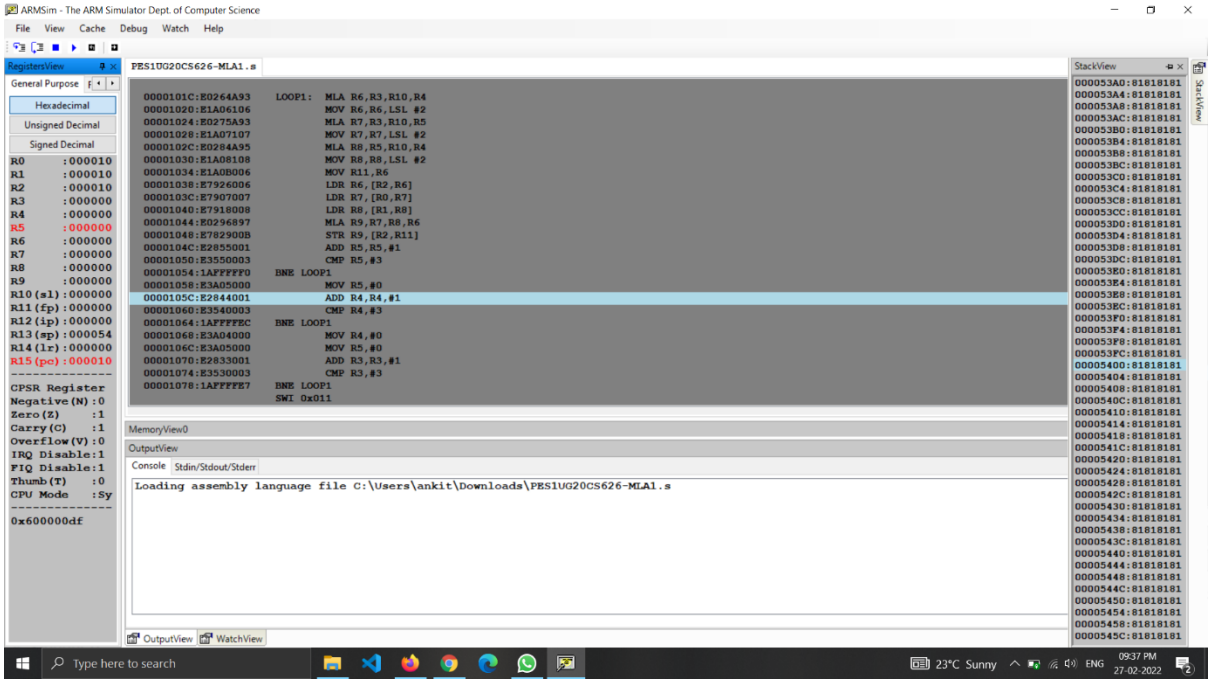
MPCA WEEK 5

NAME: Ankitha C
SRN: PES1UG20CS626
SECTION: K

1. Write a program in ARM7TDMI-ISA to multiply 2 matrices of order3.
i.e., implement c[i][j]=c[i][j] + a[i][j] x b[i][j].
a. Use MLA instruction

```
.DATA
A:.WORD 0,1,2,3,4,5,6,7,8
B:.WORD 0,1,2,3,4,5,6,7,8
C:.WORD 0,0,0,0,0,0,0,0,0

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

MOV R5,#0
MOV R3,#0
MOV R4,#0
MOV R10,#3

LOOP1:  MLA R6,R3,R10,R4
        MOV R6,R6,LSL #2
        MLA R7,R3,R10,R5
        MOV R7,R7,LSL #2
        MLA R8,R5,R10,R4
        MOV R8,R8,LSL #2
        MOV R11,R6
        LDR R6,[R2,R6]
        LDR R7,[R0,R7]
        LDR R8,[R1,R8]
        MLA R9,R7,R8,R6
        STR R9,[R2,R11]
        ADD R5,R5,#1
        CMP R5,#3
BNE LOOP1
        MOV R5,#0
        ADD R4,R4,#1
        CMP R4,#3
BNE LOOP1
```

