**Microprocessor and Computer Architecture**

**UE22CS251B**

# LAB-4

**4th Semester, Academic Year 2023-2024**

Date:08-02-2024

| | | |
|---|---|---|
| Name: V V Mohith | SRN:-PES2UG22CS641 | Section:-K |

Week#____4_____

## PROGRAM :- 1(a)

(A)Write an ALP to perform Convolution using MUL instruction (Addition of

multiplication of respective numbers of loc A and loc B)

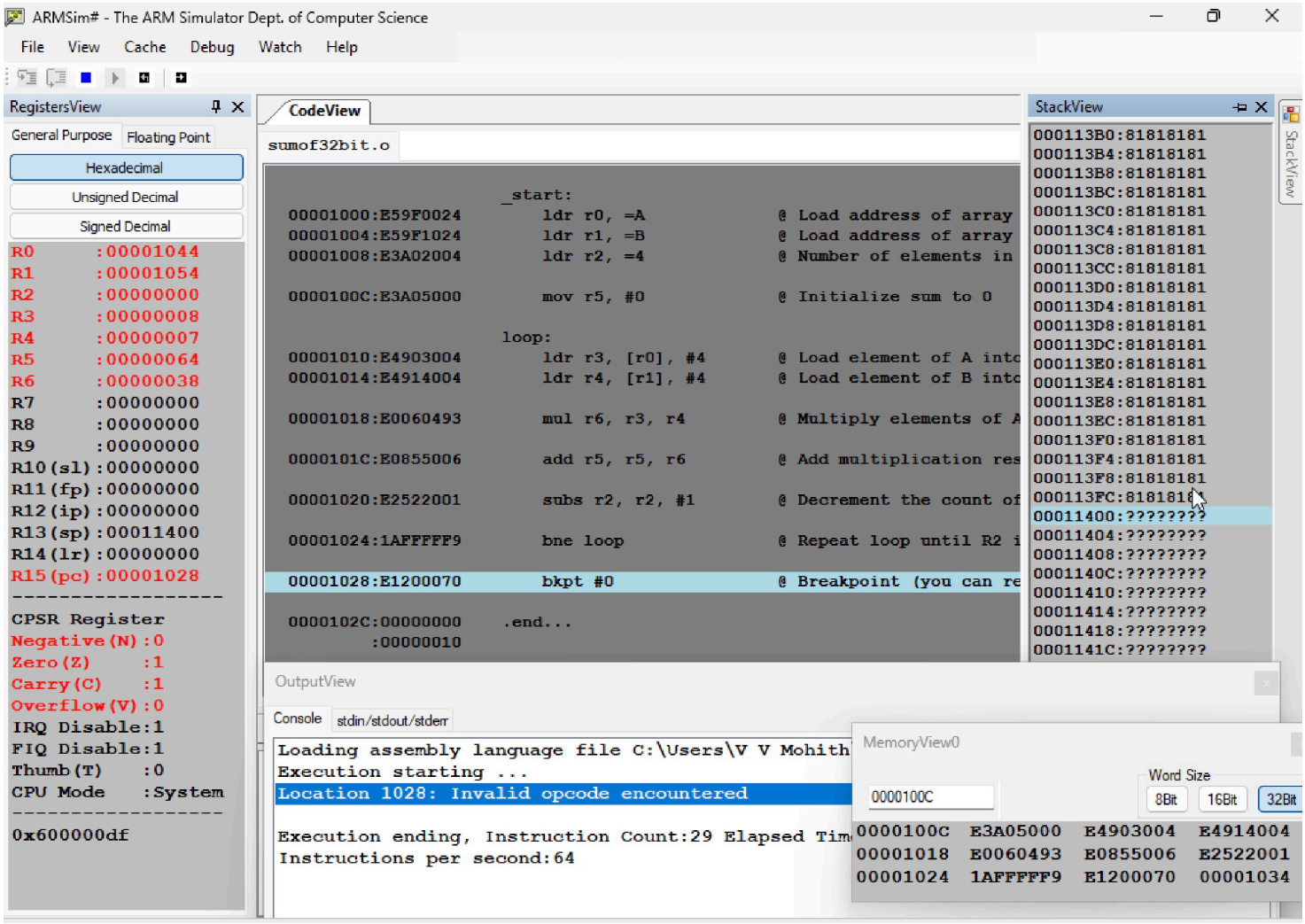
## COMMANDS:-

```
.data
A:   .word 2, 4, 6, 8
B:   .word 1, 3, 5, 7
```

```
.text
.global _start

_start:
    ldr r0, =A
   ldr r1, =B
    ldr r2, =4

    mov r5, #0

loop:
    ldr r3, [r0], #4
   ldr r4, [r1], #4

    mul r6, r3, r4

    add r5, r5, r6

    subs r2, r2, #1

    bne loop
```

```
        bkpt #0

.end
```

