

WEEK-4

1. Write an ALP to read from a 2D array such that
 $B=a[i][j]$
2. Write an ALP to implement $C[k]=a[i]+b[j]$
3. Write an ALP to implement $Sum[i]+=a[i][j]$
4. Write an ALP to implement $C[i][j]=a[i][j]+b[i][j]$

Name: Atharva Menkudle

SRN: PES2UG21CS104

Sec: B

Date: 24-02-2023

1. Write an ALP to read from a 2D array such that
 $B=a[i][j]$

Code:

```
.data
a: .word 1,2,3,4,5,6,7,8,9
b: .word 0

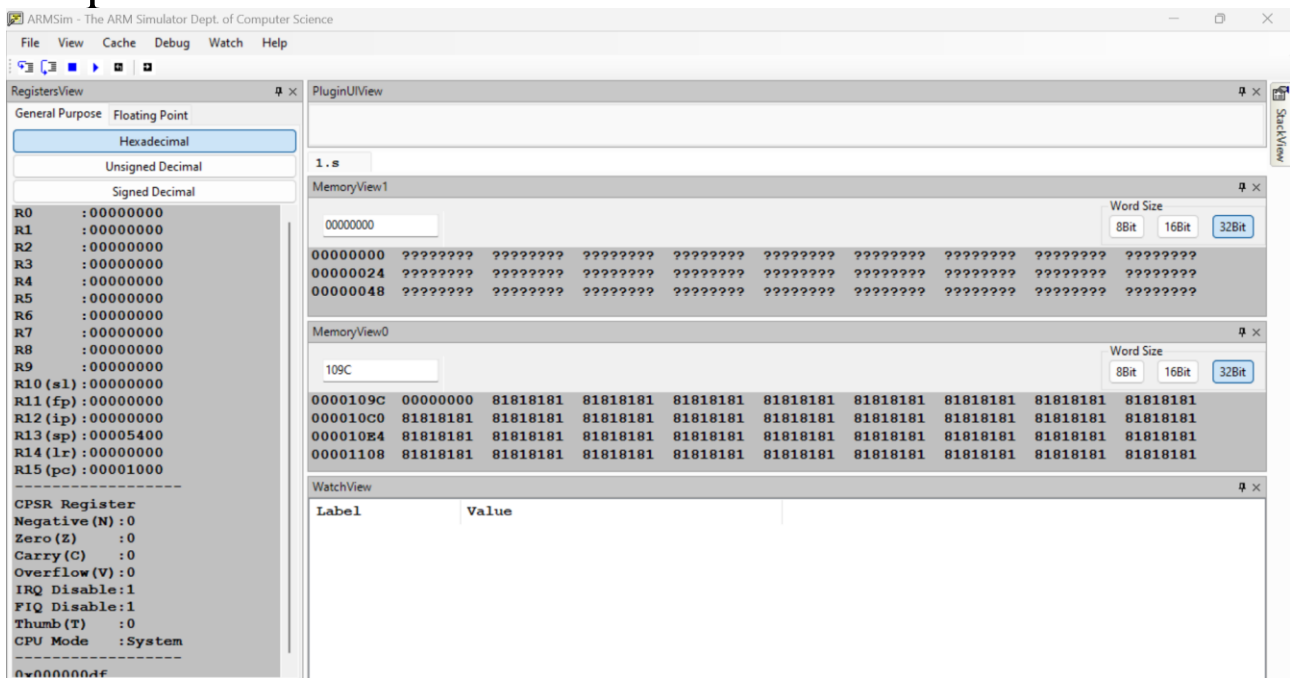
.text 1
dr r0,=a
ldr r1,=b
mov r2,#3
mov r3,#3
mov r4,#0
mov r5,#0
for_i:
for_j:
    stmfd r13!,{r4,r5}
    bl get_addr
    ldmfd r13!,{r4,r5,r6} \
    add r7,r0,r6
    add r8,r1,r6
    ldr r6,[r7]
    str r6,[r8]
    add r5,r5,#1
    cmp r5,r3
    bne for_j
```

```

    mov r5,#0
    add r4,r4,#1
    cmp r4,r2
    beq exit
    b for_i
get_addr:
    ldmdf r13!,{r4,r5}
    mla r7,r3,r4,r5
    mov r8,#4
    mul r6,r7,r8
    stmfd r13!,{r4,r5,r6}
bx lr
exit: SWI 0x011
.end

```

Output Screenshot:



