PES
UNIVERSITY

Department of Computer Science & Engineering
Microprocessor & Computer Architecture Lab

Lab 2 Programs

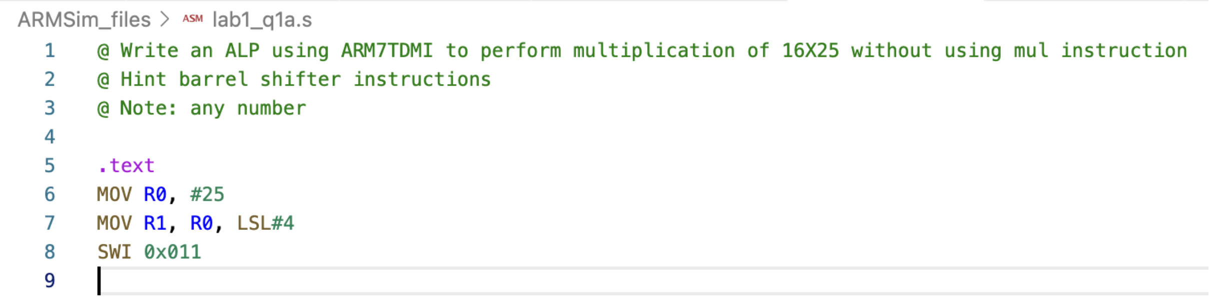
UE23CS251B

Name of Student: Pranav Hemanth	SRN: PES1UG23CS433

1 Write an ALP using ARM7TDMI to perform the multiplication of 16X25without using MUL instructions.

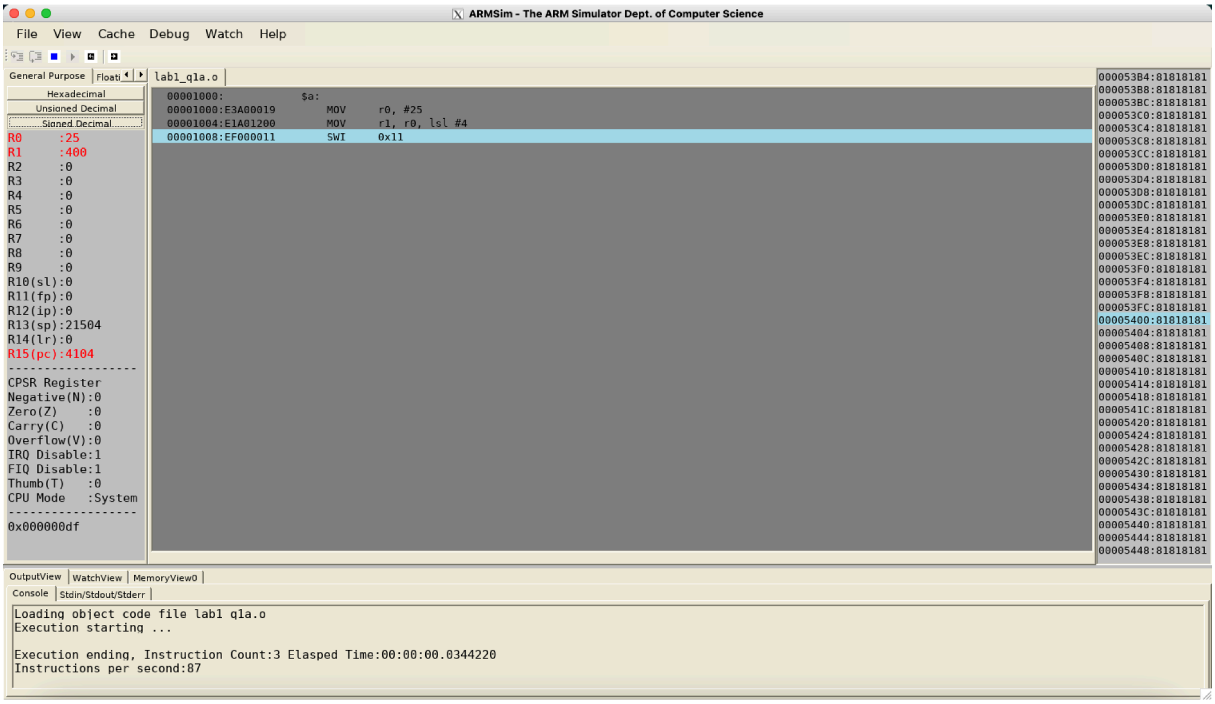
(Hint: barrel shifter instructions.)

(Note : Any number can be considered as multiplier)



ARMSim_files > ASM lab1_q1a.s

```
@ Write an ALP using ARM7TDMI to perform multiplication of 16X25 without using mul instruction
@ Hint barrel shifter instructions
@ Note: any number

.text
MOV R0, #25
MOV R1, R0, LSL#4
SWI 0x011

```

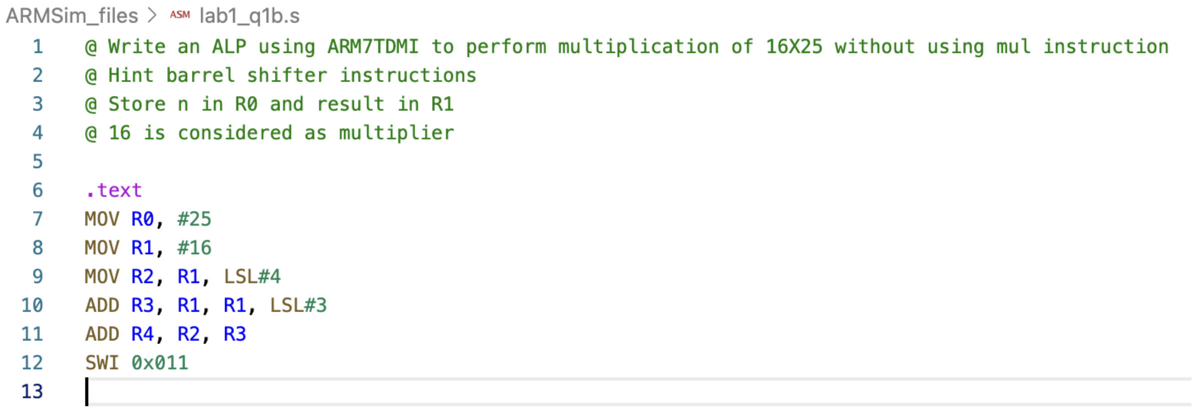


```
R0       :25
R1       :400
R2       :0
R3       :0
R4       :0
R5       :0
R6       :0
R7       :0
R8       :0
R9       :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4104
------------------
CPSR Register
Negative(N):0
Zero(Z)     :0
Carry(C)    :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
------------------
0x000000df
```