MOV R4,#0
MOV R5,#0
ADD R3,R3,#1
CMP R3,#3
BNE LOOP1
SWI 0x011



b. Use MUL instruction

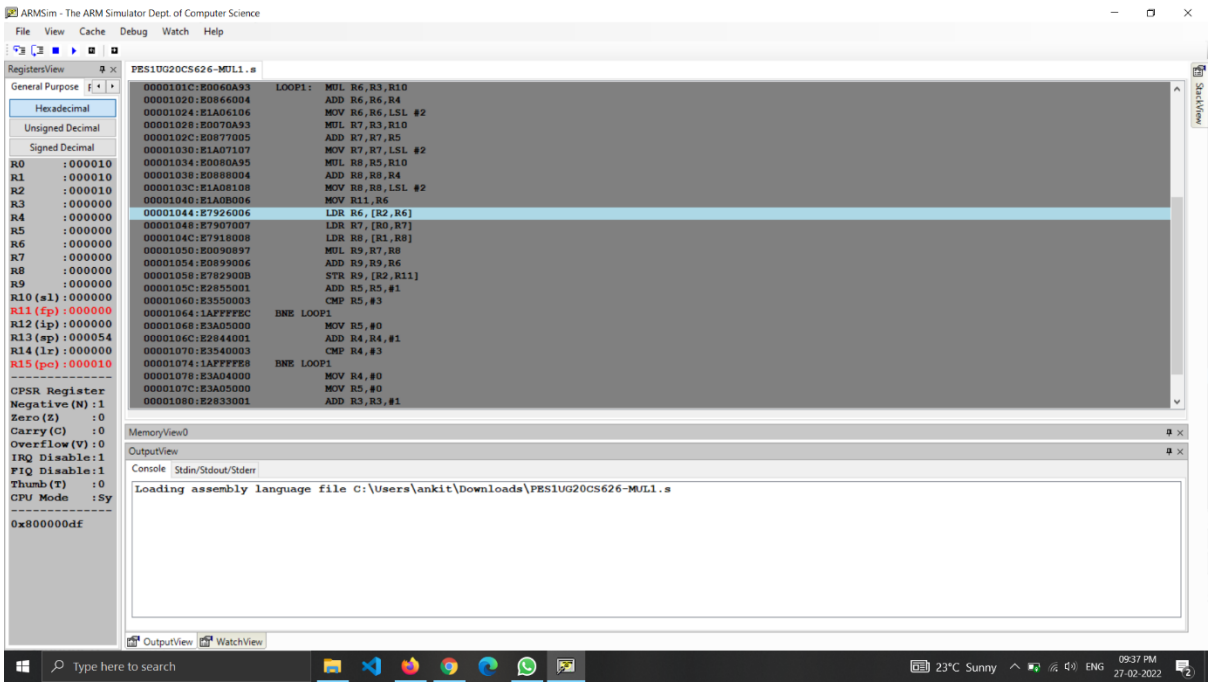
.DATA
A:.WORD 0,1,2,3,4,5,6,7,8
B:.WORD 0,1,2,3,4,5,6,7,8
C:.WORD 0,0,0,0,0,0,0,0,0

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

MOV R5,#0 ;Innermost loop index-k
MOV R3,#0 ;Outer Loop index-i
MOV R4,#0 ;Inner Loop index-j
MOV R10,#3 ;Number of elements in a row

