Microprocessor and Computer Architecture Laboratory

UE19CS256

4th Semester, Academic Year 2020-21

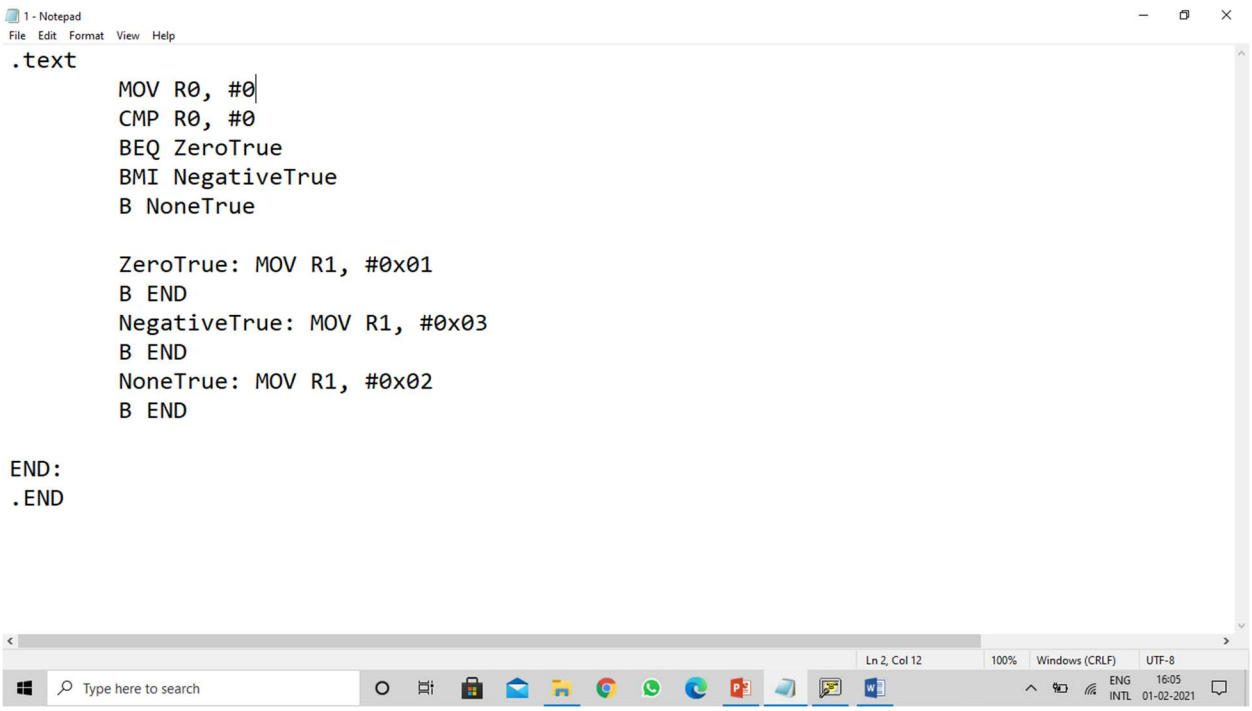
Date:28-01-21

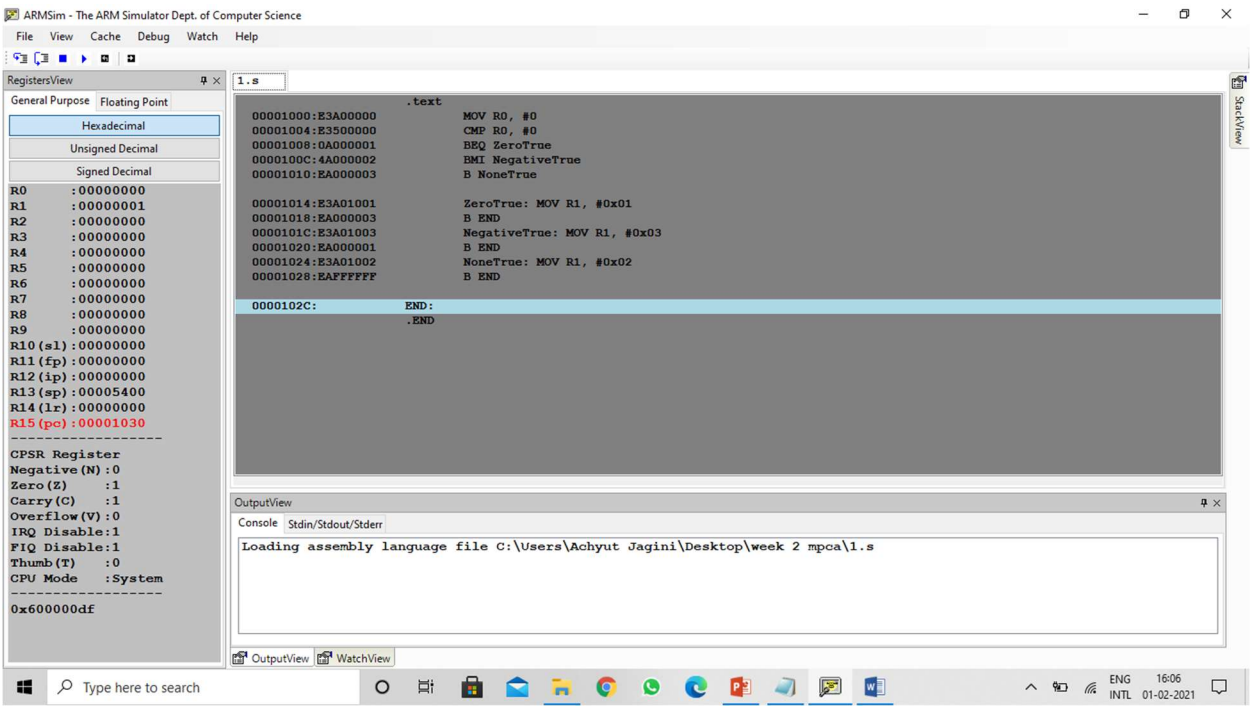
Name: Achyut Jagini	SRN:PES2UG19CS013	Section A

