Microprocessor and Computer Architecture

UE22CS251B

4th Semester, Academic Year 2023-24

Date:09/02/2024

Name: **Ankith Gowda B S**	SRN: **PES2UG22CS077**	Section: **B**

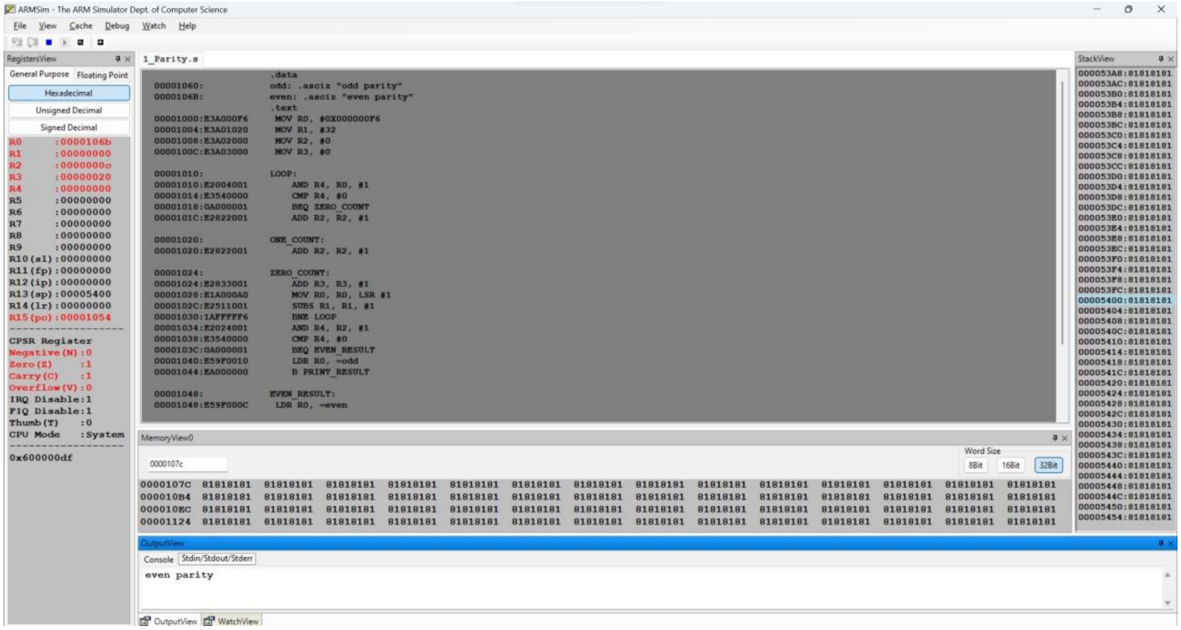
LAB #____3______ Program Number: ____1___

Title of the Program Write an ALP to check whether the given number has odd or even number of 1’s (Even Parity and Odd Parity). ARM Assembly Code

```
.data odd: .asciz "odd

parity" even: .asciz

"even parity"

.text

 MOV R0, #0X00000021

 MOV R1, #32

 MOV R2, #0

MOV R3, #0


LOOP:

   AND R4, R0, #1

   CMP R4, #0

   BEQ ZERO_COUNT

   ADD R2, R2, #1
```

```
ONE_COUNT:
    ADD R2, R2, #1

ZERO_COUNT:
    ADD R3, R3, #1
    MOV R0, R0, LSR #1
    SUBS R1, R1, #1
    BNE LOOP
    AND R4, R2, #1
    CMP R4, #0
    BEQ EVEN_RESULT
    LDR R0, =odd
    B PRINT_RESULT

EVEN_RESULT:
 LDR R0, =odd

PRINT_RESULT:
    SWI 0X02
    B EXIT

EXIT:
 SWI 0X11
.end
```

Output Screen Shots *(Two Screenshots one for odd parity,one for even parity including Register Window,Memory Window and Code Window)*


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
1_Parity.s
StackView
MemoryView0
Word Size
8Bit 16Bit 32Bit
OutputView
Console Stdin/Stdout/Stderr
OutputView WatchView

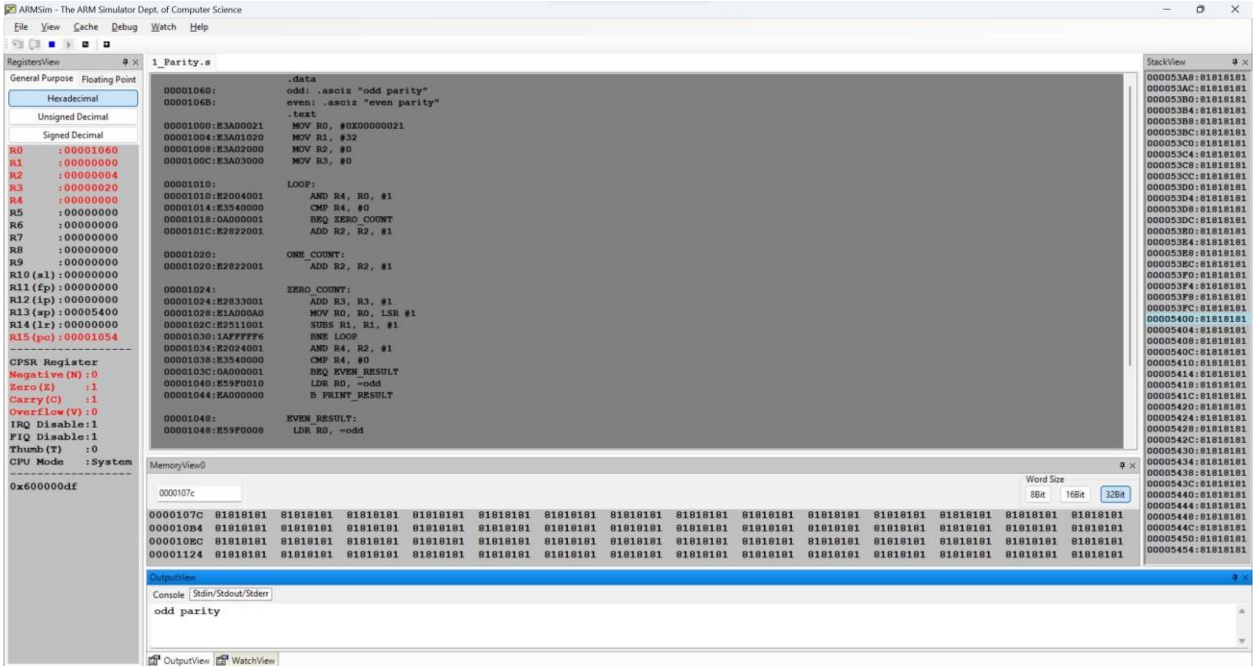
```
R0       :0000106b
R1       :00000000
R2       :0000000c
R3       :00000020
R4       :00000000
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl) :00000000
R11(fp) :00000000
R12(ip) :00000000
R13(sp) :00005400
R14(lr) :00000000
R15(pc) :00001054
------------------
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
```

```
                         .data
00001060:                odd: .asciz "odd parity"
0000106B:                even: .asciz "even parity"
                         .text
00001000:E3A000F6         MOV R0, #0X000000F6
00001004:E3A01020         MOV R1, #32
00001008:E3A02000         MOV R2, #0
0000100C:E3A03000         MOV R3, #0

00001010:                LOOP:
00001010:E2004001            AND R4, R0, #1
00001014:E3540000            CMP R4, #0
00001018:0A000001            BEQ ZERO_COUNT
0000101C:E2822001            ADD R2, R2, #1

00001020:                ONE_COUNT:
00001020:E2822001            ADD R2, R2, #1

00001024:                ZERO_COUNT:
00001024:E2833001            ADD R3, R3, #1
00001028:E1A000A0            MOV R0, R0, LSR #1
0000102C:E2511001            SUBS R1, R1, #1
00001030:1AFFFFF6            BNE LOOP
00001034:E2024001            AND R4, R2, #1
00001038:E3540000            CMP R4, #0
0000103C:0A000001            BEQ EVEN_RESULT
00001040:E59F0010            LDR R0, =odd
00001044:EA000000            B PRINT_RESULT

00001048:                EVEN_RESULT:
00001048:E59F000C         LDR R0, =even
```

```
0000107c
0000107C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010B4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010EC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001124  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
000053A8:81818181
000053AC:81818181
000053B0:81818181
000053B4:81818181
000053B8:81818181
000053BC:81818181
000053C0:81818181
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:81818181
000053FC:81818181
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
00005414:81818181
00005418:81818181
0000541C:81818181
00005420:81818181
00005424:81818181
00005428:81818181
0000542C:81818181
00005430:81818181
00005434:81818181
00005438:81818181
0000543C:81818181
00005440:81818181
00005444:81818181
00005448:81818181
0000544C:81818181
00005450:81818181
00005454:81818181
```

```
even parity
```