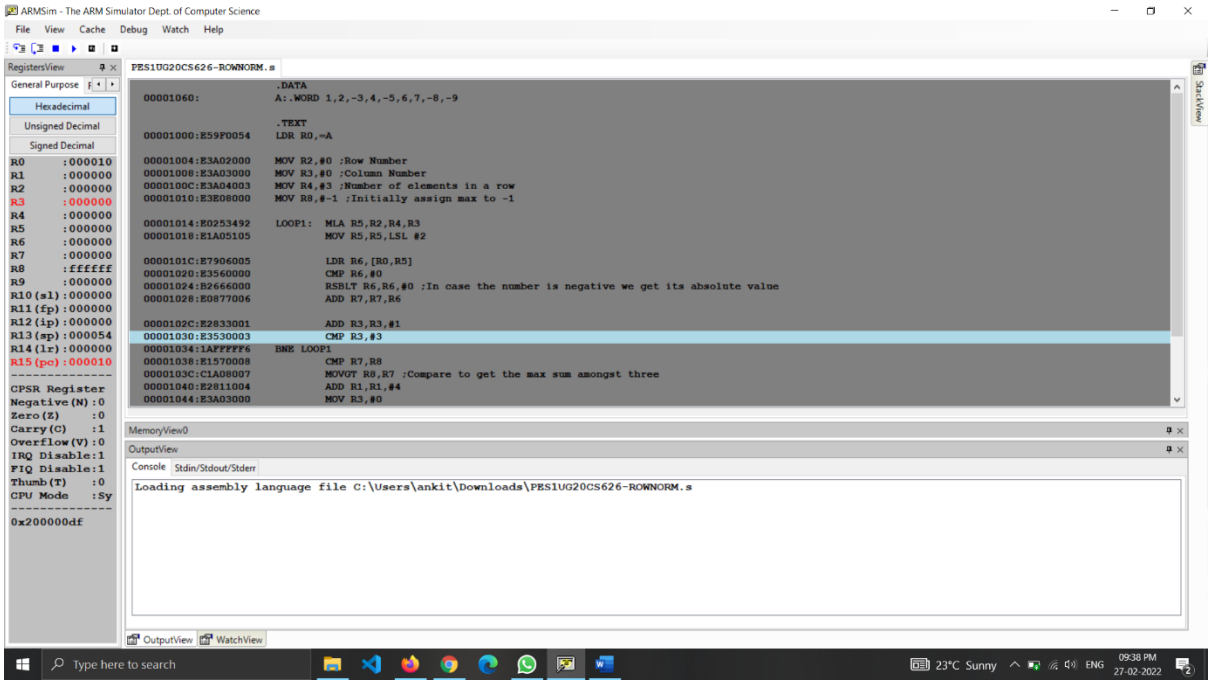
LOOP1: MUL R6,R3,R10
ADD R6,R6,R4
MOV R6,R6,LSL #2

```
        MUL R7,R3,R10
        ADD R7,R7,R5
        MOV R7,R7,LSL #2
        MUL R8,R5,R10
        ADD R8,R8,R4
        MOV R8,R8,LSL #2
        MOV R11,R6
        LDR R6,[R2,R6]
        LDR R7,[R0,R7]
        LDR R8,[R1,R8]
        MUL R9,R7,R8
        ADD R9,R9,R6
        STR R9,[R2,R11]
        ADD R5,R5,#1
        CMP R5,#3
BNE LOOP1
        MOV R5,#0
        ADD R4,R4,#1
        CMP R4,#3
BNE LOOP1
        MOV R4,#0
        MOV R5,#0
        ADD R3,R3,#1
        CMP R3,#3
BNE LOOP1
SWI 0x011
```