## OUTPUT SCREENSHOT:



**Microprocessor and Computer Architecture**

**UE22CS251B**

**4th Semester, Academic Year 2023-2024**

Date:08-02-2024

| | | |
|---|---|---|
| Name: V V Mohith | SRN:PES2UG22CS641 | Section:-K |

Week#____4_____

## PROGRAM :- 1(b)

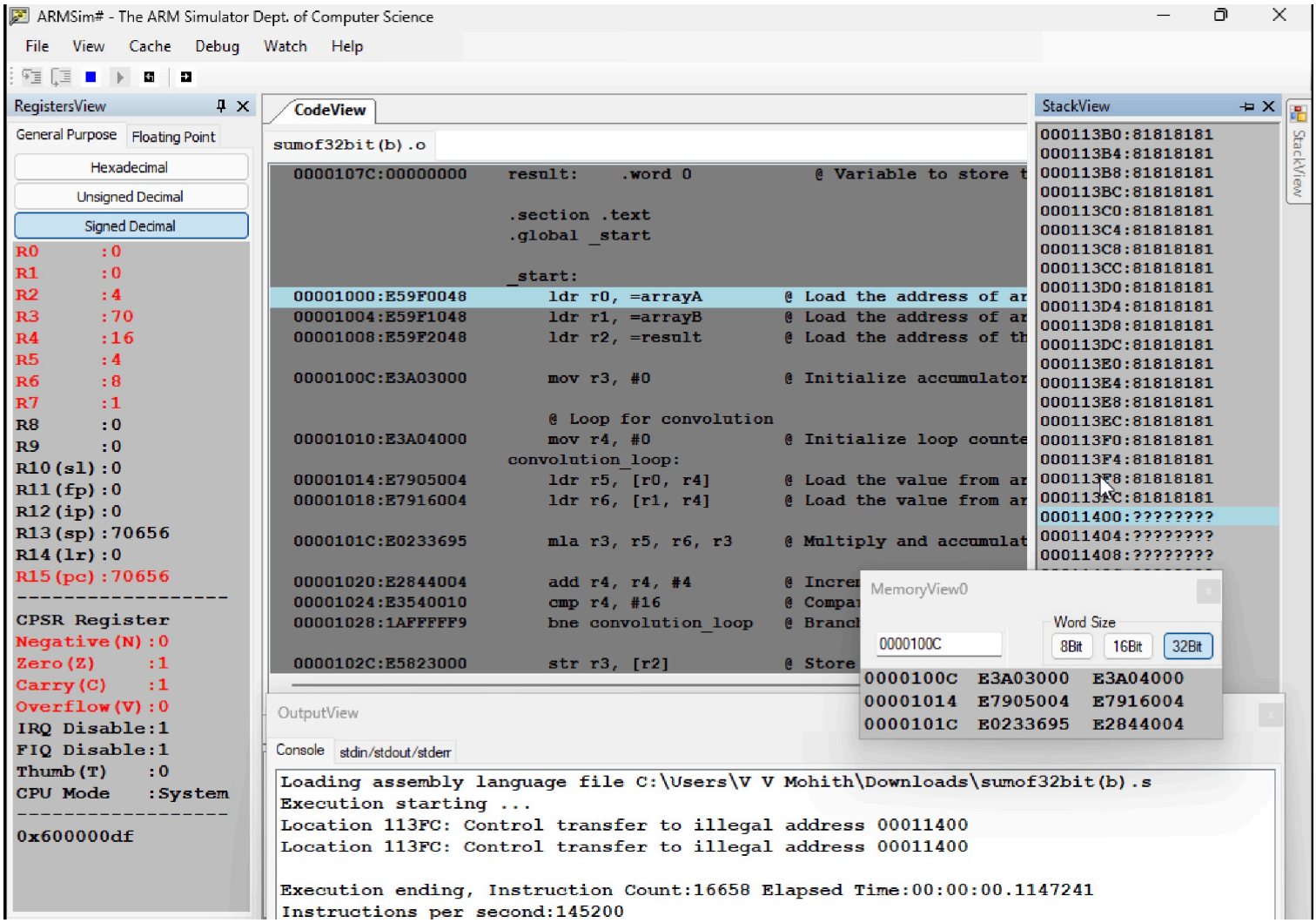
Write an ALP to perform Convolution using MLA instruction (Addition of multiplication of respective numbers of loc A and loc B).