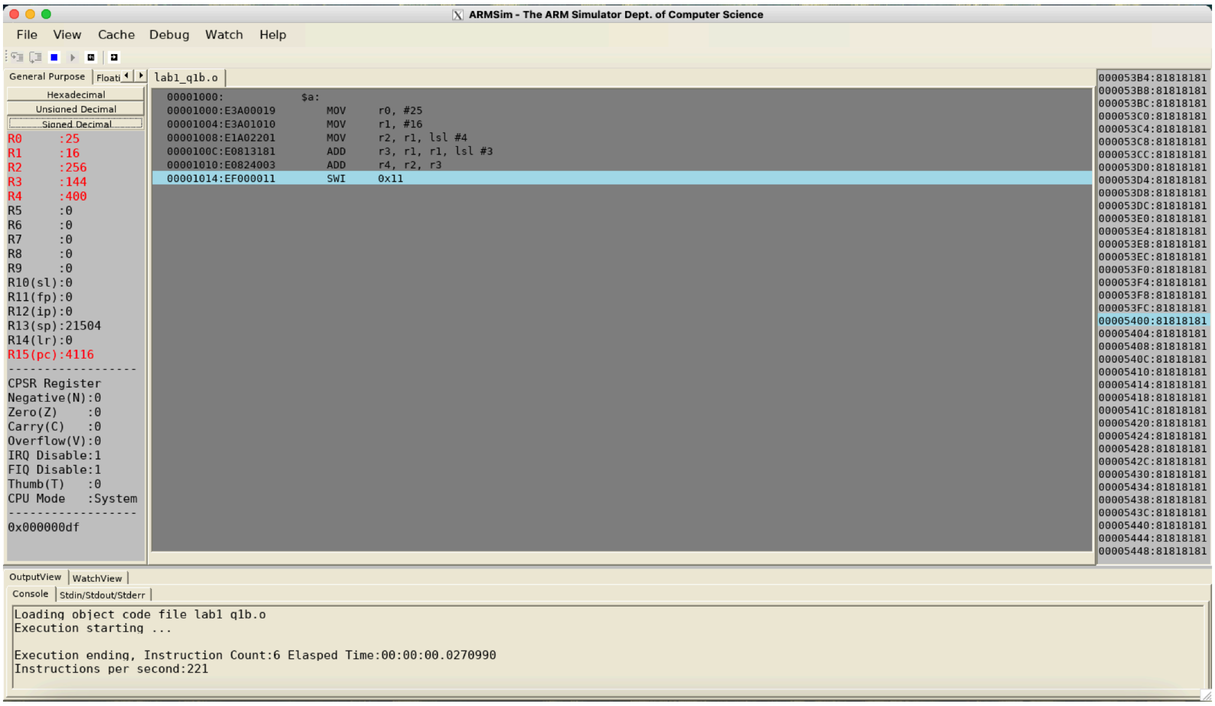
```
00001000:            $a:
00001000:E3A00019        MOV      r0, #25
00001004:E1A01200        MOV      r1, r0, lsl #4
00001008:EF000011        SWI      0x11
```

```
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
```

```
Loading object code file lab1 q1a.o
Execution starting ...

Execution ending, Instruction Count:3 Elasped Time:00:00:00.0344220
Instructions per second:87
```



```
@ Write an ALP using ARM7TDMI to perform multiplication of 16X25 without using mul instruction
@ Hint barrel shifter instructions
@ Store n in R0 and result in R1
@ 16 is considered as multiplier

.text
MOV R0, #25
MOV R1, #16
MOV R2, R1, LSL#4
ADD R3, R1, R1, LSL#3
ADD R4, R2, R3
SWI 0x011

```



```
R0        :25
R1        :16
R2        :256
R3        :144
R4        :400
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):4116
------------------
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
```

```
00001000:             $a:
00001000:E3A00019        MOV     r0, #25
00001004:E3A01010        MOV     r1, #16
00001008:E1A02201        MOV     r2, r1, lsl #4
0000100C:E0813181        ADD     r3, r1, r1, lsl #3
00001010:E0824003        ADD     r4, r2, r3
00001014:EF000011        SWI     0x11
```

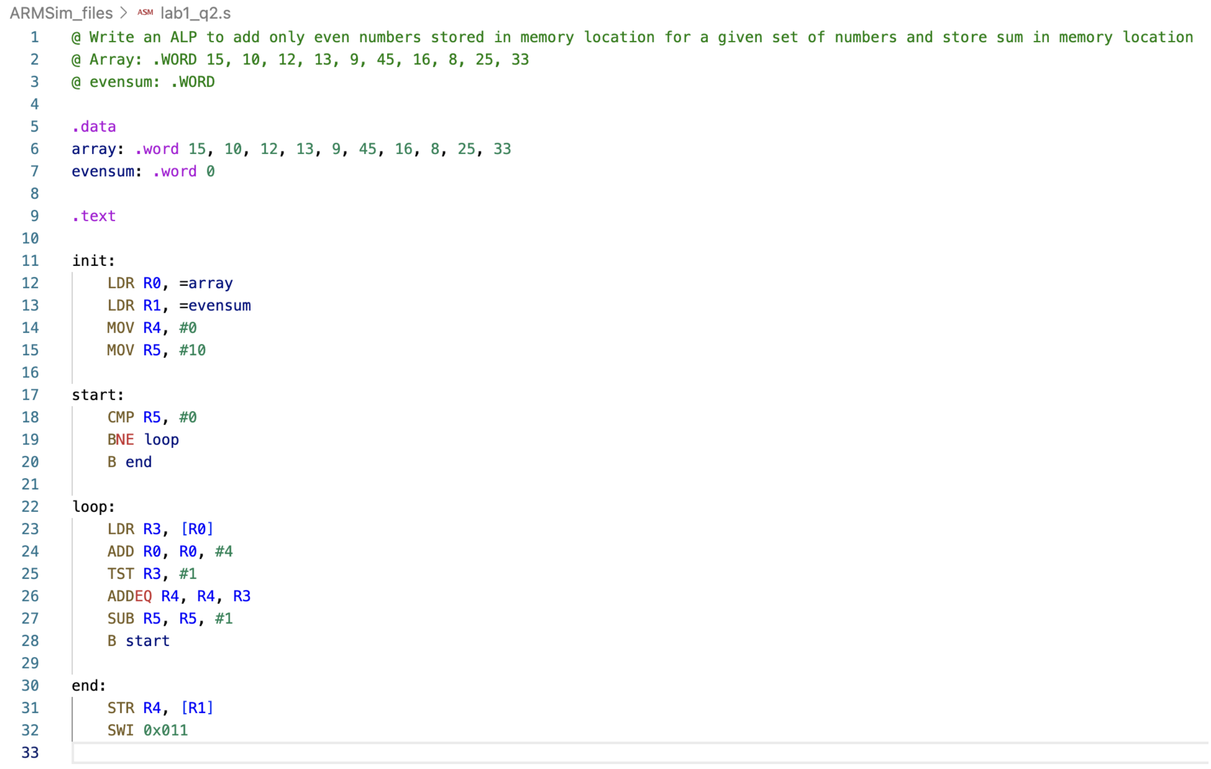
```
000053B4:81818181
000053B8:81818181
000053BC:81818181
000053C0:81818181
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
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
```

```
Loading object code file lab1 q1b.o
Execution starting ...

Execution ending, Instruction Count:6 Elasped Time:00:00:00.0270990
Instructions per second:221
```