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
1_Parity.s
StackView
MemoryView0
Word Size
8Bit 16Bit 32Bit
OutputView
Console Stdin/Stdout/Stderr
OutputView WatchView

```
R0       :00001060
R1       :00000000
R2       :00000004
R3       :00000020
R4       :00000000
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl) :00000000
R11(fp) :00000000
R12(ip) :00000000
R13(sp) :00005400
R14(lr) :00000000
R15(pc) :00001054
------------------
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
```

```
                         .data
00001060:                odd: .asciz "odd parity"
0000106B:                even: .asciz "even parity"
                         .text
00001000:E3A00021         MOV R0, #0X00000021
00001004:E3A01020         MOV R1, #32
00001008:E3A02000         MOV R2, #0
0000100C:E3A03000         MOV R3, #0

00001010:                LOOP:
00001010:E2004001            AND R4, R0, #1
00001014:E3540000            CMP R4, #0
00001018:0A000001            BEQ ZERO_COUNT
0000101C:E2822001            ADD R2, R2, #1

00001020:                ONE_COUNT:
00001020:E2822001            ADD R2, R2, #1

00001024:                ZERO_COUNT:
00001024:E2833001            ADD R3, R3, #1
00001028:E1A000A0            MOV R0, R0, LSR #1
0000102C:E2511001            SUBS R1, R1, #1
00001030:1AFFFFF6            BNE LOOP
00001034:E2024001            AND R4, R2, #1
00001038:E3540000            CMP R4, #0
0000103C:0A000001            BEQ EVEN_RESULT
00001040:E59F0010            LDR R0, =odd
00001044:EA000000            B PRINT_RESULT

00001048:                EVEN_RESULT:
00001048:E59F0008         LDR R0, =odd
```

```
0000107c
0000107C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010B4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010EC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001124  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
000053A8:81818181
000053AC:81818181
000053B0:81818181
000053B4:81818181
000053B8:81818181
000053BC:81818181
000053C0:81818181
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:81818181
000053FC:81818181
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
00005414:81818181
00005418:81818181
0000541C:81818181
00005420:81818181
00005424:81818181
00005428:81818181
0000542C:81818181
00005430:81818181
00005434:81818181
00005438:81818181
0000543C:81818181
00005440:81818181
00005444:81818181
00005448:81818181
0000544C:81818181
00005450:81818181
00005454:81818181
```

```
odd parity
```

LAB #____3

Title of the Program

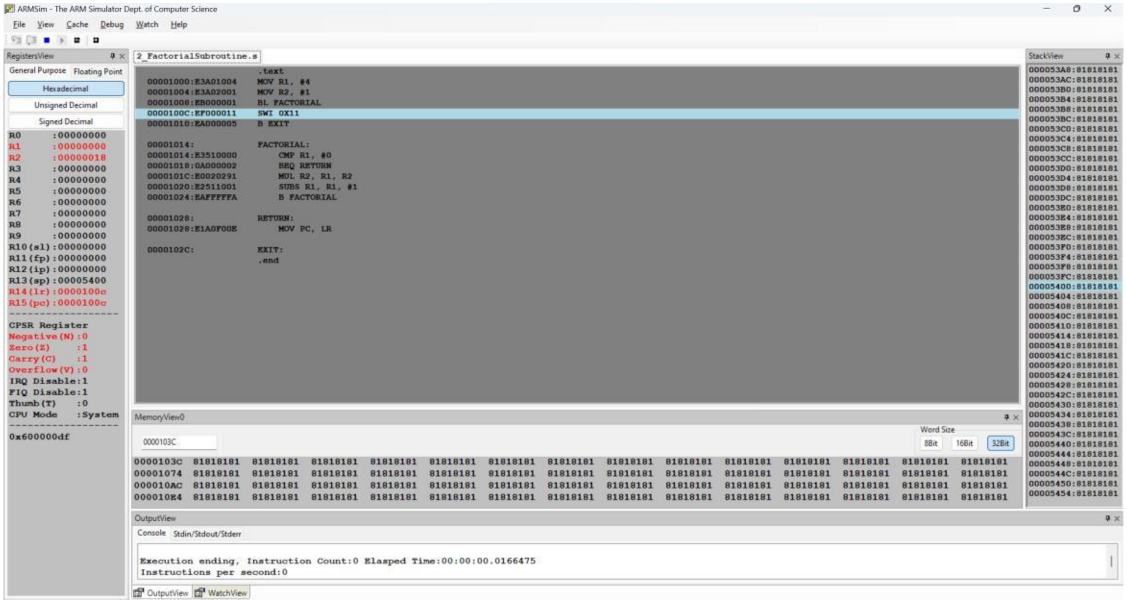
_______ Program Number: ____2___

Write a program to compute the factorial of a number using subroutines

ARM Assembly Code

```
.text

MOV R1, #4

MOV R2, #1

BL FACTORIAL

SWI 0X11

B EXIT


FACTORIAL:

   CMP R1, #0

   BEQ RETURN

   MUL R2, R1, R2

   SUBS R1, R1, #1

   B FACTORIAL


RETURN:

   MOV PC, LR


EXIT:

.end
```