## Commands:-

```
_start:
    ldr r0, =arrayA
    ldr r1, =arrayB
    ldr r2, =result

    mov r3, #0

    mov r4, #0
convolution_loop:
    ldr r5, [r0, r4]
    ldr r6, [r1, r4]

    mla r3, r5, r6, r3

    add r4, r4, #4
```

```
cmp r4, #16
bne convolution_loop

str r3, [r2]

mov r0, #1
ldr r1, =result
ldr r2, =4
mov r7, #4
swi 0

mov r7, #1
mov r0, #0
swi 0
```

## OUTPUT SCREENSHOT:




# Microprocessor and Computer Architecture

## UE22CS251B

## 4th Semester, Academic Year 2023-2024

Date:08-02-2024

| Name: V V Mohith | SRN:-PES2UG22CS641 | Section:-K |
|---|---|---|

Week#____4_____

## PROGRAM:-2

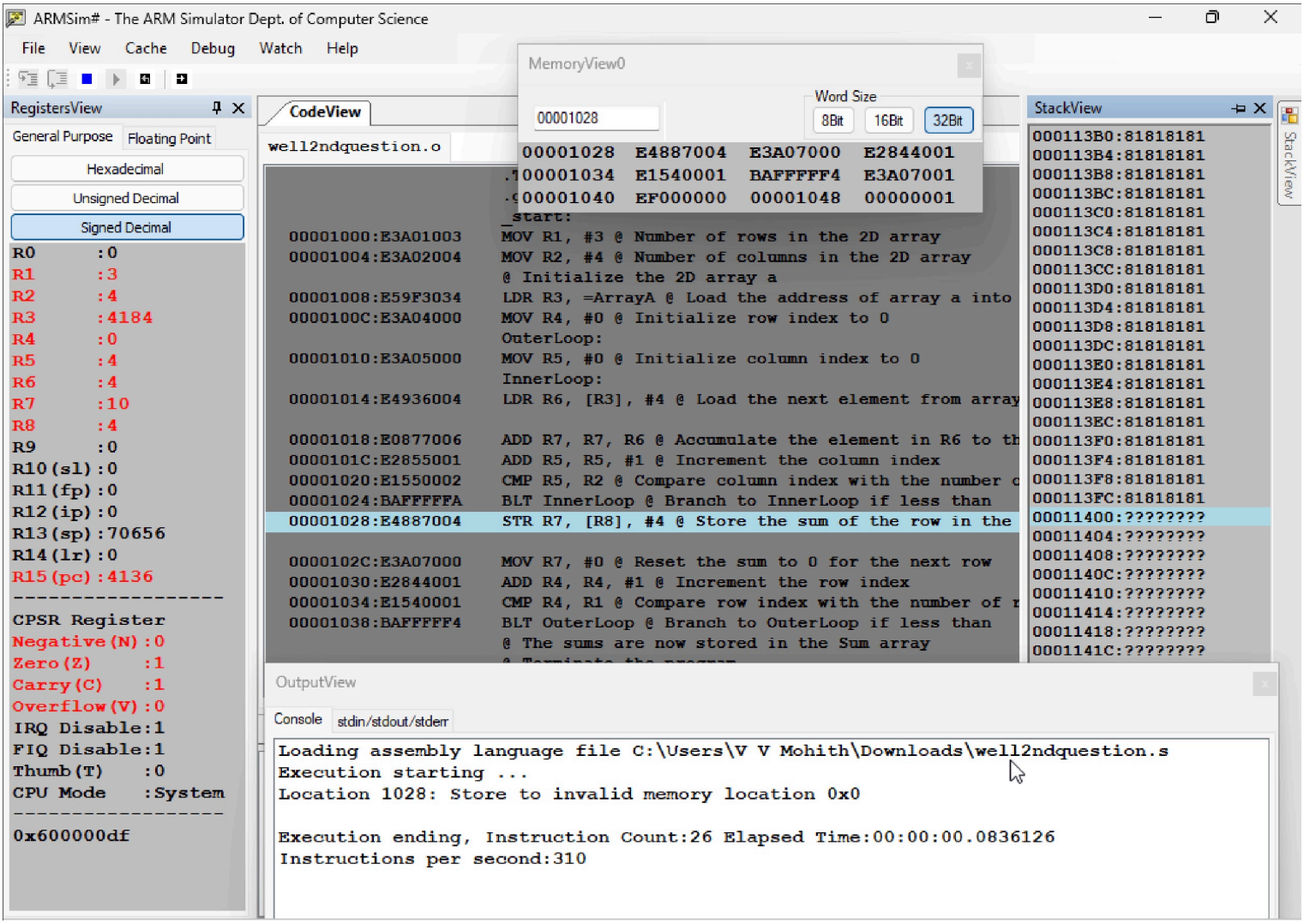
Write an ALP to implement Sum[I]+=a[i][j]