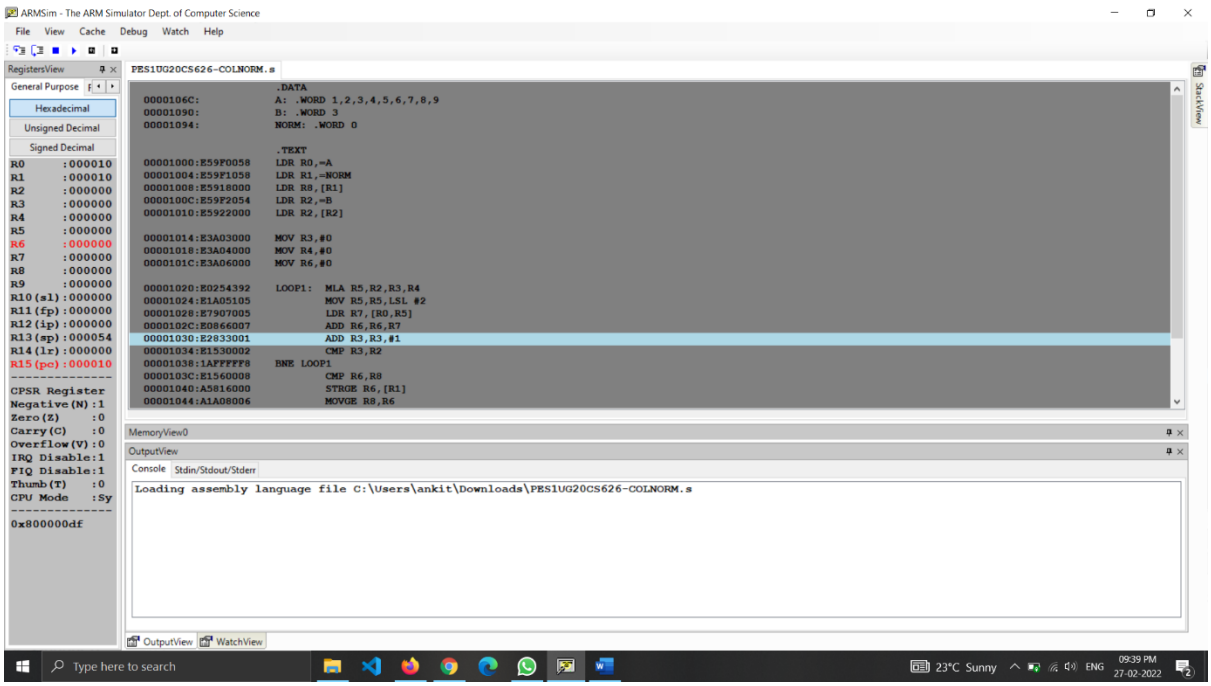
2. Write a program in ARM7TDMI-ISA to find the NORM of a square matrix of order n

```
#ROWNORM
.DATA
A:.WORD 1,2,-3,4,-5,6,7,-8,-9

.TEXT
LDR R0,=A

MOV R2,#0 ;Row Number
MOV R3,#0 ;Column Number
MOV R4,#3 ;Number of elements in a row
MOV R8,#-1 ;Initially assign max to -1

LOOP1:  MLA R5,R2,R4,R3
        MOV R5,R5,LSL #2

        LDR R6,[R0,R5]
        CMP R6,#0
        RSBLT R6,R6,#0 ;In case the number is negative we get its absolute value
        ADD R7,R7,R6

        ADD R3,R3,#1
        CMP R3,#3
BNE LOOP1
        CMP R7,R8
        MOVGT R8,R7 ;Compare to get the max sum amongst three
        ADD R1,R1,#4
        MOV R3,#0
        MOV R7,#0
        ADD R2,R2,#1
        CMP R2,#3
BNE LOOP1
SWI 0x011
```




```
#COLUMNNORM
.DATA
A: .WORD 1,2,3,4,5,6,7,8,9
B: .WORD 3
NORM: .WORD 0

.TEXT
LDR R0,=A
LDR R1,=NORM
LDR R8,[R1]
LDR R2,=B
LDR R2,[R2]

MOV R3,#0
MOV R4,#0
MOV R6,#0

LOOP1:  MLA R5,R2,R3,R4
        MOV R5,R5,LSL #2
        LDR R7,[R0,R5]
        ADD R6,R6,R7
        ADD R3,R3,#1
        CMP R3,R2
BNE LOOP1
        CMP R6,R8
        STRGE R6,[R1]
        MOVGE R8,R6
        MOV R6,#0
```

```
        MOV R3,#0
        ADD R4,R4,#1
        CMP R4,R2
BNE LOOP1
SWI 0x011
```