Week#____2_____ Program Number: _____

Title of the Program

5.





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

Type here to search

ENG INTL 16:11 01-02-2021

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      :0000005a
R1      :0000002d
R2      :fffffd3
R3      :00000000
R4      :00000000
R5      :00000000
R6      :00000000
R7      :00000000
R8      :00000000
R9      :00000000
R10(sl):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
```

StackView

OutputView

Console | Stdin/Stdout/Stderr

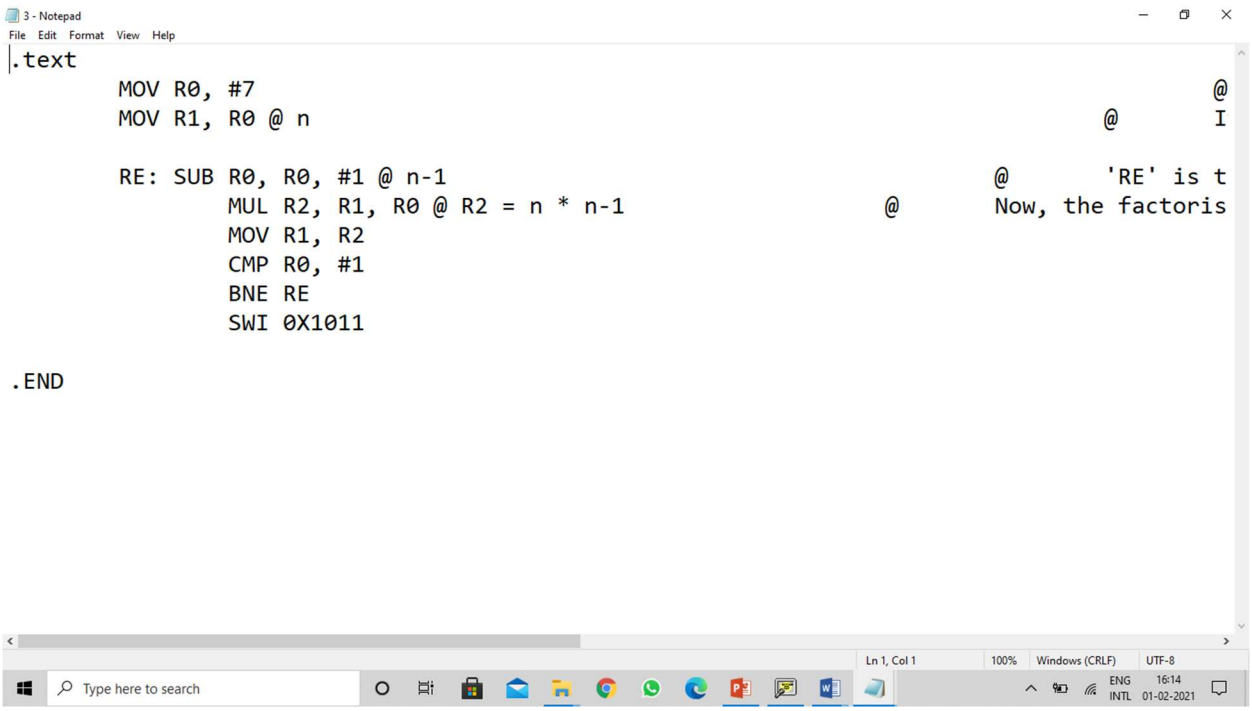
```
Loading assembly language file C:\Users\Achyut Jagini\Desktop\week 2 mpca\2.s
PC out of valid memory range, address:00000008
```