**Commands:-**

```
.global _start
_start:
MOV R1, #3
MOV R2, #4
LDR R3, =ArrayA
MOV R4, #0
OuterLoop:
MOV R5, #0
InnerLoop:
LDR R6, [R3], #4
ADD R7, R7, R6
ADD R5, R5, #1
CMP R5, R2
BLT InnerLoop
```

```
STR R7, [R8], #4
MOV R7, #0
ADD R4, R4, #1
CMP R4, R1
BLT OuterLoop
MOV R7, #1
SWI 0
.DATA
ArrayA: .WORD 1, 2, 3, 4
.WORD 5, 6, 7, 8
.WORD 9, 10, 11, 12
Sum: .SPACE 12
```

OUTPUT SCREENSHOT:-




# Microprocessor and Computer Architecture

## UE22CS251B

## 4th Semester, Academic Year 2023-2024

Date:08-02-2024

| Name: V V Mohith | SRN:-PES2UG22CS641 | Section:-K |
|---|---|---|

Week#____4_____

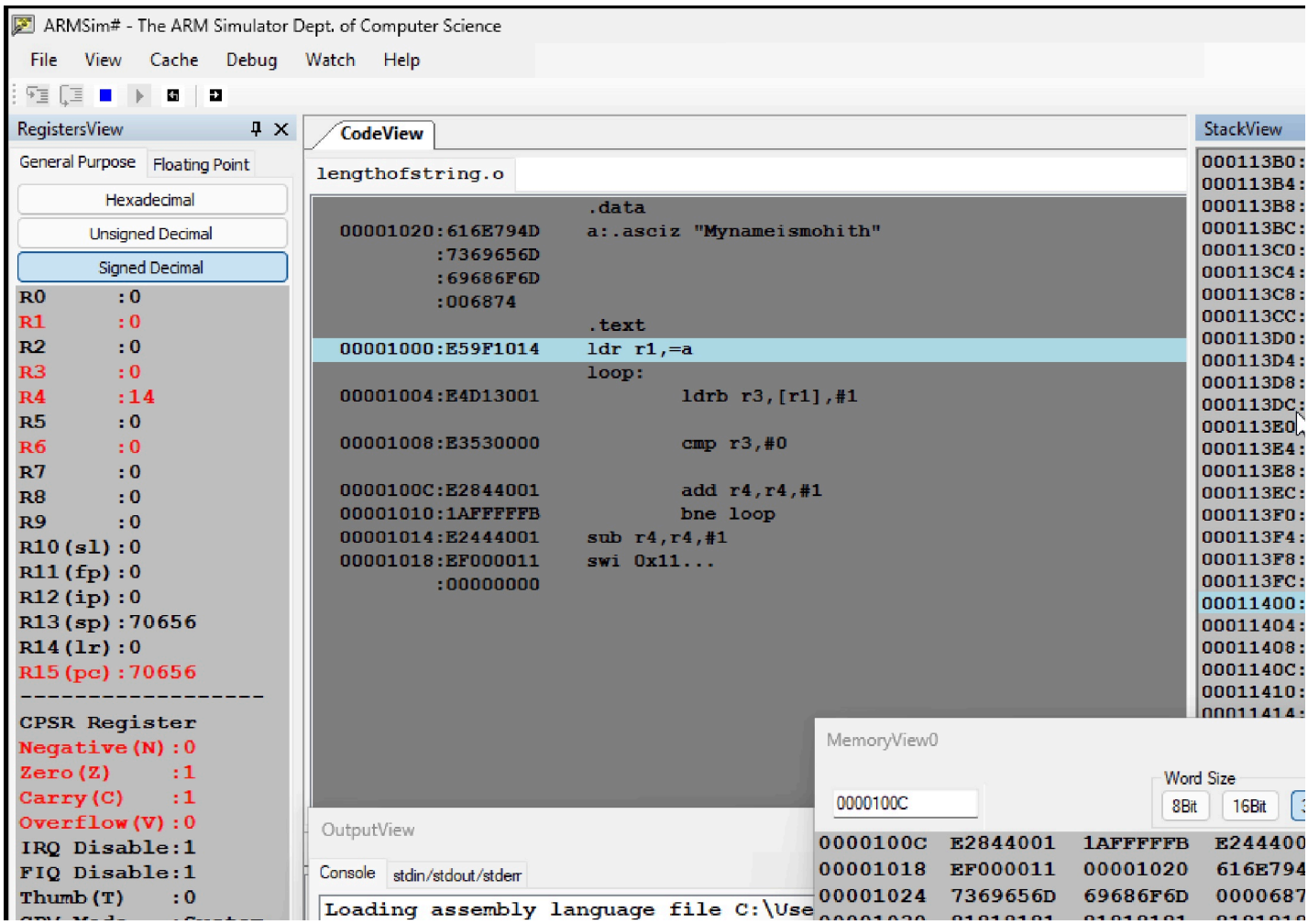
## PROGRAM:-3

## Write an ALP to find the length of a given string

## COMMAND:-

```
.data
a:.asciz "Myname"
.text
ldr r1,=a
loop:
        ldrb r0,[r1],#1
        add r10,r10,#1
        cmp r0,#0