Output Screen Shots *(One Screenshot including Register Window,Memory Window and Code Window)*



```
R0       :00000000
R1       :00000000
R2       :00000018
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)  :00005400
R14(lr)  :0000100c
R15(pc)  :0000100c
------------------
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
```

```
                        .text
00001000:E3A01004       MOV R1, #4
00001004:E3A02001       MOV R2, #1
00001008:EB000001       BL FACTORIAL
0000100C:EF000011       SWI 0X11
00001010:EA000005       B EXIT

00001014:               FACTORIAL:
00001014:E3510000           CMP R1, #0
00001018:0A000002           BEQ RETURN
0000101C:E0020291           MUL R2, R1, R2
00001020:E2511001           SUBS R1, R1, #1
00001024:EAFFFFFA           B FACTORIAL

00001028:               RETURN:
00001028:E1A0F00E           MOV PC, LR

0000102C:               EXIT:
                        .end
```

```
0000103C
0000103C 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001074 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010AC 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010E4 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
```

```
000053A8:81818181
000053AC:81818181
000053B0:81818181
000053B4:81818181
000053B8:81818181
000053BC:81818181
000053C0:81818181
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:81818181
000053FC:81818181
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
00005414:81818181
00005418:81818181
0000541C:81818181
00005420:81818181
00005424:81818181
00005428:81818181
0000542C:81818181
00005430:81818181
00005434:81818181
00005438:81818181
0000543C:81818181
00005440:81818181
00005444:81818181
00005448:81818181
0000544C:81818181
00005450:81818181
00005454:81818181
```

```
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0166475
Instructions per second:0
```

LAB #____3

Title of the Program

_______ Program Number: ____3___

Write an ALP to find the sum of all the digits of a given decimal number

ARM Assembly Code

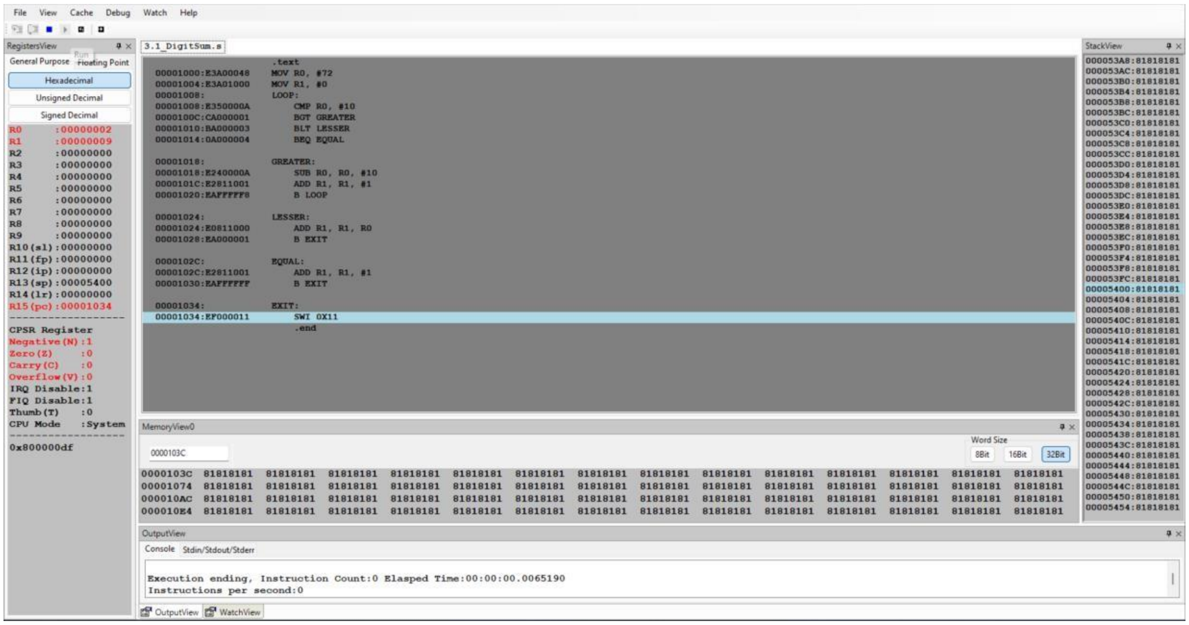
```
.text
MOV R0, #30
MOV R1, #0
LOOP:
   CMP R0, #10
   BGT GREATER
   BLT LESSER
   BEQ EQUAL

GREATER:
   SUB R0, R0, #10
   ADD R1, R1, #1
   B LOOP

LESSER:
   ADD R1, R1, R0
   B EXIT

EQUAL:
   ADD R1, R1, #1
   B EXIT
```

EXIT:

SWI 0X11

.end

Output Screen Shot *(One Screenshot including Register Window,Memory Window and Code Window)*



```
R0       :00000002
R1       :00000009
R2       :00000000
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)  :00005400
R14(lr)  :00000000
R15(pc)  :00001034
------------------
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   :System
------------------
0x800000df
```

```
                        .text
00001000:E3A00048       MOV R0, #72
00001004:E3A01000       MOV R1, #0
00001008:               LOOP:
00001008:E350000A           CMP R0, #10
0000100C:CA000001           BGT GREATER
00001010:BA000003           BLT LESSER
00001014:0A000004           BEQ EQUAL

00001018:               GREATER:
00001018:E240000A           SUB R0, R0, #10
0000101C:E2811001           ADD R1, R1, #1
00001020:EAFFFFF8           B LOOP

00001024:               LESSER:
00001024:E0811000           ADD R1, R1, R0
00001028:EA000001           B EXIT

0000102C:               EQUAL:
0000102C:E2811001           ADD R1, R1, #1
00001030:EAFFFFFF           B EXIT

00001034:               EXIT:
00001034:EF000011           SWI 0X11
                            .end
```

```
0000103C
0000103C 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001074 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010AC 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010E4 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
```

```
000053A8:81818181
000053AC:81818181
000053B0:81818181
000053B4:81818181
000053B8:81818181
000053BC:81818181
000053C0:81818181
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:81818181
000053FC:81818181
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
00005414:81818181
00005418:81818181
0000541C:81818181
00005420:81818181
00005424:81818181
00005428:81818181
0000542C:81818181
00005430:81818181
00005434:81818181
00005438:81818181
0000543C:81818181
00005440:81818181
00005444:81818181
00005448:81818181
0000544C:81818181
00005450:81818181
00005454:81818181
```

```
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0065190
Instructions per second:0
```