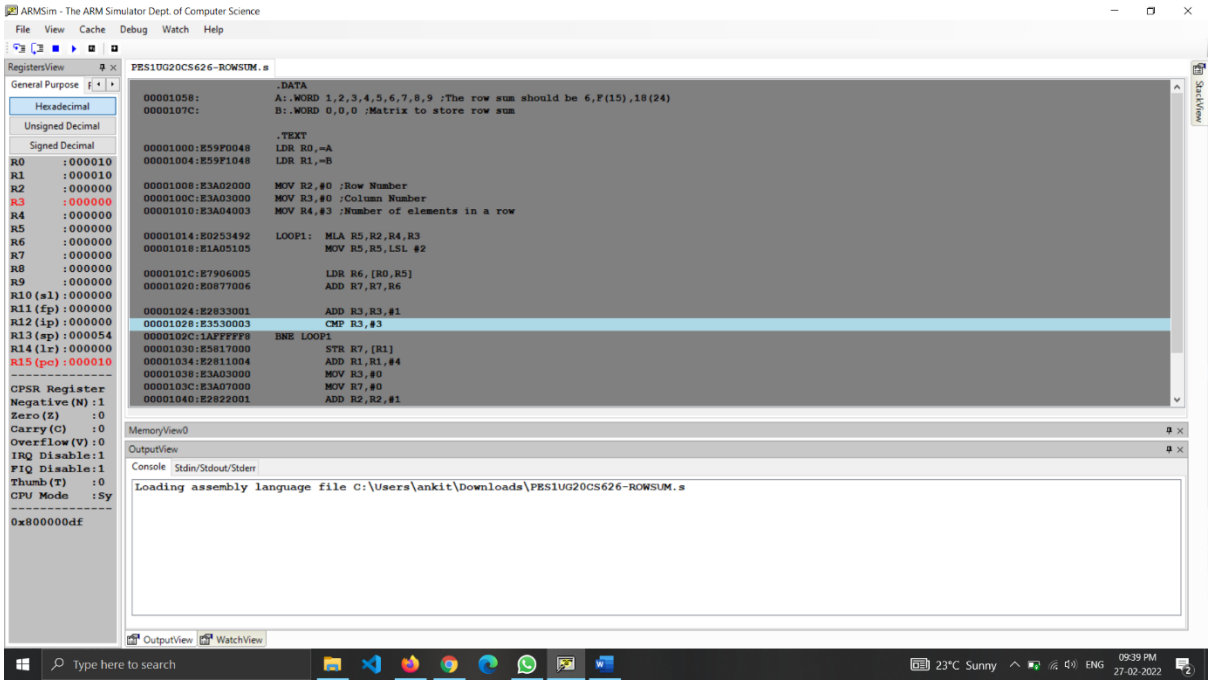


.

3. Write a program in ARM7TDMI-ISA to find the ROWSUM of a matrix.

```
.DATA

A:.WORD 1,2,3,4,5,6,7,8,9 ;The row sum should be 6,F(15),18(24)

B:.WORD 0,0,0 ;Matrix to store row sum



.TEXT

LDR R0,=A

LDR R1,=B



MOV R2,#0 ;Row Number

MOV R3,#0 ;Column Number
```

```
MOV R4,#3 ;Number of elements in a row


LOOP1:  MLA R5,R2,R4,R3
        MOV R5,R5,LSL #2


        LDR R6,[R0,R5]
        ADD R7,R7,R6


        ADD R3,R3,#1
        CMP R3,#3
BNE LOOP1
        STR R7,[R1]
        ADD R1,R1,#4
        MOV R3,#0
        MOV R7,#0
        ADD R2,R2,#1
        CMP R2,#3
BNE LOOP1
SWI 0x011
```


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :000010
R1 :000010
R2 :000000
R3 :000000
R4 :000000
R5 :000000
R6 :000000
R7 :000000
R8 :000000
R9 :000000
R10(sl):000000
R11(fp):000000
R12(ip):000000
R13(sp):000054
R14(lr):000000
R15(pc):000010
CPSR Register
Negative(N):1
Zero(Z) :0
Carry(C) :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :Sy
0x800000df
PES1UG20CS626-ROWSUM.s
.DATA
00001058: A:.WORD 1,2,3,4,5,6,7,8,9 ;The row sum should be 6,F(15),18(24)
0000107C: B:.WORD 0,0,0 ;Matrix to store row sum
.TEXT
00001000:E59F0048 LDR R0,=A
00001004:E59F1048 LDR R1,=B
00001008:E3A02000 MOV R2,#0 ;Row Number
0000100C:E3A03000 MOV R3,#0 ;Column Number
00001010:E3A04003 MOV R4,#3 ;Number of elements in a row
00001014:E0253492 LOOP1: MLA R5,R2,R4,R3
00001018:E1A05105 MOV R5,R5,LSL #2
0000101C:E7906005 LDR R6,[R0,R5]
00001020:E0877006 ADD R7,R7,R6
00001024:E2833001 ADD R3,R3,#1
00001028:E3530003 CMP R3,#3
0000102C:1AFFFFF8 BNE LOOP1
00001030:E5817000 STR R7,[R1]
00001034:E2811004 ADD R1,R1,#4
00001038:E3A03000 MOV R3,#0
0000103C:E3A07000 MOV R7,#0
00001040:E2822001 ADD R2,R2,#1
MemoryView0
OutputView
Console
Stdin/Stdout/Stderr
Loading assembly language file C:\Users\ankit\Downloads\PES1UG20CS626-ROWSUM.s
OutputView
WatchView
StackView
Type here to search
23°C Sunny
ENG
09:39 PM
27-02-2022