        bne loop
swi 0x11
    BNE LOOP
    BEQ EXIT
EXIT: SWI 0x011
.end
```

## OUTPUT SCREENSHOT:-




**Microprocessor and Computer Architecture**

**UE22CS251B**

**4th Semester, Academic Year 2023-2024**

Date:08-02-2024

| Name: V V Mohith | SRN:PES2UG22CS641 | Section:-K |
|---|---|---|

Week#____4_____

**PROGRAM:-4**

**Write an ALP to copy string from one location to another**

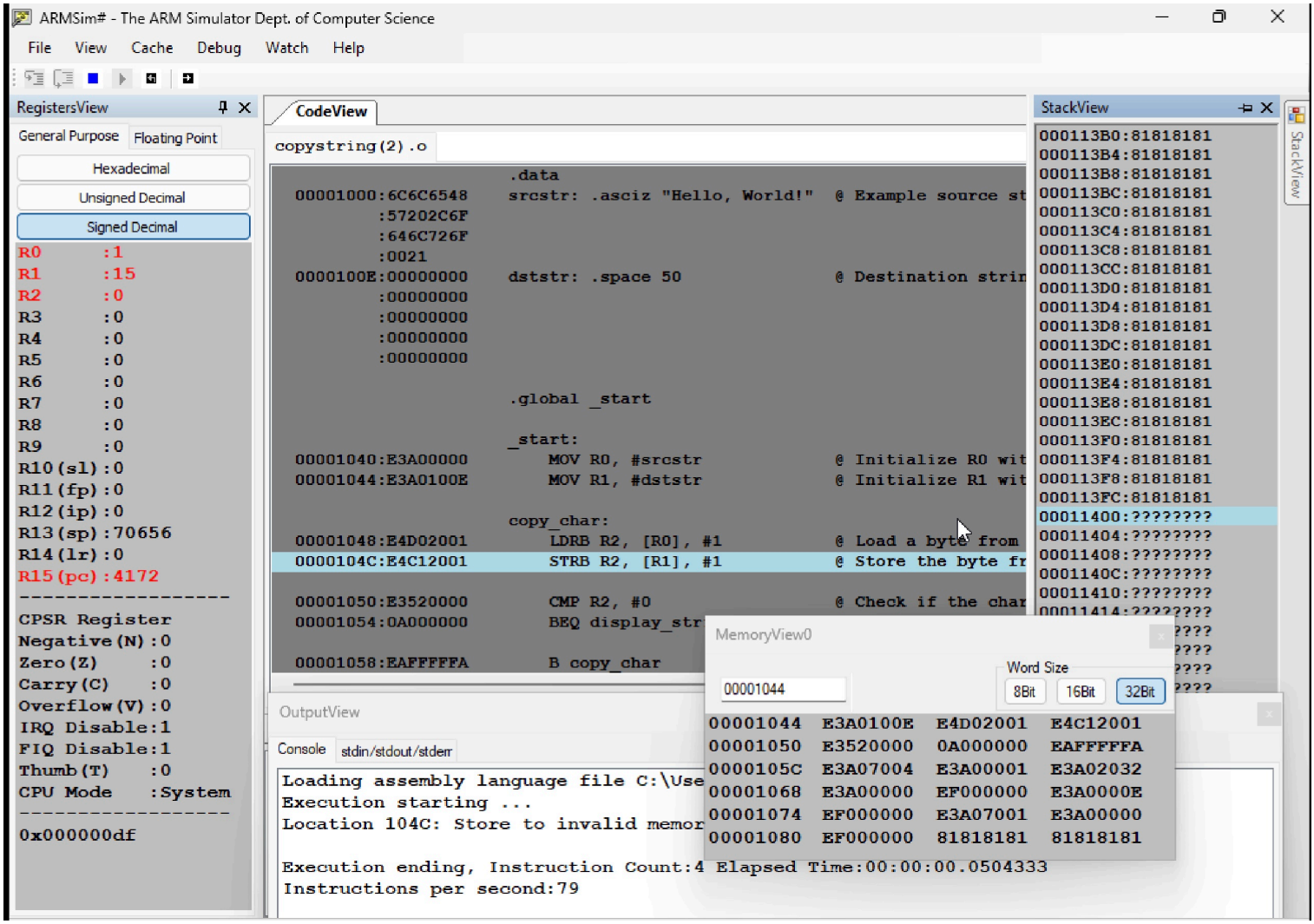
**COMMANDS:-**

```
.data
srcstr: .asciz "Hello, World!"
dststr: .space 50

.global _start

_start:
    MOV R0, #srcstr
    MOV R1, #dststr

copy_char:
```

```
    LDRB R2, [R0], #1
    STRB R2, [R1], #1

    CMP R2, #0
    BEQ display_strings

    B copy_char

display_strings:
    MOV R7, #4
    MOV R0, #1
    MOV R2, #50

    MOV R0, #srcstr
    SWI 0

    MOV R0, #dststr
    SWI 0

    MOV R7, #1
    MOV R0, #0
```