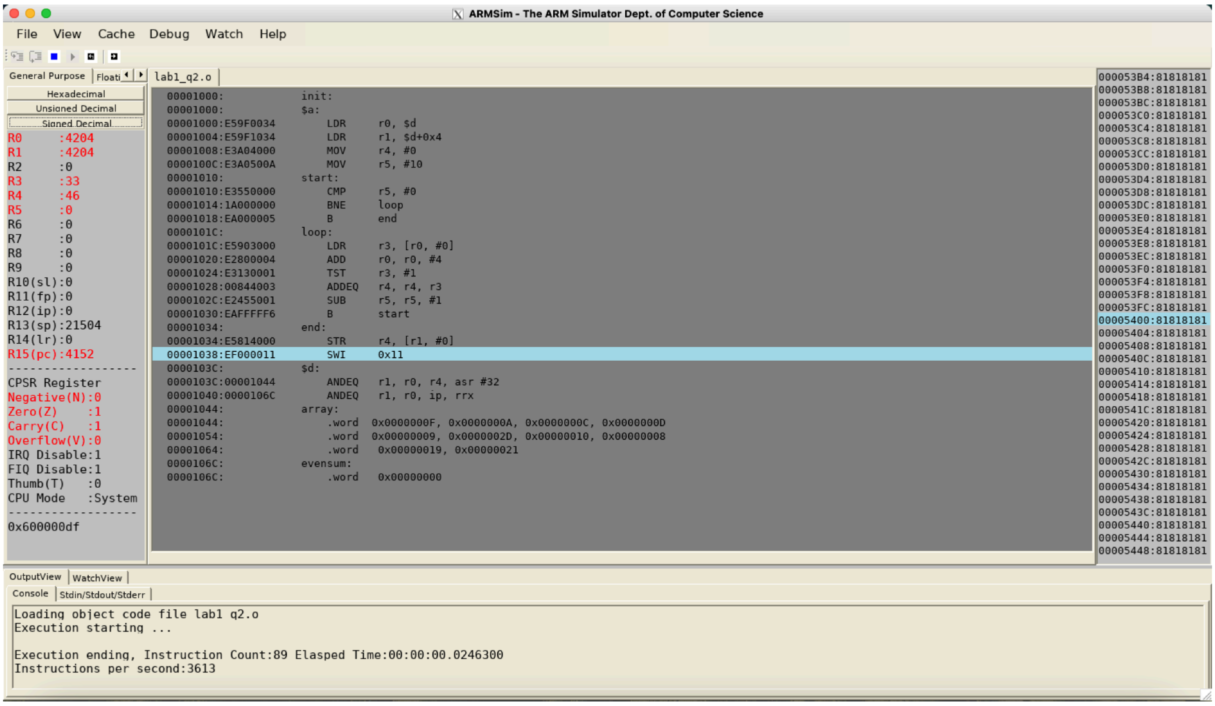
2 Write an ALP using ARM7TDMI to add only even numbers stored in memory location for a given set of numbers and store the sum in the memory location.

Array:. WORD 15,10,12,13,9,45,16,8,25,33

evensum:. WORD



```
@ Write an ALP to add only even numbers stored in memory location for a given set of numbers and store sum in memory location
@ Array: .WORD 15, 10, 12, 13, 9, 45, 16, 8, 25, 33
@ evensum: .WORD

.data
array: .word 15, 10, 12, 13, 9, 45, 16, 8, 25, 33
evensum: .word 0

.text

init:
    LDR R0, =array
    LDR R1, =evensum
    MOV R4, #0
    MOV R5, #10

start:
    CMP R5, #0
    BNE loop
    B end

loop:
    LDR R3, [R0]
    ADD R0, R0, #4
    TST R3, #1
    ADDEQ R4, R4, R3
    SUB R5, R5, #1
    B start

end:
    STR R4, [R1]
    SWI 0x011

```



```
R0      :4204
R1      :4204
R2      :0
R3      :33
R4      :46
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):4152
------------------
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
```

```
00001000:               init:
00001000:               $a:
00001000:E59F0034           LDR     r0, $d
00001004:E59F1034           LDR     r1, $d+0x4
00001008:E3A04000           MOV     r4, #0
0000100C:E3A0500A           MOV     r5, #10
00001010:               start:
00001010:E3550000           CMP     r5, #0
00001014:1A000000           BNE     loop
00001018:EA000005           B       end
0000101C:               loop:
0000101C:E5903000           LDR     r3, [r0, #0]
00001020:E2800004           ADD     r0, r0, #4
00001024:E3130001           TST     r3, #1
00001028:00844003           ADDEQ   r4, r4, r3
0000102C:E2455001           SUB     r5, r5, #1
00001030:EAFFFFF6           B       start
00001034:               end:
00001034:E5814000           STR     r4, [r1, #0]
00001038:EF000011           SWI     0x11
0000103C:               $d:
0000103C:00001044           ANDEQ   r1, r0, r4, asr #32
00001040:0000106C           ANDEQ   r1, r0, ip, rrx
00001044:               array:
00001044:                   .word  0x0000000F, 0x0000000A, 0x0000000C, 0x0000000D
00001054:                   .word  0x00000009, 0x0000002D, 0x00000010, 0x00000008
00001064:                   .word   0x00000019, 0x00000021
0000106C:               evensum:
0000106C:                   .word   0x00000000
```

```
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
```