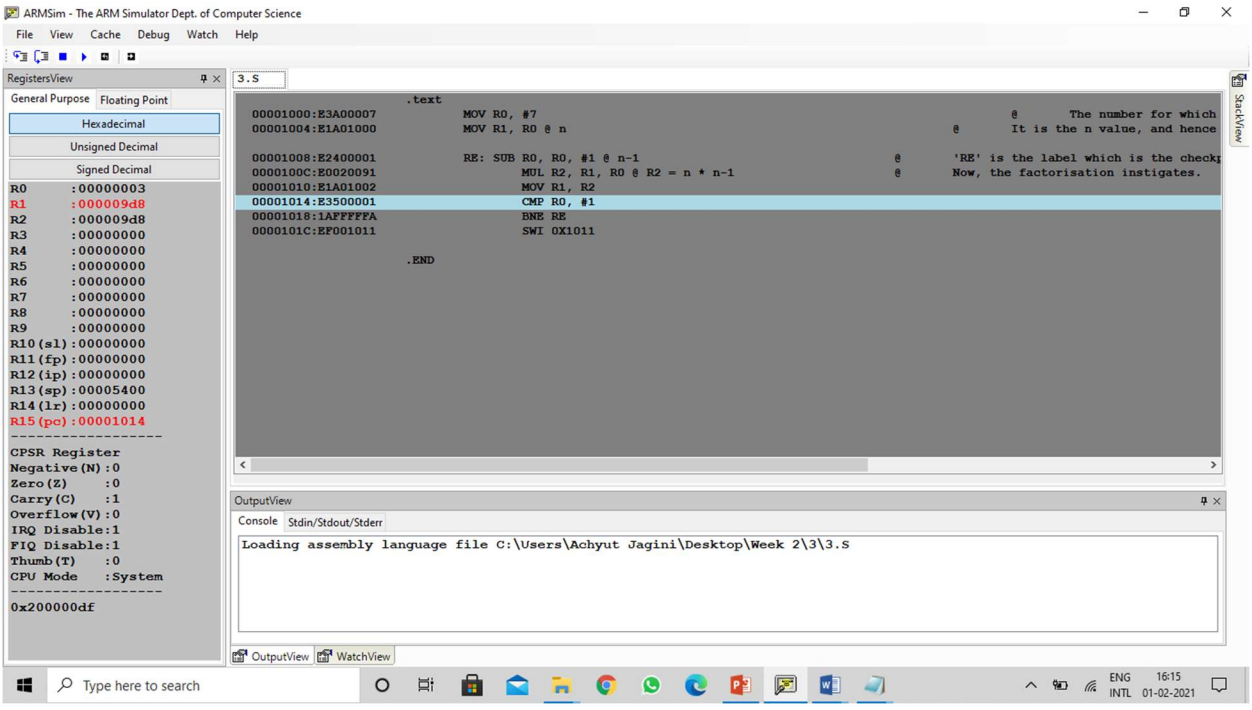
OutputView | WatchView

Type here to search

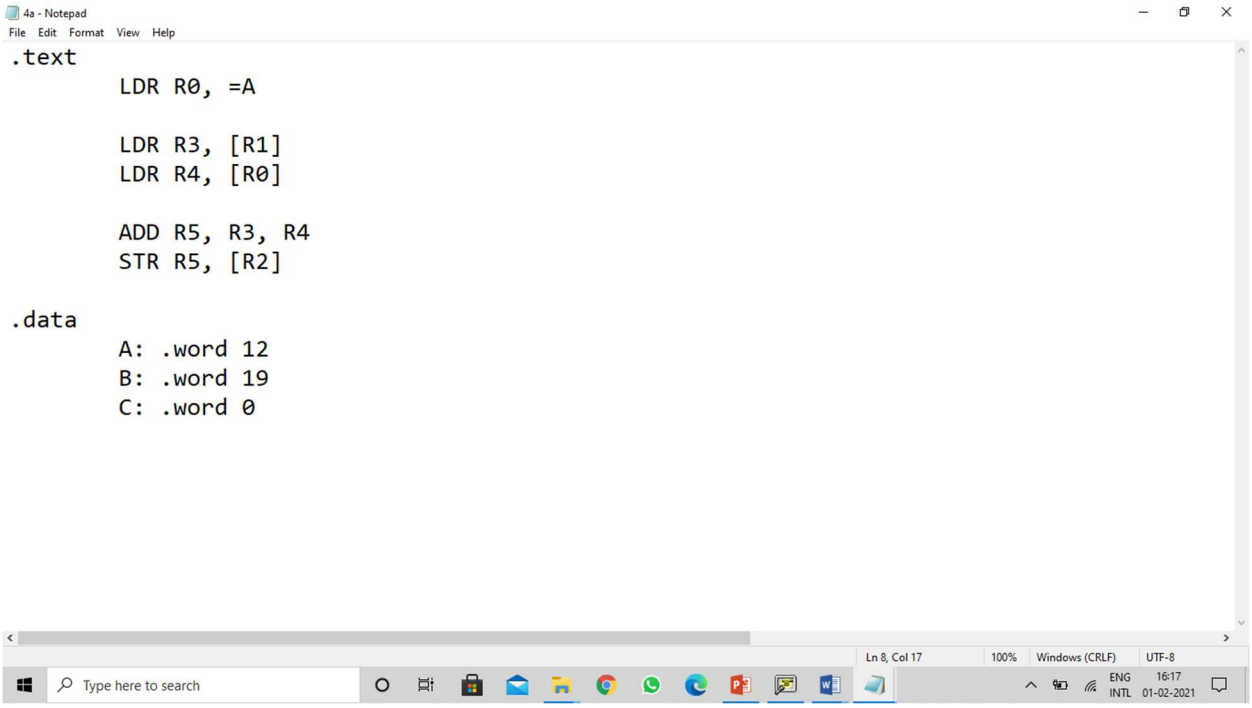
ENG INTL 16:12 01-02-2021

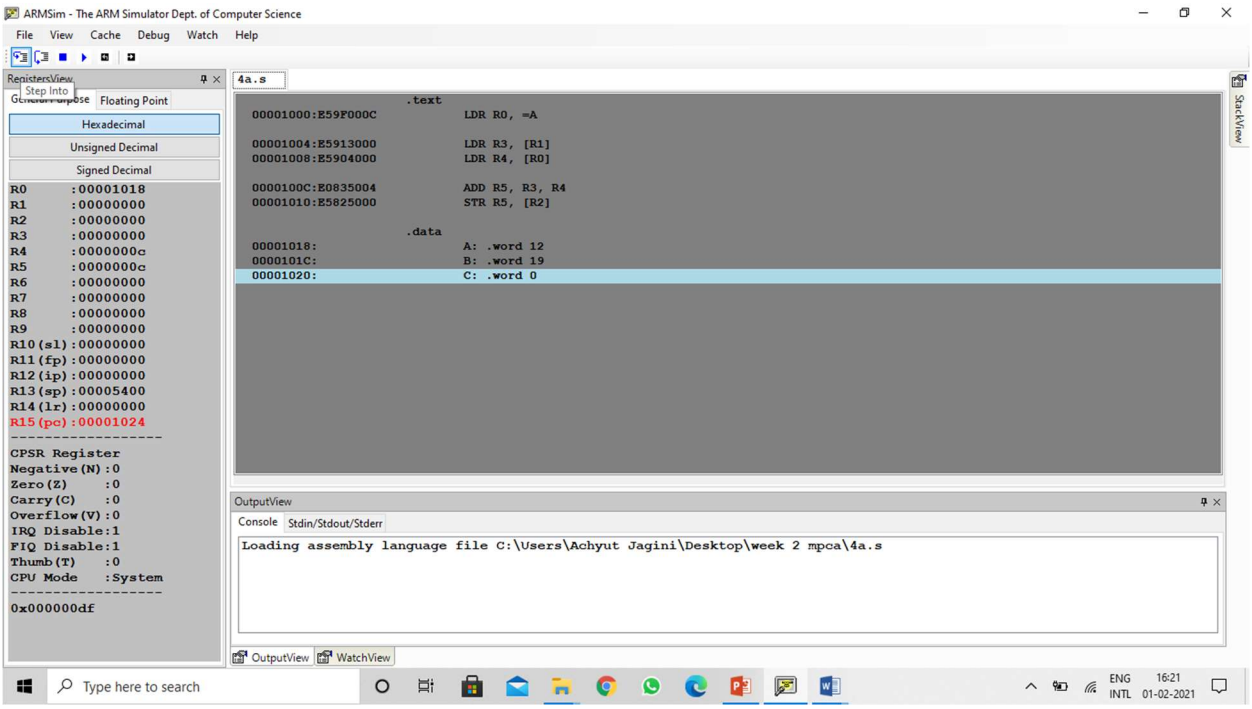
7. Write an ALP to find the factorial of a number stored in R0. Store the value in R1 (without using LDR and STR instructions).Use only registers. (Program shown in class)