SWI 0

## OUTPUTSCREENSHOT:



**Microprocessor and Computer Architecture**

**UE22CS251B**

**4th Semester, Academic Year 2023-2024**

Date:08-02-2024

| Name: V V Mohith | SRN:PES2UG22CS641 | Section:-K |
|---|---|---|

Week#____4_____

## PROGRAM:-1(assignment)

Write an ALP to find whether a given character is present in a string. If
present, find how many times the given character is present in a string.

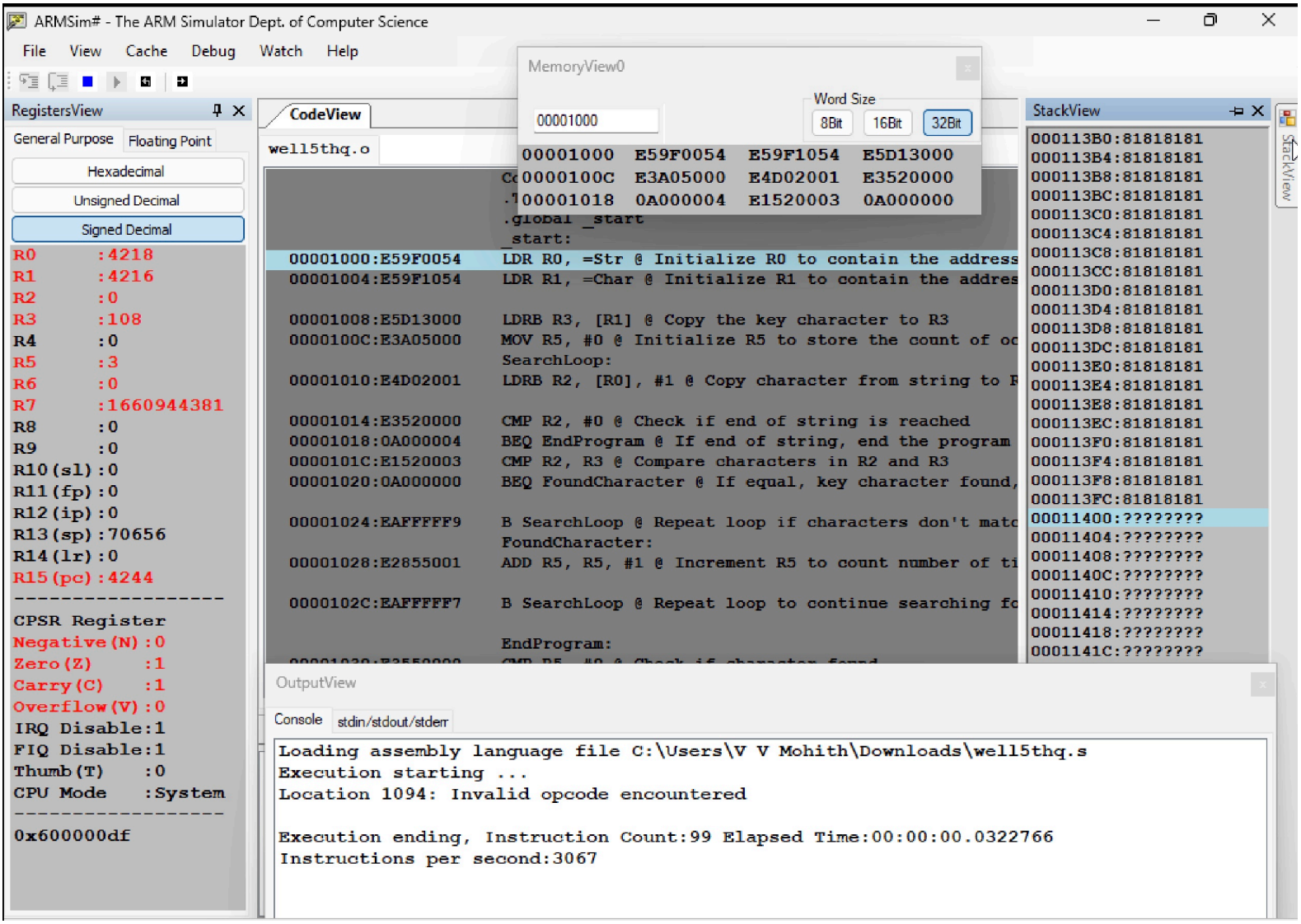
## COMMANDS:-

```
.TEXT

.global _start

_start:
```

```
LDR R0, =Str
LDR R1, =Char
LDRB R3, [R1]
MOV R5, #0
SearchLoop:
LDRB R2, [R0], #1
CMP R2, #0
BEQ EndProgram
CMP R2, R3
BEQ FoundCharacter
B SearchLoop
FoundCharacter:
ADD R5, R5, #1
B SearchLoop
EndProgram:
CMP R5, #0
BEQ CharacterNotFound
MOV R7, #4
LDR R0, =FoundMsg
SWI 0
B Exit
```

```
CharacterNotFound:
MOV R7, #4
LDR R0, =NotFoundMsg
SWI 0
Exit:
MOV R7, #1
SWI 0
.DATA
Str: .ASCIZ "hello world"
Char: .ASCIZ "l"
FoundMsg: .ASCIZ "Character Found\n"
NotFoundMsg: .ASCIZ "Character Not Found\n"
```

