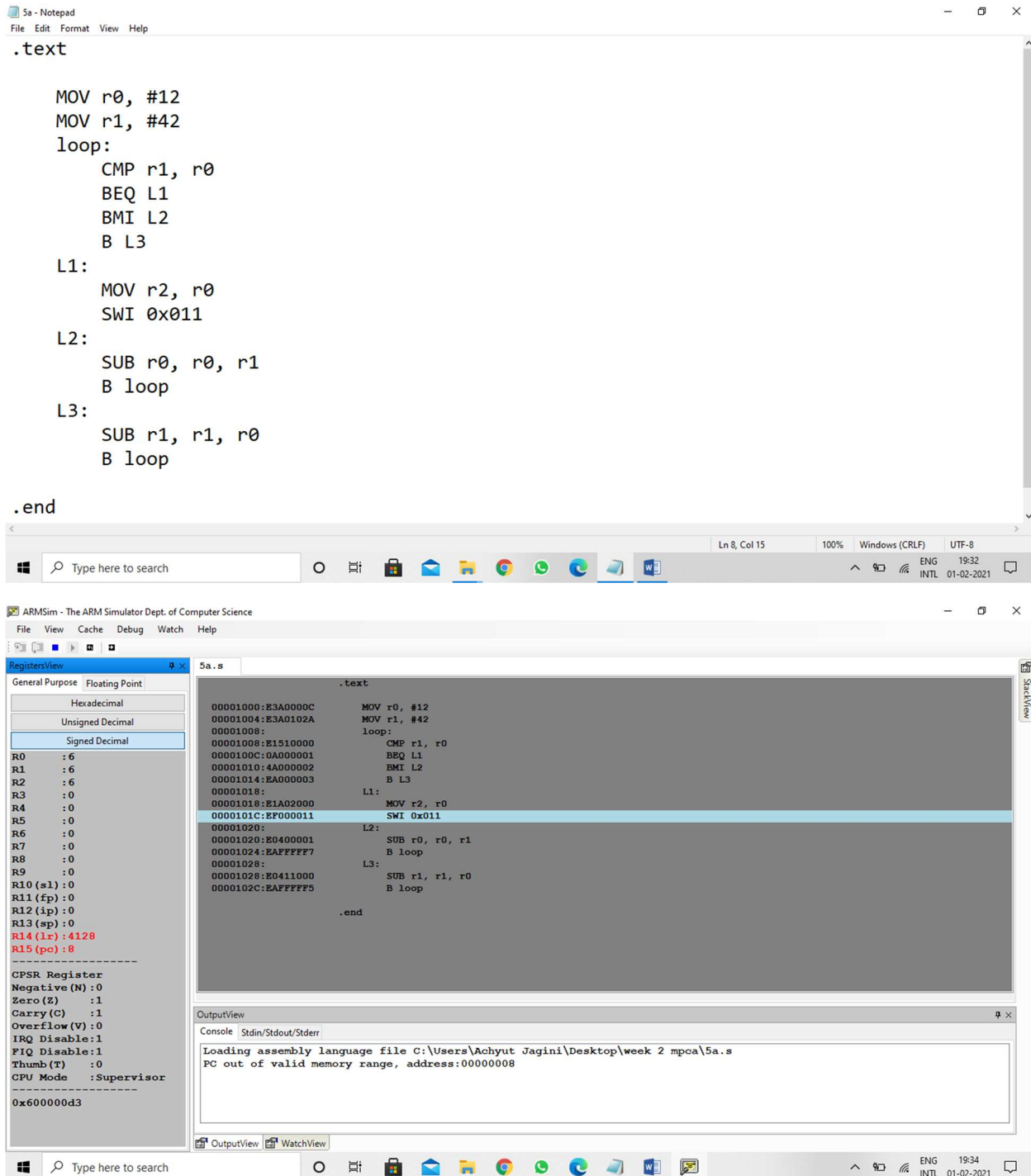
.data
00001028:          A: .hword 12
0000102A:          B: .hword 19
0000102C:          C: .hword 0
```

OutputView

Console Stdin/Stdout/Stderr

Loading assembly language file C:\Users\Achyut Jagini\Desktop\week 2 mpc\4b.s

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



The image shows two windows. The top window is Notepad, displaying assembly code for finding the GCD of two numbers in registers. The code uses the Euclidean algorithm with subtraction. The bottom window is ARMSim, showing the same assembly code loaded into memory, the current register values, and the output console.