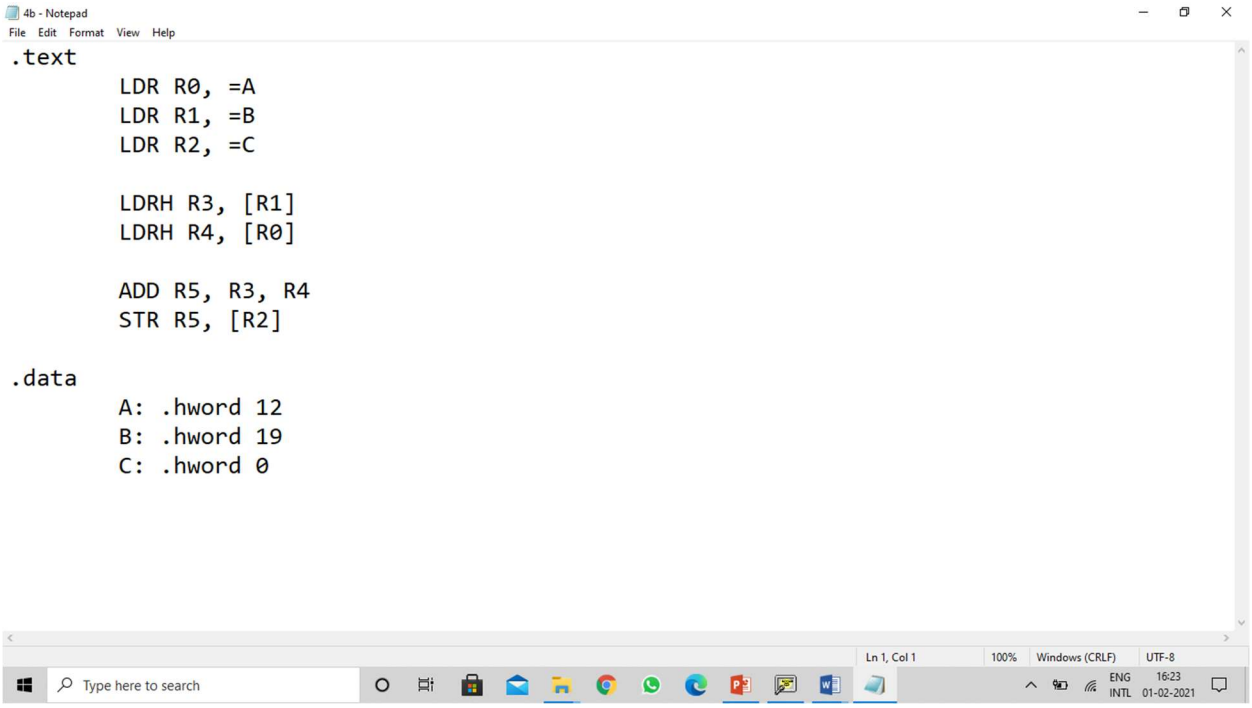


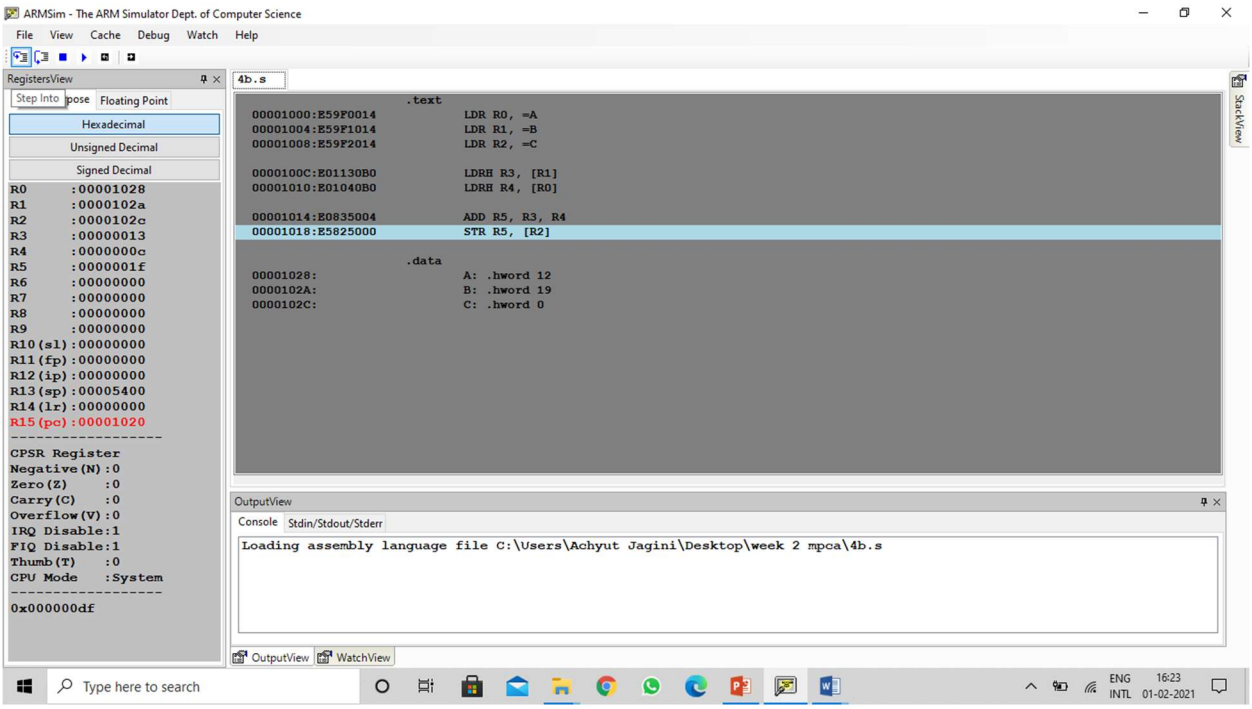
8. a) Write an ALP to add two 32 bit numbers loaded from memory and store the result in memory.





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





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.

5a - Notepad

File Edit Format View Help