2. Write an ALP to implement $C[k]=a[i]+b[j]$
Code:

```

.data
a: .word 10,20,30,40,50
b: .word 10,20,30,40,50
c: .word 0,0,0,0,0
.text
ldr r0,=a
ldr r1,=b
ldr r2,=c
mov r6,#5

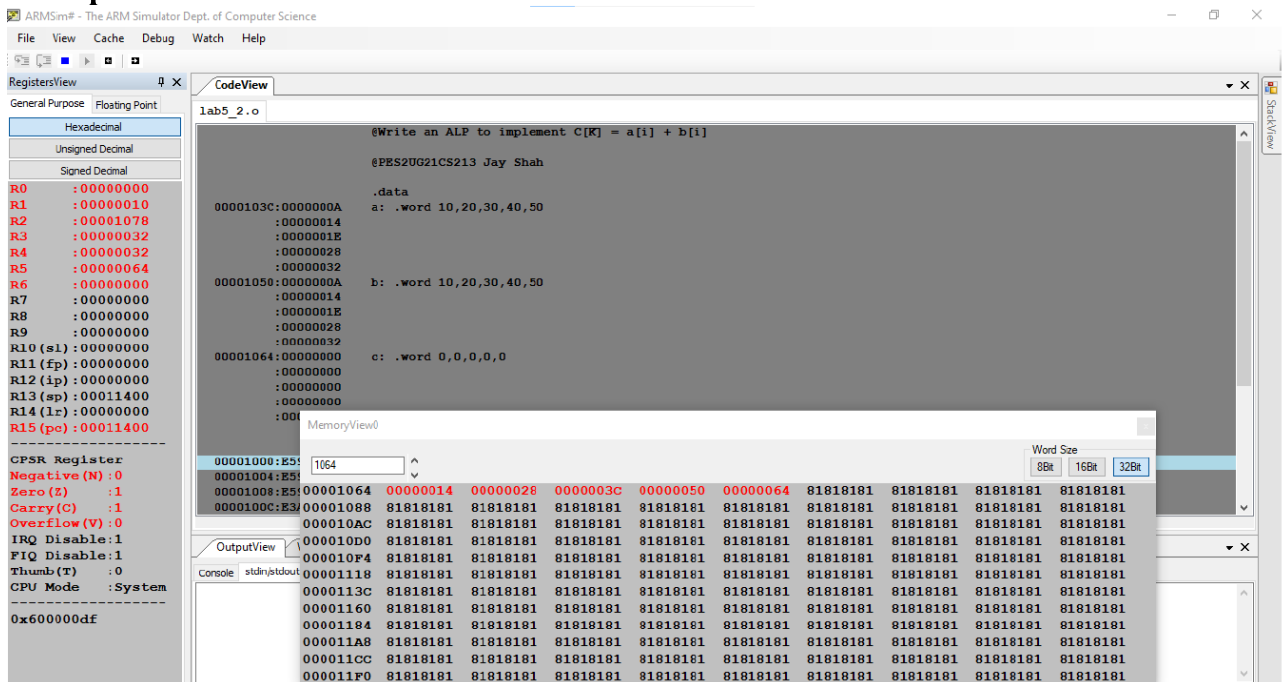
```

```

loop:
ldr r3,[r0],#4
ldr r4,[r1],#4
add r5,r3,r4
str r5,[r2],#4
sub r6,r6,#1
cmp r6,#0
bne loop
close : swi 0x11
.end

```

Output Screenshot:



3. Write an ALP to implement $\text{Sum}[i] += a[i][j]$

Code:

```

.data
a: .word 1,2,3,4,5,6,7,8,9
.text
ldr r0,=a
mov r1,#0 @ =Sum
mov r2,#3 @ =rows
mov r3,#3 @ =columns
mov r4,#0 @ =i
mov r5,#0 @ =j
for_i:
for_j:
stmfd r13!,{r4,r5}
bl get_addr

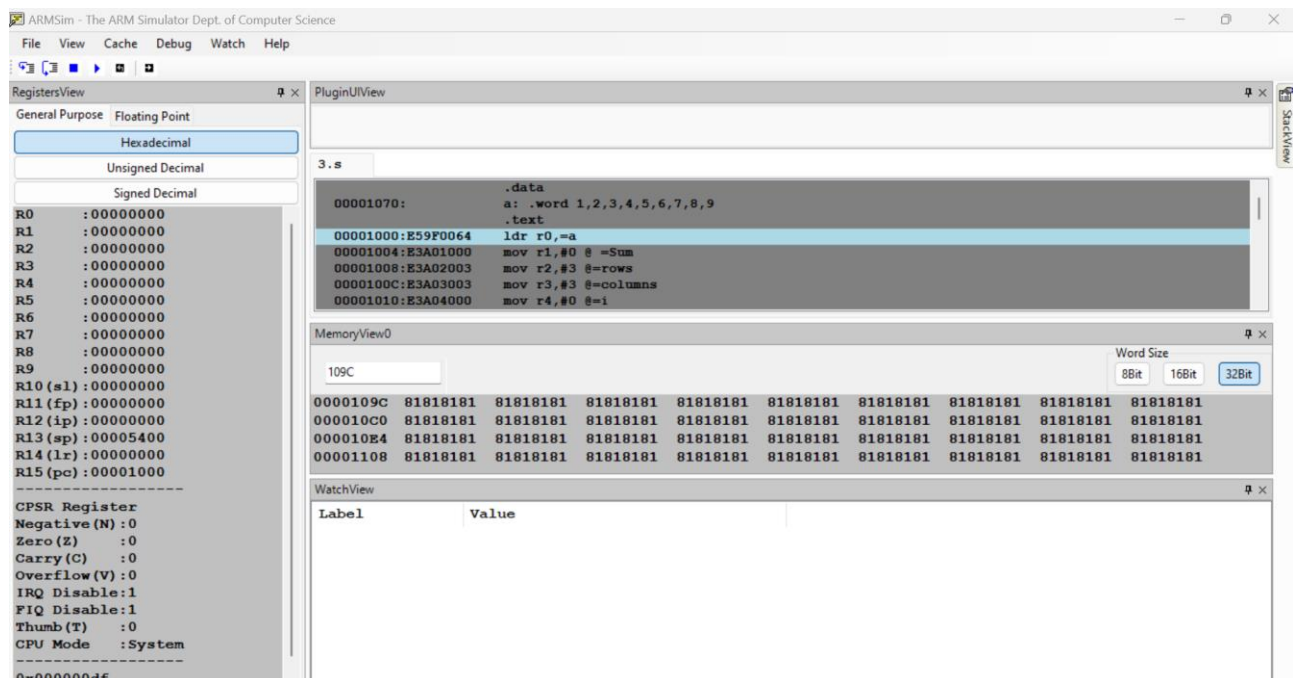
```

```

ldmfd r13!,{r4,r5,r6}
add r6,r0,r6
ldr r6,[r6]
add r1,r1,r6 @sum[i] += a[i][j]
add r5,r5,#1
cmp r5,r3
bne for_j
mov r5,#0
add r4,r4,#1
cmp r4,r2
beq exit
b for_i
get_addr:
ldmfd r13!,{r4,r5}
mla r7,r3,r4,r5
mov r8,#4
mul r6,r7,r8
stmfd r13!,{r4,r5,r6}
bx lr
exit: swi 0x011
.end

```

Output Screenshot:



4. Write an ALP to implement $C[i][j] = a[i][j] + b[i][j]$

Code:

```
.data
a: .word 1,2,4,5,6,7,8,9
b: .word 1,2,4,5,6,7,8,9
c: .word 0
.text
ldr r0,=a
ldr r1,=b
ldr r2,=c
mov r3,#3
mov r4,#3
mov r5,#0
mov r6,#0
for_i:
for_j:
stmfd r13!,{r5,r6}
bl get_addr
ldmfd r13!,{r5,r6,r7}
add r8,r0,r7
add r9,r1,r7
ldr r8,[r8]
ldr r9,[r9]
add r8,r8,r9
add r9,r2,r7
str r8,[r9]
add r6,r6,#1
cmp r6,r4
bne for_j
mov r6,#0
add r5,r5,#1
cmp r5,r3
beq exit
b for_i
get_addr:
ldmfd r13!,{r5,r6}
mla r8,r4,r5,r6
mov r9,#4
mul r7,r8,r9
stmfd r13!,{r5,r6,r7}
bx lr
exit: swi 0x011
.end
```

Output Screenshot:

ARMSim - The ARM Simulator Dept. of Computer Science

File View Cache Debug Watch Help

RegistersView Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 0000108c
R1 : 000010ac
R2 : 000010cc
R3 : 00000003
R4 : 00000003
R5 : 00000003
R6 : 00000000
R7 : 00000020
R8 : 00000003
R9 : 000010ec
R10 (s1) : 00000000
R11 (fp) : 00000000
R12 (ip) : 00000000
R13 (sp) : 00000000
R14 (lr) : 00001080
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

PluginUI View

4.s

```
00001000:E59F0078 ldr r0,=a  
00001004:E59F1078 ldr r1,=b  
00001008:E59F2078 ldr r2,=c  
0000100C:E3A03003 mov r3,#3  
00001010:E3A04003 mov r4,#3  
00001014:E3A05000 mov r5,#0  
00001018:E3A06000 mov r6,#0  
0000101C: for i:
```

MemoryView0

Word Size
8Bit 16Bit 32Bit

10CC

| | | | | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 000010CC | 00000002 | 00000004 | 00000008 | 0000000A | 0000000C | 0000000E | 00000010 | 00000012 | 00000003 |
| 000010F0 | 81818181 | 81818181 | 81818181 | 81818181 | 81818181 | 81818181 | 81818181 | 81818181 | 81818181 |
| 00001114 | 81818181 | 81818181 | 81818181 | 81818181 | 81818181 | 81818181 | 81818181 | 81818181 | 81818181 |
| 00001138 | 81818181 | 81818181 | 81818181 | 81818181 | 81818181 | 81818181 | 81818181 | 81818181 | 81818181 |

WatchView

| Label | Value |
|-------|-------|
|-------|-------|