## OUTPUTSCREESHOT:-


ARMSim# - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
MemoryView0
Word Size
8Bit 16Bit 32Bit
00001000 E59F0054 E59F1054 E5D13000
0000100C E3A05000 E4D02001 E3520000
00001018 0A000004 E1520003 0A000000
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
R0 :4218
R1 :4216
R2 :0
R3 :108
R4 :0
R5 :3
R6 :0
R7 :1660944381
R8 :0
R9 :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):70656
R14(lr):0
R15(pc):4244
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 :System
0x600000df
CodeView
well5thq.o
.global _start
_start:
00001000:E59F0054 LDR R0, =Str @ Initialize R0 to contain the address
00001004:E59F1054 LDR R1, =Char @ Initialize R1 to contain the addres
00001008:E5D13000 LDRB R3, [R1] @ Copy the key character to R3
0000100C:E3A05000 MOV R5, #0 @ Initialize R5 to store the count of oc
SearchLoop:
00001010:E4D02001 LDRB R2, [R0], #1 @ Copy character from string to R
00001014:E3520000 CMP R2, #0 @ Check if end of string is reached
00001018:0A000004 BEQ EndProgram @ If end of string, end the program
0000101C:E1520003 CMP R2, R3 @ Compare characters in R2 and R3
00001020:0A000000 BEQ FoundCharacter @ If equal, key character found,
00001024:EAFFFFF9 B SearchLoop @ Repeat loop if characters don't matc
FoundCharacter:
00001028:E2855001 ADD R5, R5, #1 @ Increment R5 to count number of ti
0000102C:EAFFFFF7 B SearchLoop @ Repeat loop to continue searching fo
EndProgram:
StackView
000113B0:81818181
000113B4:81818181
000113B8:81818181
000113BC:81818181
000113C0:81818181
000113C4:81818181
000113C8:81818181
000113CC:81818181
000113D0:81818181
000113D4:81818181
000113D8:81818181
000113DC:81818181
000113E0:81818181
000113E4:81818181
000113E8:81818181
000113EC:81818181
000113F0:81818181
000113F4:81818181
000113F8:81818181
000113FC:81818181
00011400:????????
00011404:????????
00011408:????????
0001140C:????????
00011410:????????
00011414:????????
00011418:????????
0001141C:????????
OutputView
Console stdin/stdout/stderr
Loading assembly language file C:\Users\V V Mohith\Downloads\well5thq.s
Execution starting ...
Location 1094: Invalid opcode encountered
Execution ending, Instruction Count:99 Elapsed Time:00:00:00.0322766
Instructions per second:3067