```
.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
```

Ln 8, Col 15 100% Windows (CRLF) UTF-8

Type here to search ENG INTL 19:32 01-02-2021

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      :6
R1      :6
R2      :6
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):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:EA000003           B L3
00001018:               L1:
00001018:E1A02000           MOV r2, r0
0000101C:EF000011           SWI 0x011
00001020:               L2:
00001020:E0400001           SUB r0, r0, r1
00001024:EAFFFFF7           B loop
00001028:               L3:
00001028:E0411000           SUB r1, r1, r0
0000102C:EAFFFFF5           B loop

                    .end
```

OutputView

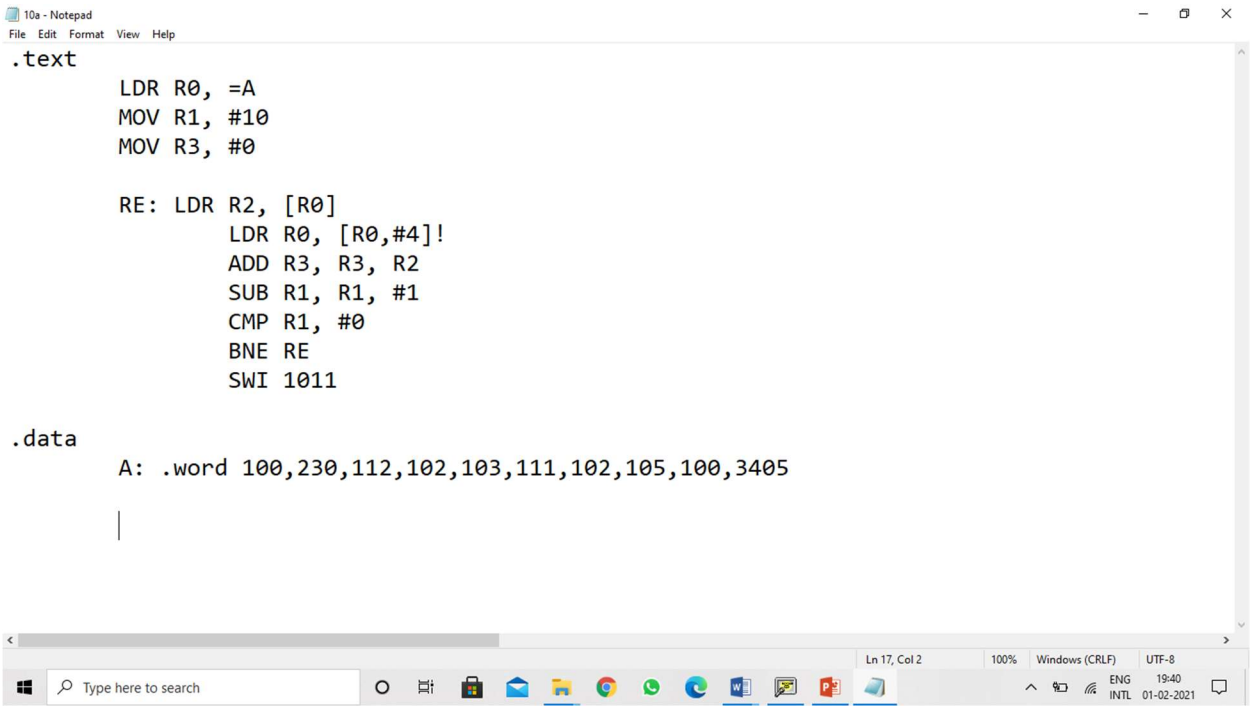
Console | Stdin/Stdout/Stderr

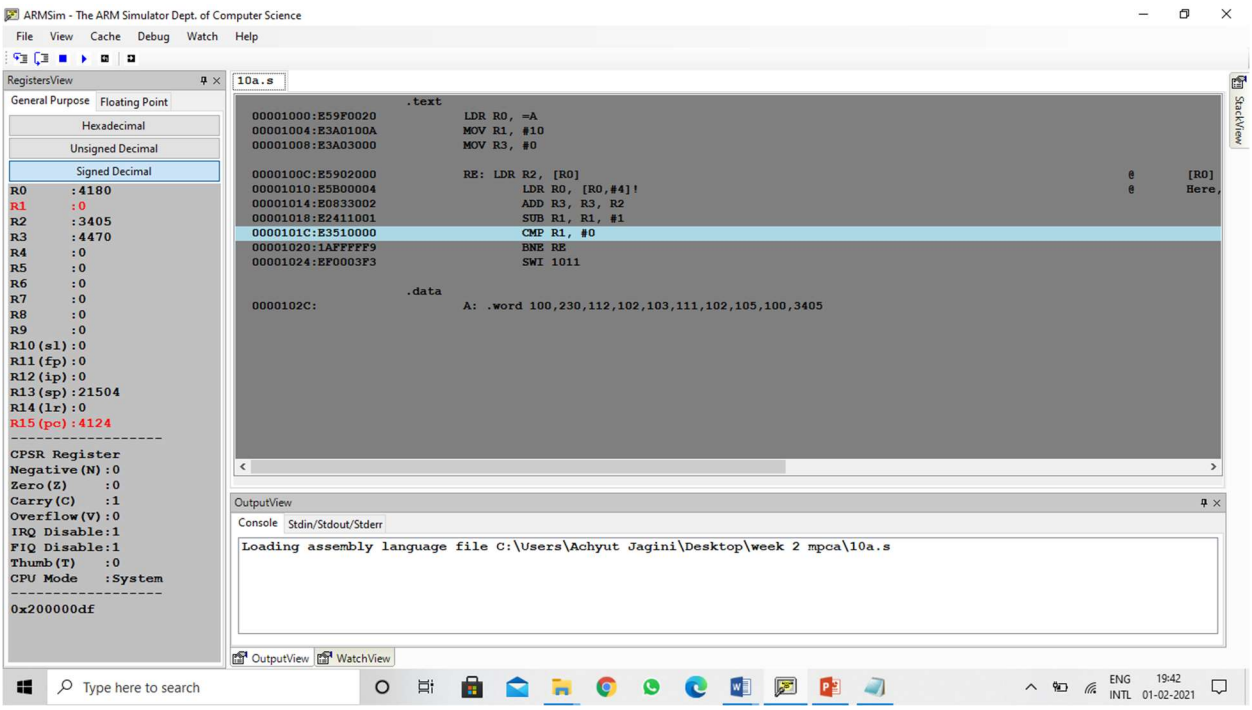
```
Loading assembly language file C:\Users\Achyut Jagini\Desktop\week 2 mpca\5a.s
PC out of valid memory range, address:00000008
```

OutputView WatchView

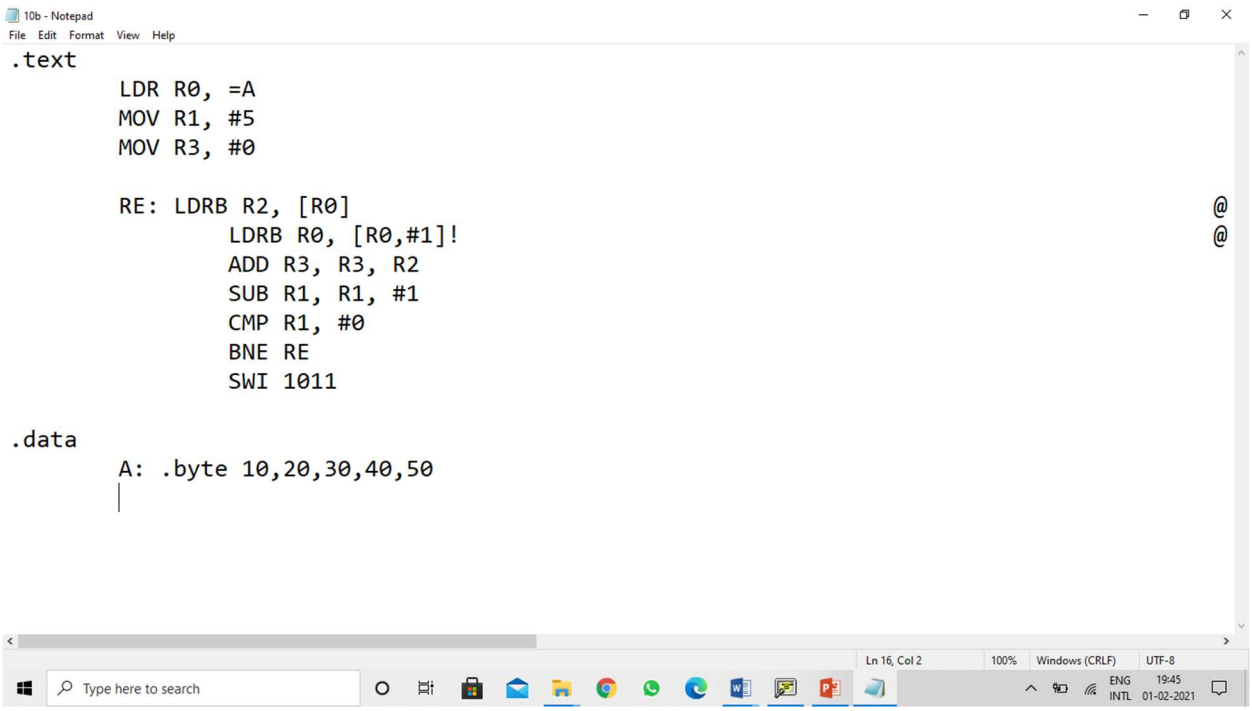
Type here to search ENG INTL 19:34 01-02-2021