LAB #____3

Title of the Program

_______ Program Number: ____4___

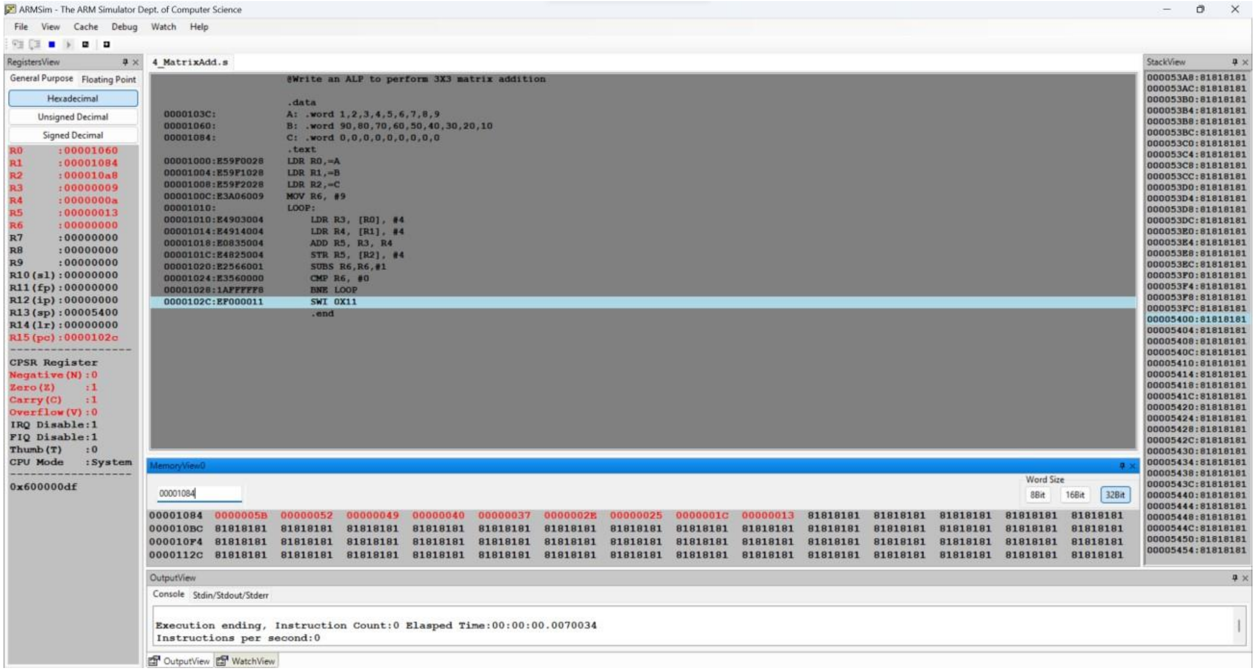
____

Write a program to perform 3X3 matrix addition.

ARM Assembly Code:

```
.data
A: .word 1,2,3,4,5,6,7,8,9
B: .word 90,80,70,60,50,40,30,20,10
C: .word 0,0,0,0,0,0,0,0,0
.text
LDR R0,=A
LDR R1,=B
LDR R2,=C
MOV R6, #9
LOOP:
   LDR R3, [R0], #4
   LDR R4, [R1], #4
   ADD R5, R3, R4
   STR R5, [R2], #4
   SUBS R6,R6,#1
   CMP R6, #0
   BNE LOOP
   SWI 0X11
   .end
```

Output Screen Shots ***(One Screenshot including Register Window,Memory Window and Code Window)***



```
R0        :00001060
R1        :00001084
R2        :000010a8
R3        :00000009
R4        :0000000a
R5        :00000013
R6        :00000000
R7        :00000000
R8        :00000000
R9        :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):0000102c
------------------
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
```

```
                        @Write an ALP to perform 3X3 matrix addition

                        .data
0000103C:               A: .word 1,2,3,4,5,6,7,8,9
00001060:               B: .word 90,80,70,60,50,40,30,20,10
00001084:               C: .word 0,0,0,0,0,0,0,0,0
                        .text
00001000:E59F0028       LDR R0,=A
00001004:E59F1028       LDR R1,=B
00001008:E59F2028       LDR R2,=C
0000100C:E3A06009       MOV R6, #9
00001010:               LOOP:
00001010:E4903004           LDR R3, [R0], #4
00001014:E4914004           LDR R4, [R1], #4
00001018:E0835004           ADD R5, R3, R4
0000101C:E4825004           STR R5, [R2], #4
00001020:E2566001           SUBS R6,R6,#1
00001024:E3560000           CMP R6, #0
00001028:1AFFFFF8           BNE LOOP
0000102C:EF000011           SWI 0X11
                            .end
```

```
00001084
```

```
00001084  0000005B  00000052  00000049  00000040  00000037  0000002E  00000025  0000001C  00000013  81818181  81818181  81818181  81818181  81818181
000010BC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010F4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
0000112C  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
000053A8:81818181
000053AC:81818181
000053B0:81818181
000053B4:81818181
000053B8:81818181
000053BC:81818181
000053C0:81818181
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:81818181
000053FC:81818181
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
00005414:81818181
00005418:81818181
0000541C:81818181
00005420:81818181
00005424:81818181
00005428:81818181
0000542C:81818181
00005430:81818181
00005434:81818181
00005438:81818181
0000543C:81818181
00005440:81818181
00005444:81818181
00005448:81818181
0000544C:81818181
00005450:81818181
00005454:81818181
```

```
Execution ending, Instruction Count:0 Elasped Time:00:00:00.0070034
Instructions per second:0
```

Title of the Program

LAB #____3_______ Program Number: ____5___

Write a program to search for an element in an array using Linear search technique