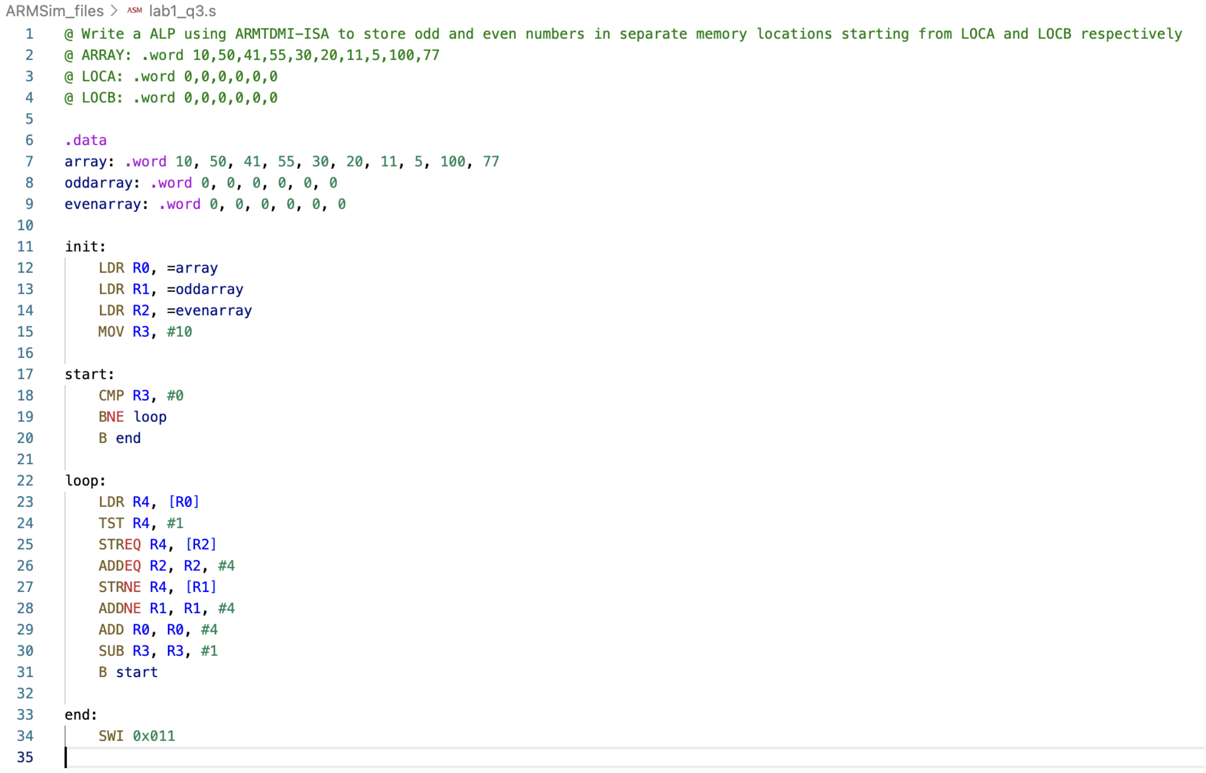
```
Loading object code file lab1 q2.o
Execution starting ...

Execution ending, Instruction Count:89 Elasped Time:00:00:00.0246300
Instructions per second:3613
```

3 Write a ALP using ARMTDMI-ISA to store odd and even numbers in separate memory locations starting from LOCA and LOCB respectively