```
.text

MOV r0, #12
MOV r1, #42
loop:
    CMP r1, r0
    BEQ L1
    BMI L2
    B L3
L1:
    MOV r2, r0
    SWI 0x011
L2:
    SUB r0, r0, r1
    B loop
L3:
    SUB r1, r1, r0
    B loop

.end
```

ARMSim - The ARM Simulator Dept. of Computer Science

Registers View

| General Purpose | Floating Point |
|------------------|----------------|
| Hexadecimal | |
| Unsigned Decimal | |
| Signed Decimal | |

R0 : 6
R1 : 6
R2 : 6
R3 : 0
R4 : 0
R5 : 0
R6 : 0
R7 : 0
R8 : 0
R9 : 0
R10 (s1) : 0
R11 (fp) : 0
R12 (ip) : 0
R13 (sp) : 0
R14 (lr) : 4128
R15 (pc) : 8

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

5a.s

```
.text

00001000:E3A0000C    MOV r0, #12
00001004:E3A0102A    MOV r1, #42
00001008:                loop:
00001008:E1510000        CMP r1, r0
0000100C:0A000001        BEQ L1
00001010:4A000002        BMI L2
00001014:EAD00003        B L3
00001018:                L1:
00001018:E1A02000        MOV r2, r0
0000101C:EF000011        SWI 0x011
00001020:                L2:
00001020:E0400001        SUB r0, r0, r1
00001024:EAF00007        B loop
00001028:                L3:
00001028:E0410000        SUB r1, r1, r0
0000102C:EAF00005        B loop

.end
```

Output View

Console Stdin/Stdout/Stderr

Loading assembly language file C:\Users\Achyut Jagini\Desktop\week 2 mpc\5a.s
PC out of valid memory range, address:00000008

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

```
10a - Notepad
File Edit Format View Help

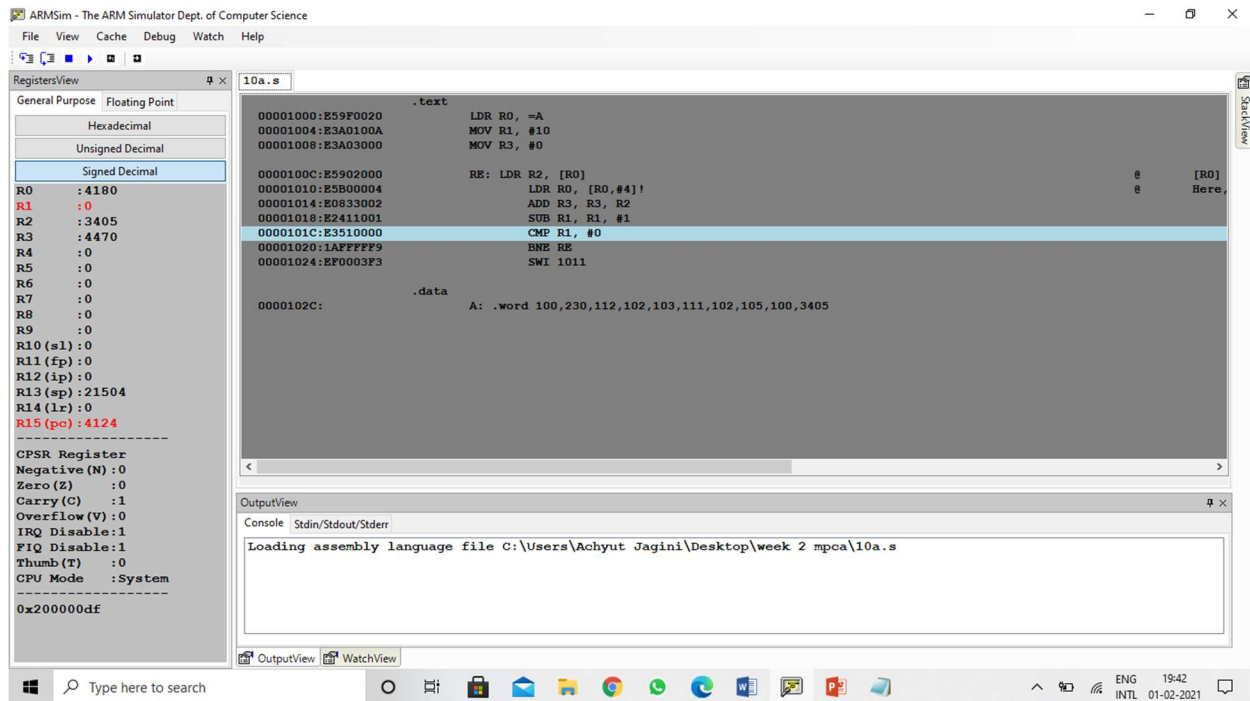
.text
    LDR R0, =A
    MOV R1, #10
    MOV R3, #0

    RE: LDR R2, [R0]
        LDR R0, [R0,#4]!
        ADD R3, R3, R2
        SUB R1, R1, #1
        CMP R1, #0
        BNE RE
        SWI 1011