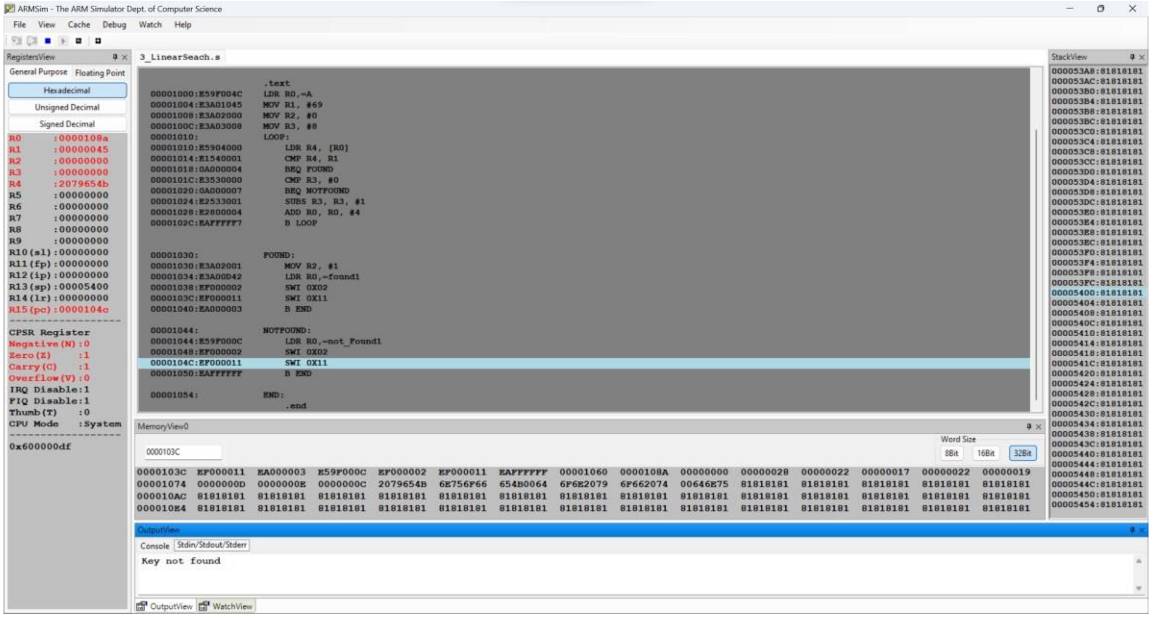
ARM Assembly Code

```
.data

A: .word 40,34,23,34,25,13,14,12

found1: .asciz "Key found"

not_Found1: .asciz "Key not found"


.text

LDR R0,=A

MOV R1, #69

MOV R2, #0

MOV R3, #8 LOOP:

   LDR R4, [R0]

   CMP R4, R1

   BEQ FOUND

   CMP R3, #0

   BEQ NOTFOUND

   SUBS R3, R3, #1

   ADD R0, R0, #4

   B LOOP


FOUND:

   MOV R2, #1
```

```
    LDR R0,=found1
    SWI 0X02
    SWI 0X11
    B END

NOTFOUND:
    LDR R0,=not_Found1
    SWI 0X02
    SWI 0X11
    B END

END:
    .end
```

Output Screen Shot

(One Screenshot including Register Window,Memory Window and Code Window)