ARRAY: .word 10,50,41,55,30,20,11,5,100,77

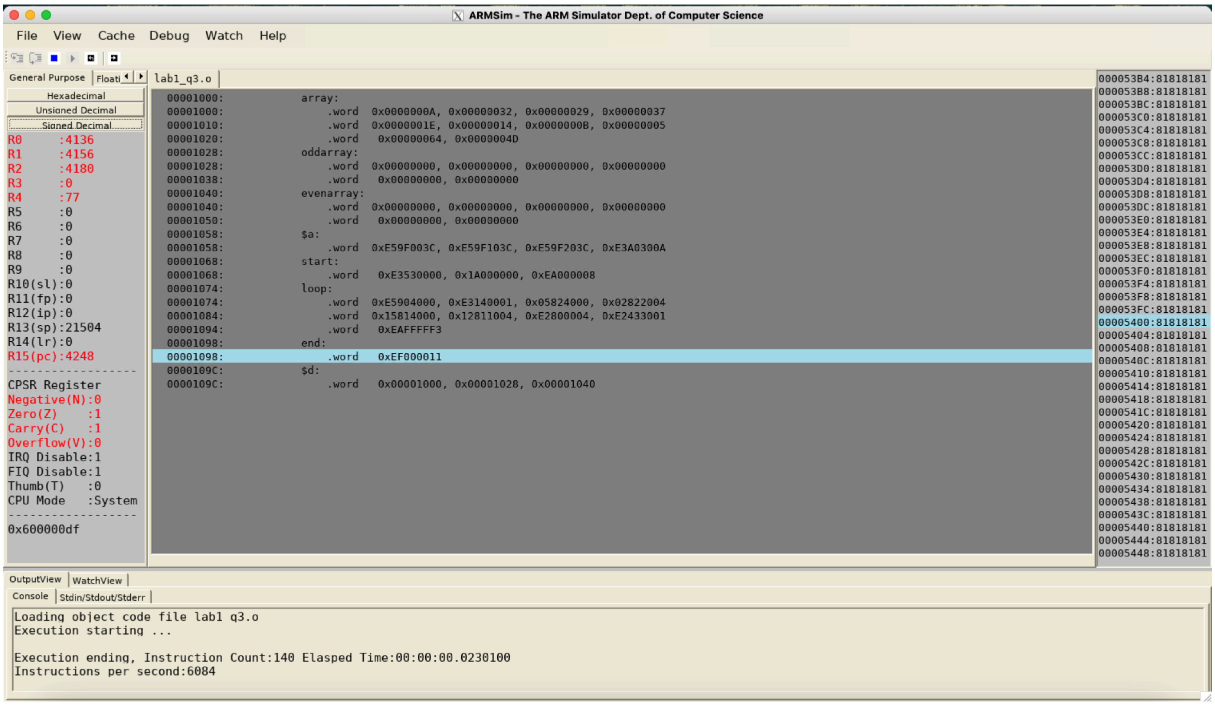
LOCA: .word 0,0,0,0,0,0

LOCB: .word 0,0,0,0,0,0



ARMSim_files > lab1_q3.s

```
@ Write a ALP using ARMTDMI-ISA to store odd and even numbers in separate memory locations starting from LOCA and LOCB respectively
@ ARRAY: .word 10,50,41,55,30,20,11,5,100,77
@ LOCA: .word 0,0,0,0,0,0
@ LOCB: .word 0,0,0,0,0,0

.data
array: .word 10, 50, 41, 55, 30, 20, 11, 5, 100, 77
oddarray: .word 0, 0, 0, 0, 0, 0
evenarray: .word 0, 0, 0, 0, 0, 0

init:
    LDR R0, =array
    LDR R1, =oddarray
    LDR R2, =evenarray
    MOV R3, #10

start:
    CMP R3, #0
    BNE loop
    B end

loop:
    LDR R4, [R0]
    TST R4, #1
    STREQ R4, [R2]
    ADDEQ R2, R2, #4
    STRNE R4, [R1]
    ADDNE R1, R1, #4
    ADD R0, R0, #4
    SUB R3, R3, #1
    B start

end:
    SWI 0x011

```



```
R0       :4136
R1       :4156
R2       :4180
R3       :0
R4       :77
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):4248
------------------
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
```

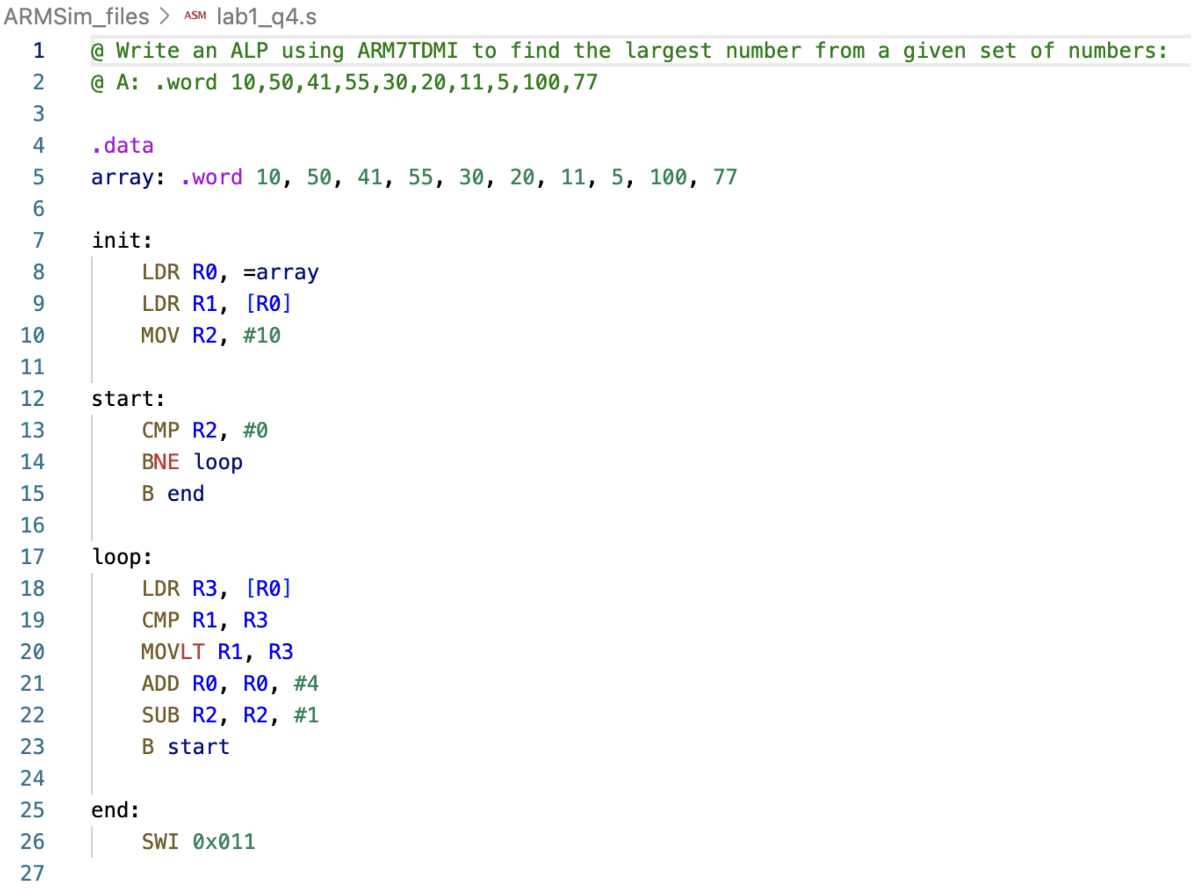
```
00001000:            array:
00001000:                .word  0x0000000A, 0x00000032, 0x00000029, 0x00000037
00001010:                .word  0x0000001E, 0x00000014, 0x0000000B, 0x00000005
00001020:                .word   0x00000064, 0x0000004D
00001028:            oddarray:
00001028:                .word  0x00000000, 0x00000000, 0x00000000, 0x00000000
00001038:                .word   0x00000000, 0x00000000
00001040:            evenarray:
00001040:                .word  0x00000000, 0x00000000, 0x00000000, 0x00000000
00001050:                .word   0x00000000, 0x00000000
00001058:            $a:
00001058:                .word  0xE59F003C, 0xE59F103C, 0xE59F203C, 0xE3A0300A
00001068:            start:
00001068:                .word   0xE3530000, 0x1A000000, 0xEA000008
00001074:            loop:
00001074:                .word  0xE5904000, 0xE3140001, 0x05824000, 0x02822004
00001084:                .word  0x15814000, 0x12811004, 0xE2800004, 0xE2433001
00001094:                .word   0xEAFFFFF3
00001098:            end:
00001098:                .word   0xEF000011
0000109C:            $d:
0000109C:                .word   0x00001000, 0x00001028, 0x00001040
```

```
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
```

```
Loading object code file lab1_q3.o
Execution starting ...

Execution ending, Instruction Count:140 Elasped Time:00:00:00.0230100
Instructions per second:6084
```