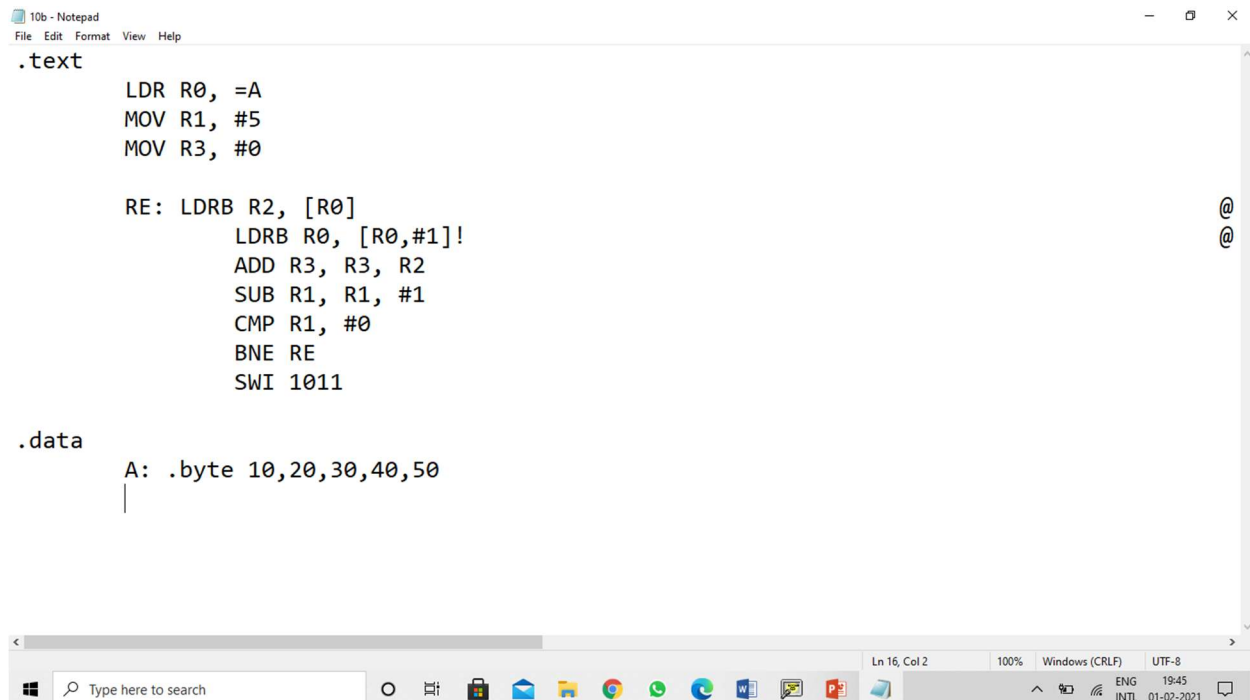
.data
    A: .word 100,230,112,102,103,111,102,105,100,3405

|

Ln 17, Col 2    100%    Windows (CRLF)    UTF-8
Type here to search  [Taskbar icons] [System tray: ENG, 19:40, 01-02-2021]
```

b) Add array of ten 16 bit numbers taking data from memory location (use .hword to store the data instead of .word)



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 : 0
R1 : 0
R2 : 0
R3 : 0
R4 : 0
R5 : 0
R6 : 0
R7 : 0
R8 : 0
R9 : 0
R10 (s1) : 0
R11 (fp) : 0
R12 (ip) : 0
R13 (sp) : 21504
R14 (lr) : 0
R15 (pc) : 4096

CPSR Register
Negative (N) : 0
Zero (Z) : 0
Carry (C) : 0
Overflow (V) : 0
IRQ Disable : 1
FIQ Disable : 1
Thumb (T) : 0
CPU Mode : System

0x000000df

10b.s

```
.text
00001000:E59F0020    LDR R0, =A
00001004:E3A01005    MOV R1, #5
00001008:E3A03000    MOV R3, #0

0000100C:E5D02000    RE: LDRB R2, [R0]
00001010:E5F00001    LDRB R0, [R0,#1]!
00001014:E0833002    ADD R3, R3, R2
00001018:E2411001    SUB R1, R1, #1
0000101C:E3510000    CMP R1, #0
00001020:1AFFFFFFF9    BNE RE
00001024:EF0003F3    SWI 1011

.data
0000102C:          A: .byte 10,20,30,40,50
```

OutputView

Console Stdin/Stdout/Stderr

Loading assembly language file C:\Users\Achyut Jagini\Desktop\week 2 mpc\10b.s

OutputView WatchView

Type here to search

ENG 19:46
INTL 01-02-2021

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:achyut

Name:Achyut Jagini

SRN:PES2UG19CS013

Section: A

Date:1-2-21