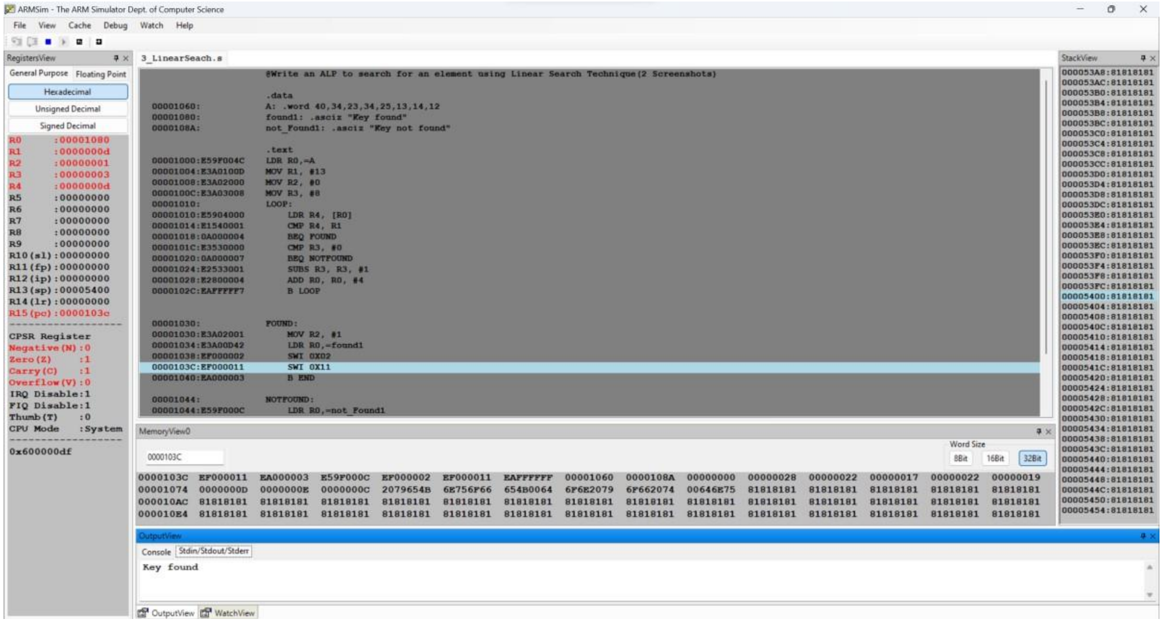
```
R0       :0000108a
R1       :00000045
R2       :00000000
R3       :00000000
R4       :2079654b
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):0000104c
------------------
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
```

```
                        .text
00001000:E59F004C       LDR R0,=A
00001004:E3A01045       MOV R1, #69
00001008:E3A02000       MOV R2, #0
0000100C:E3A03008       MOV R3, #8
00001010:               LOOP:
00001010:E5904000           LDR R4, [R0]
00001014:E1540001           CMP R4, R1
00001018:0A000004           BEQ FOUND
0000101C:E3530000           CMP R3, #0
00001020:0A000007           BEQ NOTFOUND
00001024:E2533001           SUBS R3, R3, #1
00001028:E2800004           ADD R0, R0, #4
0000102C:EAFFFFF7           B LOOP


00001030:               FOUND:
00001030:E3A02001           MOV R2, #1
00001034:E3A00D42           LDR R0,=found1
00001038:EF000002           SWI 0X02
0000103C:EF000011           SWI 0X11
00001040:EA000003           B END

00001044:               NOTFOUND:
00001044:E59F000C           LDR R0,=not_Found1
00001048:EF000002           SWI 0X02
0000104C:EF000011           SWI 0X11
00001050:EAFFFFFF           B END

00001054:               END:
                            .end
```

```
0000103C

0000103C  EF000011  EA000003  E59F000C  EF000002  EF000011  EAFFFFFF  00001060  0000108A  00000000  00000028  00000022  00000017  00000022  00000019
00001074  0000000D  0000000E  0000000C  2079654B  6E756F66  654B0064  6F6E2079  6F662074  00646E75  81818181  81818181  81818181  81818181  81818181
000010AC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010E4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
000053A8:81818181
000053AC:81818181
000053B0:81818181
000053B4:81818181
000053B8:81818181
000053BC:81818181
000053C0:81818181
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:81818181
000053FC:81818181
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
00005414:81818181
00005418:81818181
0000541C:81818181
00005420:81818181
00005424:81818181
00005428:81818181
0000542C:81818181
00005430:81818181
00005434:81818181
00005438:81818181
0000543C:81818181
00005440:81818181
00005444:81818181
00005448:81818181
0000544C:81818181
00005450:81818181
00005454:81818181
```

```
Key not found
```



```
R0       :00001080
R1       :0000000d
R2       :00000001
R3       :00000003
R4       :0000000d
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):0000103c
------------------
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
```

```
                        @Write an ALP to search for an element using Linear Search Technique(2 Screenshots)

                        .data
00001060:               A: .word 40,34,23,34,25,13,14,12
00001080:               found1: .asciz "Key found"
0000108A:               not_Found1: .asciz "Key not found"

                        .text
00001000:E59F004C       LDR R0,=A
00001004:E3A0100D       MOV R1, #13
00001008:E3A02000       MOV R2, #0
0000100C:E3A03008       MOV R3, #8
00001010:               LOOP:
00001010:E5904000           LDR R4, [R0]
00001014:E1540001           CMP R4, R1
00001018:0A000004           BEQ FOUND
0000101C:E3530000           CMP R3, #0
00001020:0A000007           BEQ NOTFOUND
00001024:E2533001           SUBS R3, R3, #1
00001028:E2800004           ADD R0, R0, #4
0000102C:EAFFFFF7           B LOOP


00001030:               FOUND:
00001030:E3A02001           MOV R2, #1
00001034:E3A00D42           LDR R0,=found1
00001038:EF000002           SWI 0X02
0000103C:EF000011           SWI 0X11
00001040:EA000003           B END

00001044:               NOTFOUND:
00001044:E59F000C           LDR R0,=not_Found1
```

```
0000103C

0000103C  EF000011  EA000003  E59F000C  EF000002  EF000011  EAFFFFFF  00001060  0000108A  00000000  00000028  00000022  00000017  00000022  00000019
00001074  0000000D  0000000E  0000000C  2079654B  6E756F66  654B0064  6F6E2079  6F662074  00646E75  81818181  81818181  81818181  81818181  81818181
000010AC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010E4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
000053A8:81818181
000053AC:81818181
000053B0:81818181
000053B4:81818181
000053B8:81818181
000053BC:81818181
000053C0:81818181
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:81818181
000053FC:81818181
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
00005414:81818181
00005418:81818181
0000541C:81818181
00005420:81818181
00005424:81818181
00005428:81818181
0000542C:81818181
00005430:81818181
00005434:81818181
00005438:81818181
0000543C:81818181
00005440:81818181
00005444:81818181
00005448:81818181
0000544C:81818181
00005450:81818181
00005454:81818181
```

```
Key found
```

LAB #____3_______ Assignment Question 1

Title of the Program

i)Write a program to search for an element in an array using binary search technique.

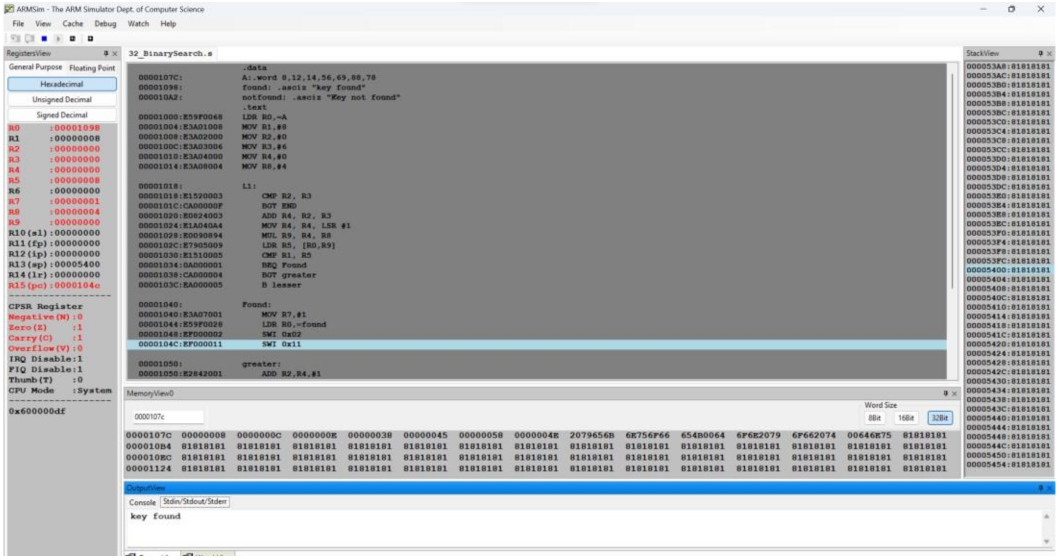
ARM Assembly Code

```
.data
```

```
A:.word 8,12,14,56,69,88,78 found:
.asciz "key found" notfound: .asciz
"Key not found"
.text
LDR R0,=A
MOV R1,#8
MOV R2,#0
MOV R3,#6
MOV R4,#0
MOV R8,#4

L1:
   CMP R2, R3
   BGT END
   ADD R4, R2, R3
   MOV R4, R4, LSR #1
   MUL R9, R4, R8
LDR R5, [R0,R9]
   CMP R1, R5
   BEQ Found
   BGT greater
   B lesser

Found:
   MOV R7,#1
```

```
    LDR R0,=found
    SWI 0x02
    SWI 0x11

greater:
    ADD R2,R4,#1
    B L1

lesser:
    SUB R3,R4,#1
    B L1

END:
    MOV R7,#0
    LDR R0,=notfound
    SWI 0x02
    SWI 0x11
    .END
```