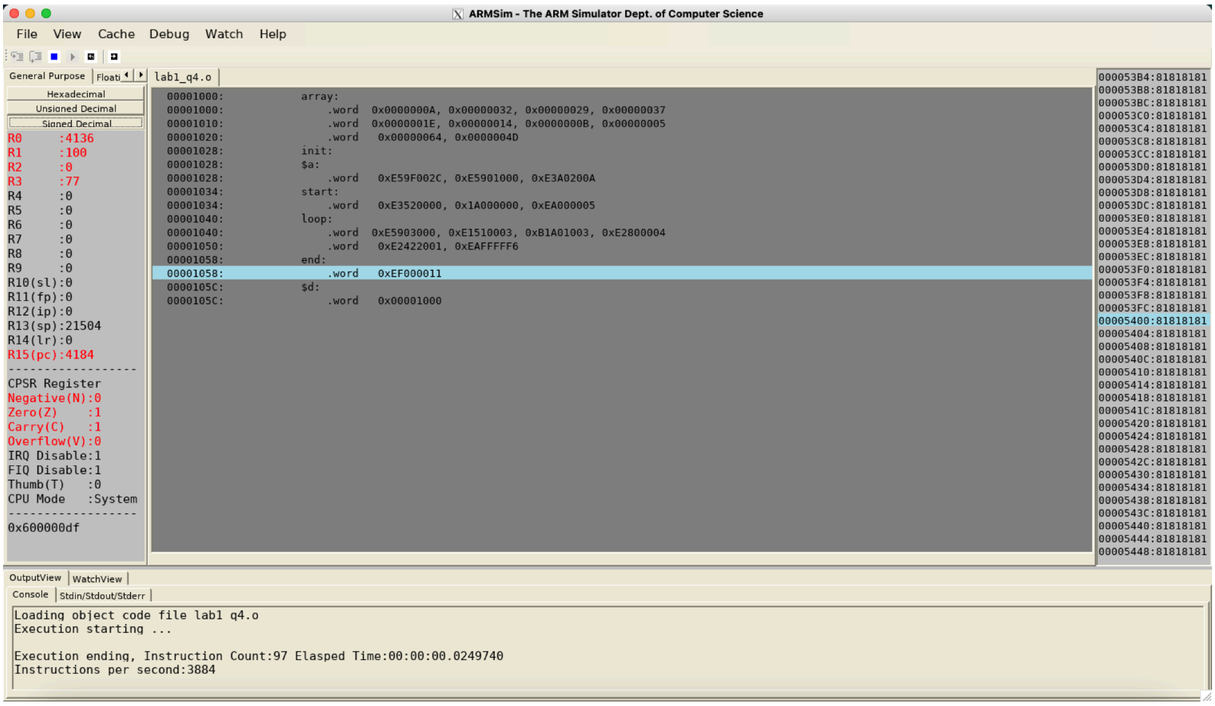
4 Write an ALP using ARM7TDMI to find the largest number from a given set of numbers:

A: .word 10,50,41,55,30,20,11,5,100,77



ARMSim_files > ASM lab1_q4.s

```
@ Write an ALP using ARM7TDMI to find the largest number from a given set of numbers:
@ A: .word 10,50,41,55,30,20,11,5,100,77

.data
array: .word 10, 50, 41, 55, 30, 20, 11, 5, 100, 77

init:
    LDR R0, =array
    LDR R1, [R0]
    MOV R2, #10

start:
    CMP R2, #0
    BNE loop
    B end

loop:
    LDR R3, [R0]
    CMP R1, R3
    MOVLT R1, R3
    ADD R0, R0, #4
    SUB R2, R2, #1
    B start

end:
    SWI 0x011
```



```
R0       :4136
R1       :100
R2       :0
R3       :77
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):4184
------------------
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
```