## PROGRAM:-2(assignment)

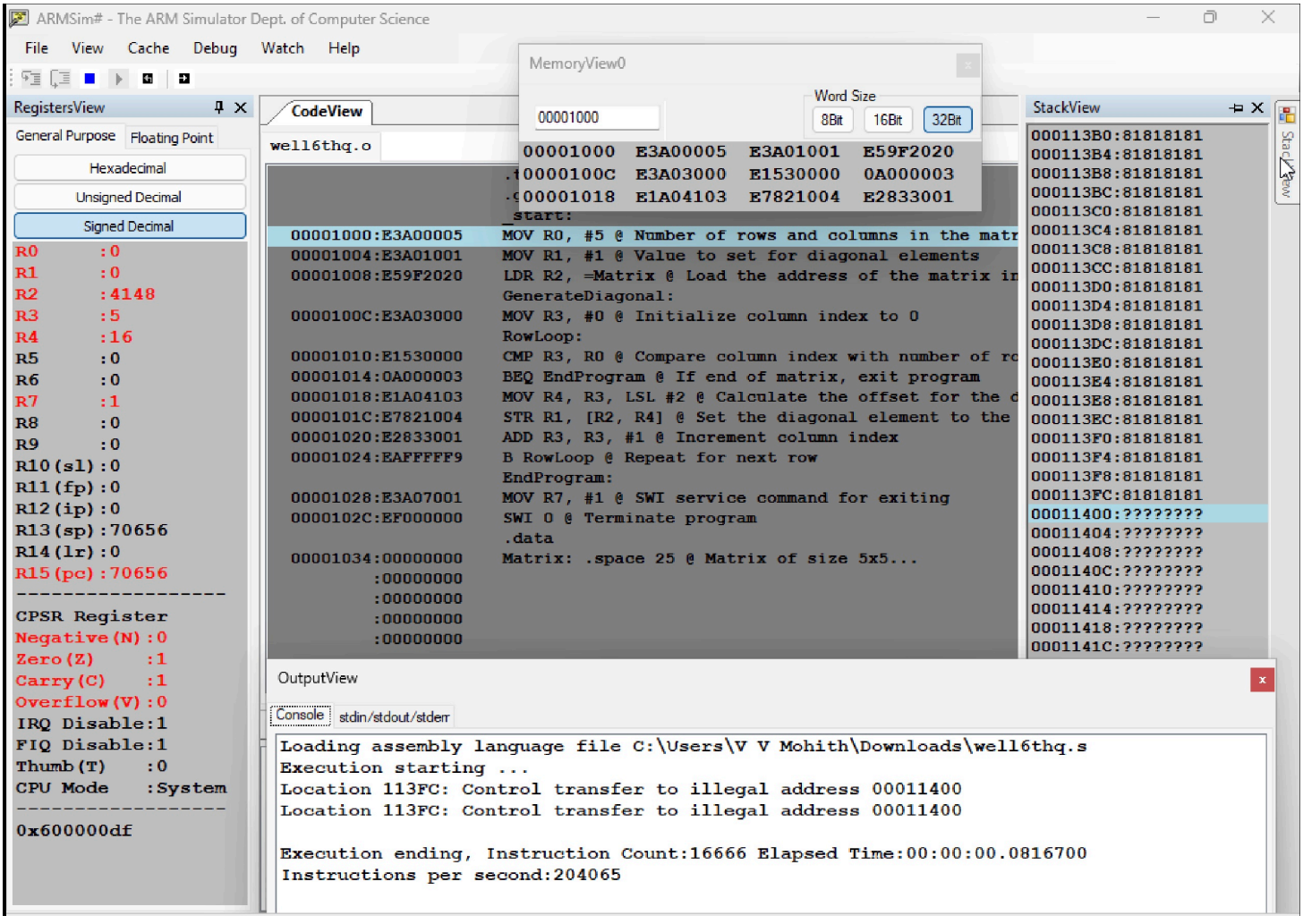
Write a program in ARM7TDMI-ISA to generate a diagonal matrix.
;Note: do not read the matrix elements.

## COMMANDS:-

```
.text

.global _start

_start:

MOV R0, #5

MOV R1, #1

LDR R2, =Matrix

GenerateDiagonal:

MOV R3, #0

RowLoop:

CMP R3, R0

BEQ EndProgram

MOV R4, R3, LSL #2
```

```
STR R1, [R2, R4]

ADD R3, R3, #1

B RowLoop

EndProgram:

MOV R7, #1

SWI 0

.data

Matrix: .space 25
```

**OUTPUTSCREESHOT:-**



**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:V V Mohith

SRN:PES2UG22CS641

Section: K

Date:18-01-2024