Output Screen Shots ***(Two Screenshots KEY FOUND,KEY NOT FOUND including Register Window,Memory Window and Code Window)***


ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
32_BinarySearch.s
MemoryView0
Word Size
8Bit 16Bit 32Bit
StackView
OutputView
Console Stdin/Stdout/Stderr

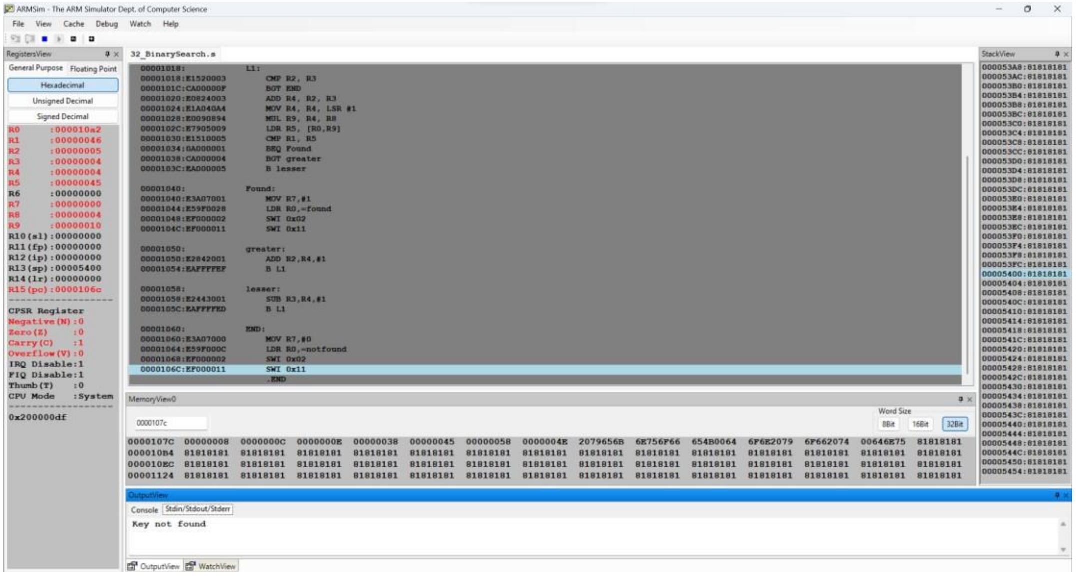
```
R0        :00001098
R1        :00000008
R2        :00000000
R3        :00000000
R4        :00000000
R5        :00000008
R6        :00000000
R7        :00000001
R8        :00000004
R9        :00000000
R10(sl)   :00000000
R11(fp)   :00000000
R12(ip)   :00000000
R13(sp)   :00005400
R14(lr)   :00000000
R15(pc)   :0000104c
------------------
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
```

```
                        .data
0000107C:               A:.word 8,12,14,56,69,88,78
00001098:               found: .asciz "key found"
000010A2:               notfound: .asciz "Key not found"
                        .text
00001000:E59F0068       LDR R0,=A
00001004:E3A01008       MOV R1,#8
00001008:E3A02000       MOV R2,#0
0000100C:E3A03006       MOV R3,#6
00001010:E3A04000       MOV R4,#0
00001014:E3A08004       MOV R8,#4

00001018:               L1:
00001018:E1520003           CMP R2, R3
0000101C:CA00000F           BGT END
00001020:E0824003           ADD R4, R2, R3
00001024:E1A040A4           MOV R4, R4, LSR #1
00001028:E0090894           MUL R9, R4, R8
0000102C:E7905009           LDR R5, [R0,R9]
00001030:E1510005           CMP R1, R5
00001034:0A000001           BEQ Found
00001038:CA000004           BGT greater
0000103C:EA000005           B lesser

00001040:               Found:
00001040:E3A07001           MOV R7,#1
00001044:E59F0028           LDR R0,=found
00001048:EF000002           SWI 0x02
0000104C:EF000011           SWI 0x11

00001050:               greater:
00001050:E2842001           ADD R2,R4,#1
```

```
0000107c

0000107C  00000008  0000000C  0000000E  00000038  00000045  00000058  0000004E  2079656B  6E756F66  654B0064  6F6E2079  6F662074  00646E75  81818181
000010B4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010EC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001124  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
000053A8:81818181
000053AC:81818181
000053B0:81818181
000053B4:81818181
000053B8:81818181
000053BC:81818181
000053C0:81818181
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:81818181
000053FC:81818181
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
00005414:81818181
00005418:81818181
0000541C:81818181
00005420:81818181
00005424:81818181
00005428:81818181
0000542C:81818181
00005430:81818181
00005434:81818181
00005438:81818181
0000543C:81818181
00005440:81818181
00005444:81818181
00005448:81818181
0000544C:81818181
00005450:81818181
00005454:81818181
```

```
key found
```



```
R0        :000010a2
R1        :00000046
R2        :00000005
R3        :00000004
R4        :00000004
R5        :00000045
R6        :00000000
R7        :00000000
R8        :00000004
R9        :00000010
R10(sl)   :00000000
R11(fp)   :00000000
R12(ip)   :00000000
R13(sp)   :00005400
R14(lr)   :00000000
R15(pc)   :0000106c
------------------
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
```

```
00001018:               L1:
00001018:E1520003           CMP R2, R3
0000101C:CA00000F           BGT END
00001020:E0824003           ADD R4, R2, R3
00001024:E1A040A4           MOV R4, R4, LSR #1
00001028:E0090894           MUL R9, R4, R8
0000102C:E7905009           LDR R5, [R0,R9]
00001030:E1510005           CMP R1, R5
00001034:0A000001           BEQ Found
00001038:CA000004           BGT greater
0000103C:EA000005           B lesser

00001040:               Found:
00001040:E3A07001           MOV R7,#1
00001044:E59F0028           LDR R0,=found
00001048:EF000002           SWI 0x02
0000104C:EF000011           SWI 0x11

00001050:               greater:
00001050:E2842001           ADD R2,R4,#1
00001054:EAFFFFEF           B L1

00001058:               lesser:
00001058:E2443001           SUB R3,R4,#1
0000105C:EAFFFFED           B L1

00001060:               END:
00001060:E3A07000           MOV R7,#0
00001064:E59F000C           LDR R0,=notfound
00001068:EF000002           SWI 0x02
0000106C:EF000011           SWI 0x11
                            .END
```

```
0000107c

0000107C  00000008  0000000C  0000000E  00000038  00000045  00000058  0000004E  2079656B  6E756F66  654B0064  6F6E2079  6F662074  00646E75  81818181
000010B4  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
000010EC  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
00001124  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181  81818181
```

```
000053A8:81818181
000053AC:81818181
000053B0:81818181
000053B4:81818181
000053B8:81818181
000053BC:81818181
000053C0:81818181
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:81818181
000053FC:81818181
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
00005414:81818181
00005418:81818181
0000541C:81818181
00005420:81818181
00005424:81818181
00005428:81818181
0000542C:81818181
00005430:81818181
00005434:81818181
00005438:81818181
0000543C:81818181
00005440:81818181
00005444:81818181
00005448:81818181
0000544C:81818181
00005450:81818181
00005454:81818181
```

```
Key not found
```