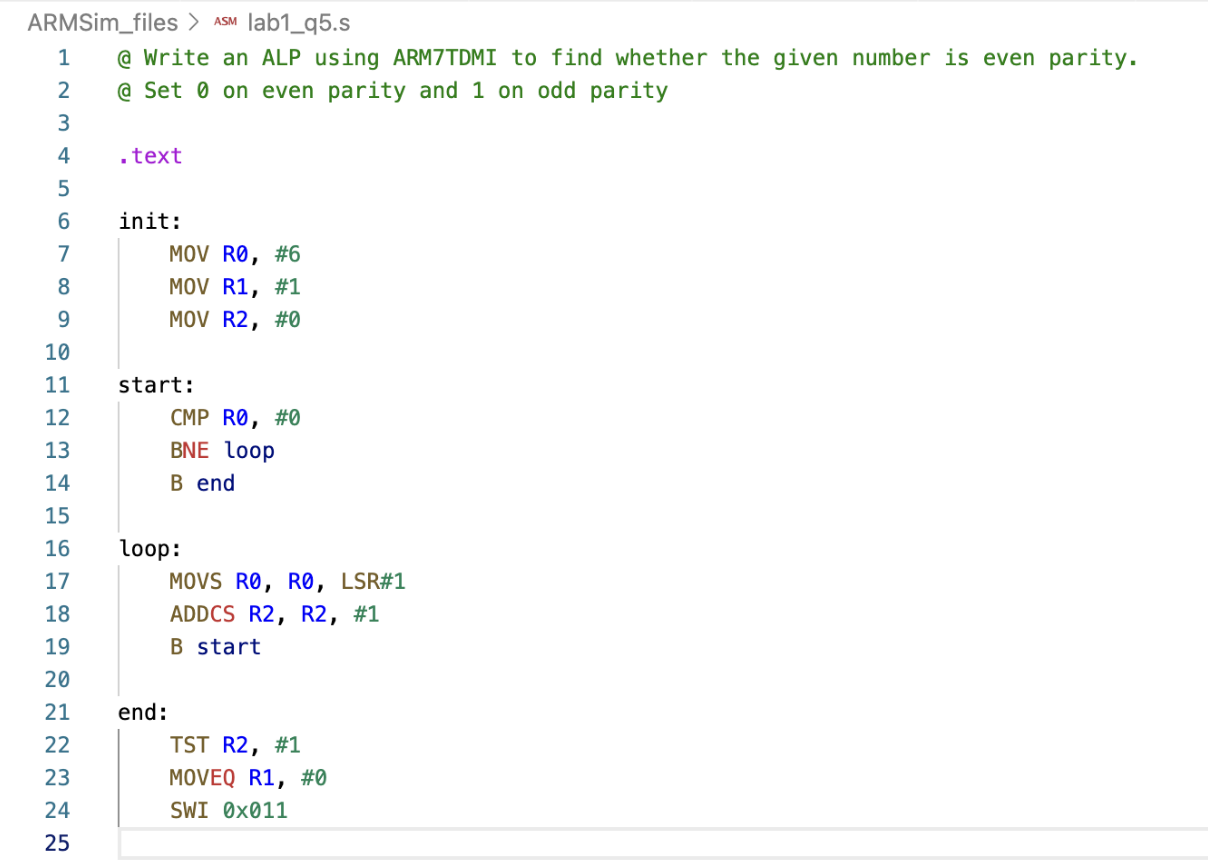
```
00001000:           array:
00001000:               .word  0x0000000A, 0x00000032, 0x00000029, 0x00000037
00001010:               .word  0x0000001E, 0x00000014, 0x0000000B, 0x00000005
00001020:               .word   0x00000064, 0x0000004D
00001028:           init:
00001028:           $a:
00001028:               .word   0xE59F002C, 0xE5901000, 0xE3A0200A
00001034:           start:
00001034:               .word   0xE3520000, 0x1A000000, 0xEA000005
00001040:           loop:
00001040:               .word  0xE5903000, 0xE1510003, 0xB1A01003, 0xE2800004
00001050:               .word   0xE2422001, 0xEAFFFFF6
00001058:           end:
00001058:               .word   0xEF000011
0000105C:           $d:
0000105C:               .word   0x00001000
```

```
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
```

```
Loading object code file lab1 q4.o
Execution starting ...

Execution ending, Instruction Count:97 Elasped Time:00:00:00.0249740
Instructions per second:3884
```

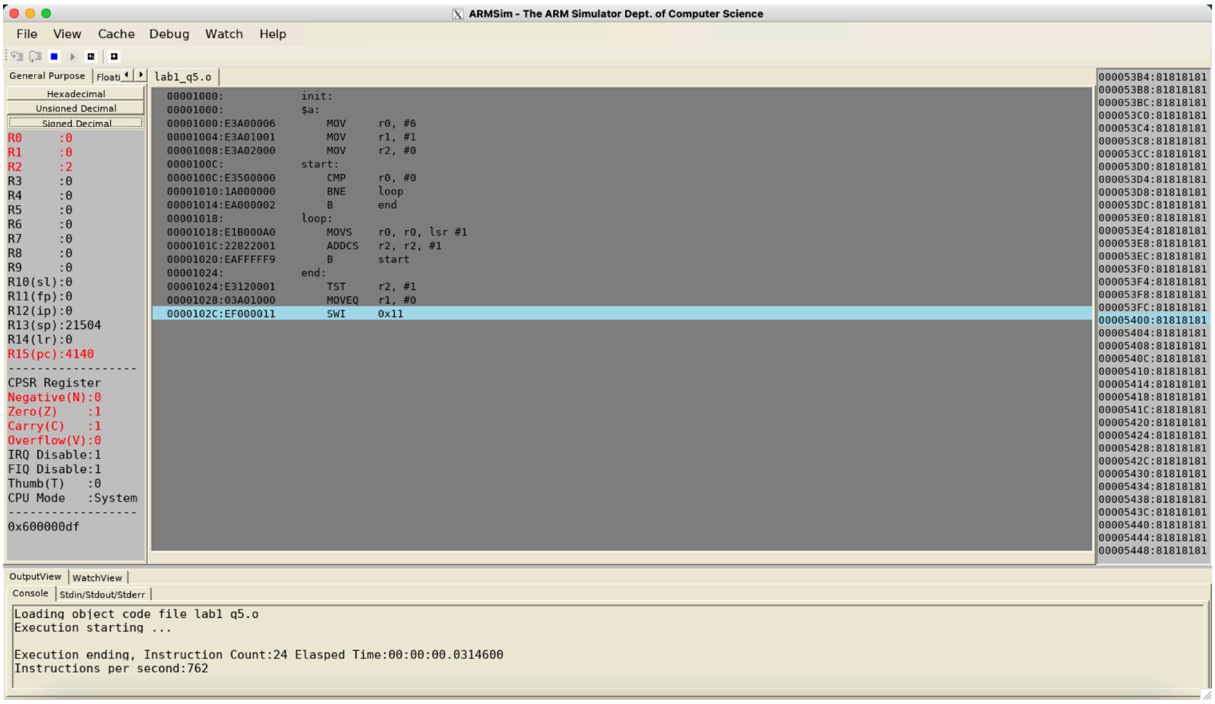
Assignments Questions

5 Write an ALP using ARM7TDMI to find whether the given number is even parity.



ARMSim_files > lab1_q5.s

```
@ Write an ALP using ARM7TDMI to find whether the given number is even parity.
@ Set 0 on even parity and 1 on odd parity

.text

init:
    MOV R0, #6
    MOV R1, #1
    MOV R2, #0

start:
    CMP R0, #0
    BNE loop
    B end

loop:
    MOVS R0, R0, LSR#1
    ADDCS R2, R2, #1
    B start

end:
    TST R2, #1
    MOVEQ R1, #0
    SWI 0x011

```



```
R0       :0
R1       :0
R2       :2
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):4140
------------------
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
```

```
00001000:              init:
00001000:              $a:
00001000:E3A00006          MOV      r0, #6
00001004:E3A01001          MOV      r1, #1
00001008:E3A02000          MOV      r2, #0
0000100C:              start:
0000100C:E3500000          CMP      r0, #0
00001010:1A000000          BNE      loop
00001014:EA000002          B        end
00001018:              loop:
00001018:E1B000A0          MOVS     r0, r0, lsr #1
0000101C:22822001          ADDCS    r2, r2, #1
00001020:EAFFFFF9          B        start
00001024:              end:
00001024:E3120001          TST      r2, #1
00001028:03A01000          MOVEQ    r1, #0
0000102C:EF000011          SWI      0x11
```

```
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
```