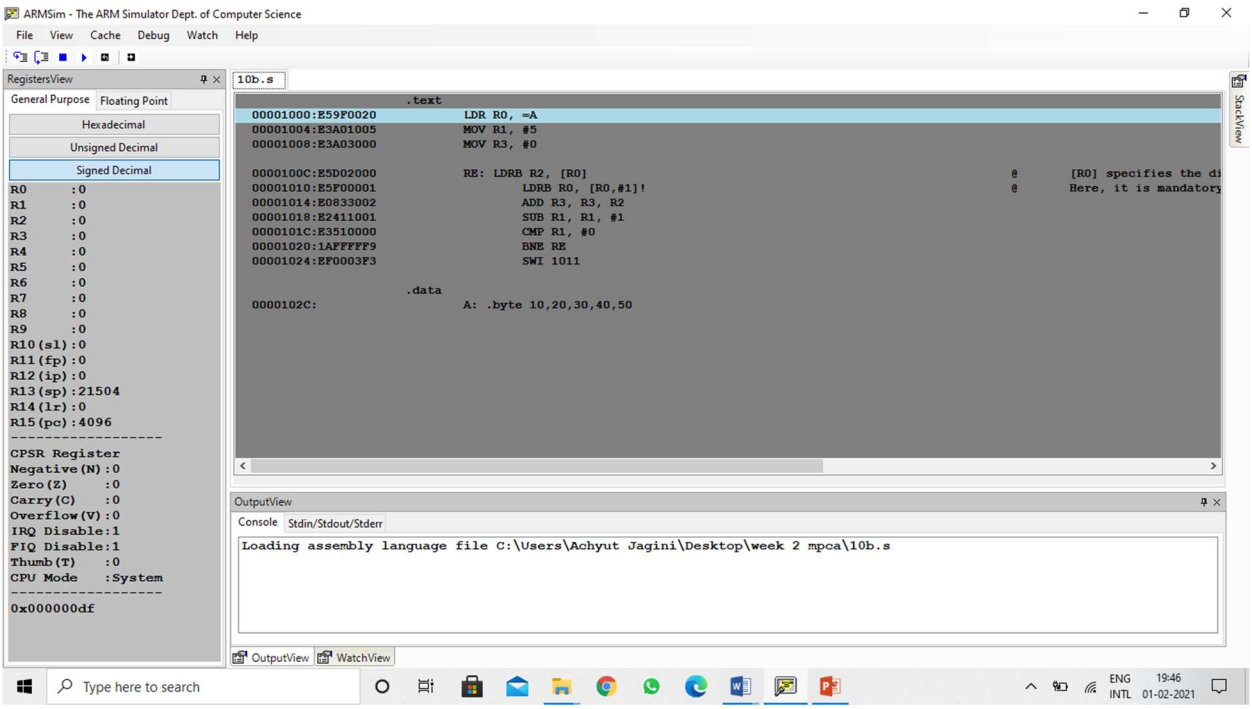
10.a) Write an ALP to add an array of ten 32 bit numbers from memory.





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(sl):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] @ [R0] specifies the di
00001010:E5F00001 LDRB R0, [R0,#1]! @ Here, it is mandatory
00001014:E0833002 ADD R3, R3, R2
00001018:E2411001 SUB R1, R1, #1
0000101C:E3510000 CMP R1, #0
00001020:1AFFFFF9 BNE RE
00001024:EF0003F3 SWI 1011
.data
0000102C: A: .byte 10,20,30,40,50
StackView
OutputView
Console Stdin/Stdout/Stderr
Loading assembly language file C:\Users\Achyut Jagini\Desktop\week 2 mpca\10b.s
OutputView WatchView
Type here to search
ENG INTL
19:46
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