Title of the Program

ii)Write a program to find the sum of N data items at alternate [odd or even positions] locations in the memory. Store the result in the memory location. ARM Assembly Code

```
.data
A: .word
54,26,12,5,6,11,50 odd:
.word 0 even: .word 0

.text
LDR R0,=A
MOV R1, #7
MOV R2, #0
MOV R3, #0
LDR R5,=odd
LDR R6,=even

ODD:
LDR R4, [R0], #4
SUB R1, R1, #1
ADD R2, R2, R4
CMP R1, #0
BEQ END
B EVEN

EVEN:
```

LDR R4, [R0], #4

SUB R1, R1, #1

ADD R3, R3, R4

CMP R1, #0

BEQ END

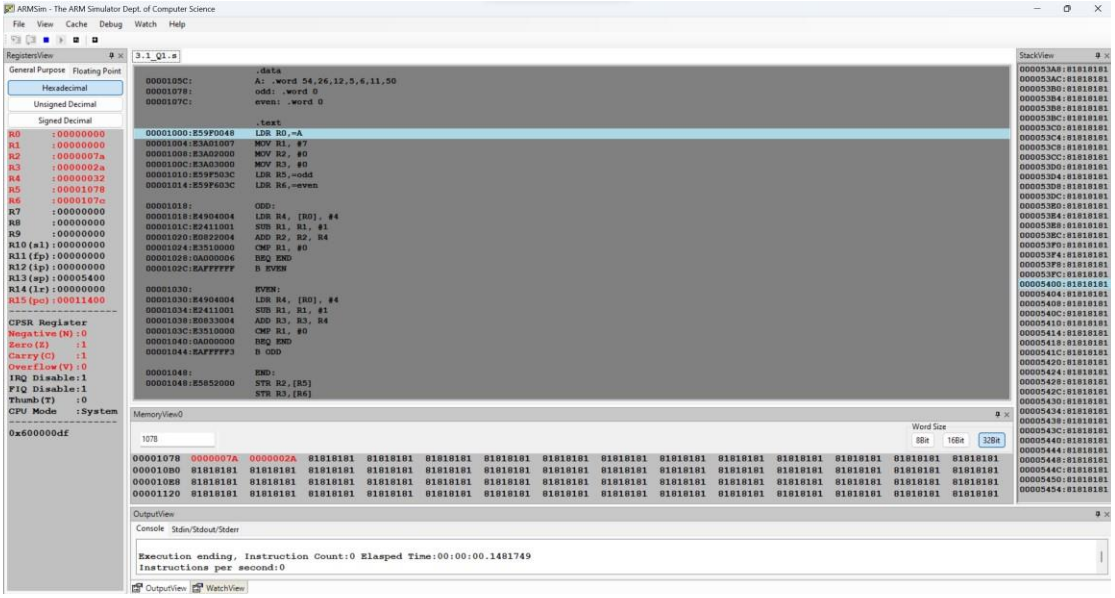
B ODD

END:

STR R2,[R5]

STR R3,[R6]

Output Screen Shots *(One Screenshot including Register Window,Memory Window and Code Window)*



ARMSim - The ARM Simulator Dept. of Computer Science
File View Cache Debug Watch Help
RegistersView
General Purpose Floating Point
Hexadecimal
Unsigned Decimal
Signed Decimal
3.1_Q1.s
MemoryView0
Word Size 8Bit 16Bit 32Bit
StackView
OutputView
Console Stdin/Stdout/Stderr
OutputView WatchView

```
R0      :00000000
R1      :00000000
R2      :0000007a
R3      :0000002a
R4      :00000032
R5      :00001078
R6      :0000107c
R7      :00000000
R8      :00000000
R9      :00000000
R10(sl) :00000000
R11(fp) :00000000
R12(ip) :00000000
R13(sp) :00005400
R14(lr) :00000000
R15(pc) :00011400
------------------
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
```

```
                        .data
0000105C:               A: .word 54,26,12,5,6,11,50
00001078:               odd: .word 0
0000107C:               even: .word 0

                        .text
00001000:E59F0048       LDR R0,=A
00001004:E3A01007       MOV R1, #7
00001008:E3A02000       MOV R2, #0
0000100C:E3A03000       MOV R3, #0
00001010:E59F503C       LDR R5,=odd
00001014:E59F603C       LDR R6,=even

00001018:               ODD:
00001018:E4904004       LDR R4, [R0], #4
0000101C:E2411001       SUB R1, R1, #1
00001020:E0822004       ADD R2, R2, R4
00001024:E3510000       CMP R1, #0
00001028:0A000006       BEQ END
0000102C:EAFFFFFF       B EVEN

00001030:               EVEN:
00001030:E4904004       LDR R4, [R0], #4
00001034:E2411001       SUB R1, R1, #1
00001038:E0833004       ADD R3, R3, R4
0000103C:E3510000       CMP R1, #0
00001040:0A000000       BEQ END
00001044:EAFFFFF3       B ODD

00001048:               END:
00001048:E5852000       STR R2,[R5]
                        STR R3,[R6]
```

```
1078
00001078 0000007A 0000002A 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010B0 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
000010E8 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
00001120 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181 81818181
```

```
000053A8:81818181
000053AC:81818181
000053B0:81818181
000053B4:81818181
000053B8:81818181
000053BC:81818181
000053C0:81818181
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:81818181
000053FC:81818181
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
00005414:81818181
00005418:81818181
0000541C:81818181
00005420:81818181
00005424:81818181
00005428:81818181
0000542C:81818181
00005430:81818181
00005434:81818181
00005438:81818181
0000543C:81818181
00005440:81818181
00005444:81818181
00005448:81818181
0000544C:81818181
00005450:81818181
00005454:81818181
```

```
Execution ending, Instruction Count:0 Elasped Time:00:00:00.1481749
Instructions per second:0
```

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.

Name: Ankith Gowda B S
SRN: PES2UG22CS077
Section: B
Date: 09/02/2024