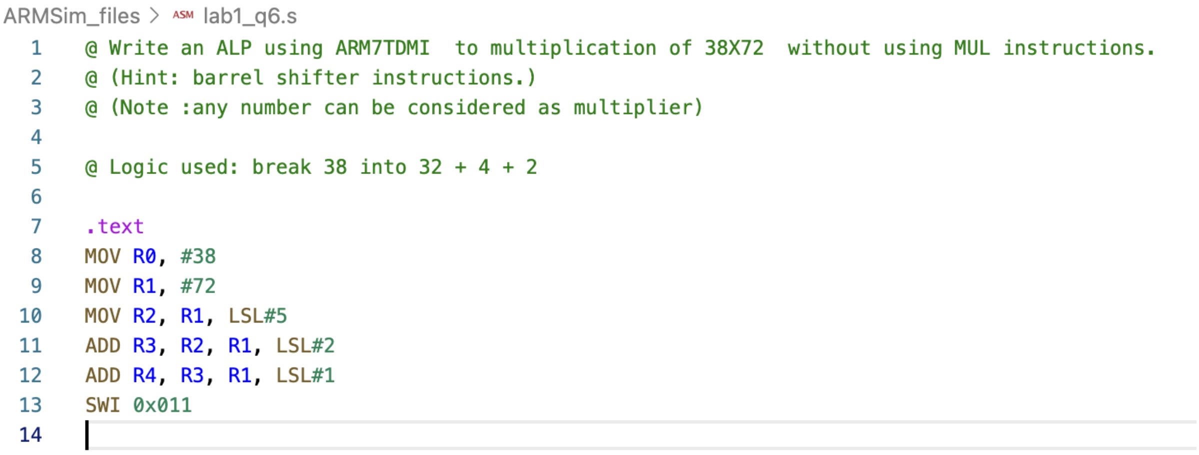
```
Loading object code file lab1 q5.o
Execution starting ...

Execution ending, Instruction Count:24 Elasped Time:00:00:00.0314600
Instructions per second:762
```

6

Write an ALP using ARM7TDMI to multiplication of 38X72 without using MUL instructions.

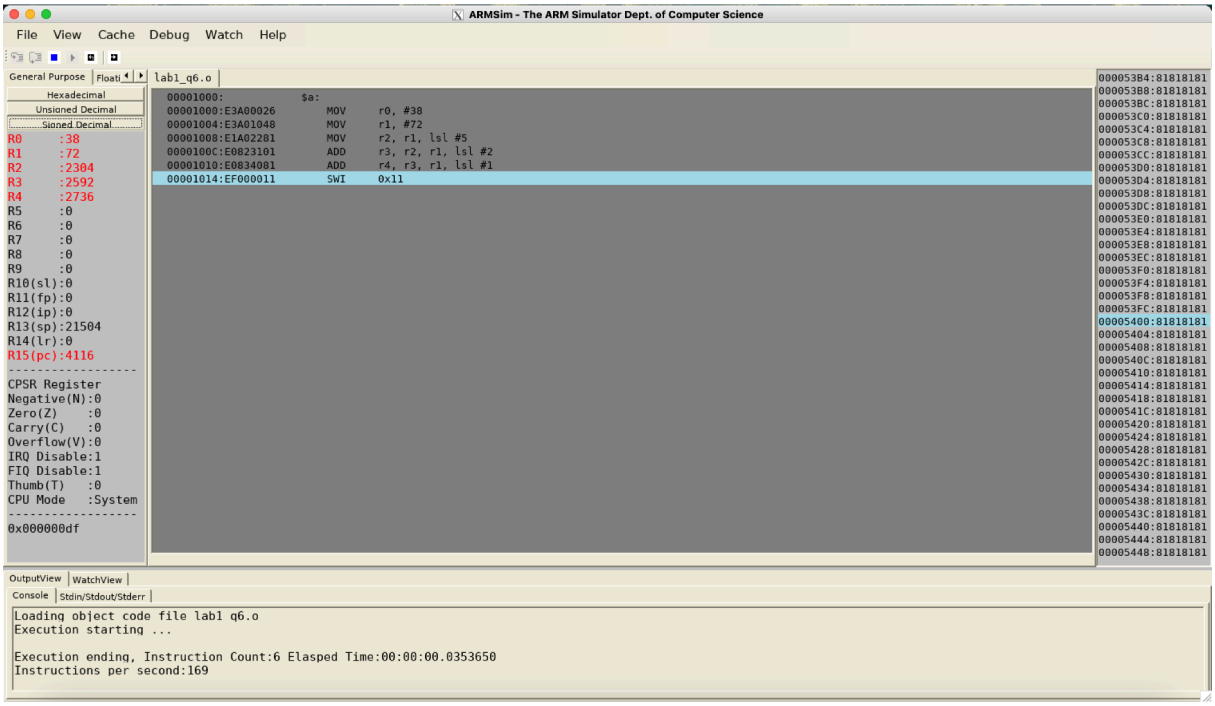
(Hint: barrel shifter instructions.)

(Note :any number can be considered as multiplier)



ARMSim_files > lab1_q6.s

```
@ Write an ALP using ARM7TDMI  to multiplication of 38X72  without using MUL instructions.
@ (Hint: barrel shifter instructions.)
@ (Note :any number can be considered as multiplier)

@ Logic used: break 38 into 32 + 4 + 2

.text
MOV R0, #38
MOV R1, #72
MOV R2, R1, LSL#5
ADD R3, R2, R1, LSL#2
ADD R4, R3, R1, LSL#1
SWI 0x011

```



```
R0      :38
R1      :72
R2      :2304
R3      :2592
R4      :2736
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):4116
------------------
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
```

```
00001000:          $a:
00001000:E3A00026       MOV     r0, #38
00001004:E3A01048       MOV     r1, #72
00001008:E1A02281       MOV     r2, r1, lsl #5
0000100C:E0823101       ADD     r3, r2, r1, lsl #2
00001010:E0834081       ADD     r4, r3, r1, lsl #1
00001014:EF000011       SWI     0x11
```

```
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
```

```
Loading object code file lab1 q6.o
Execution starting ...

Execution ending, Instruction Count:6 Elasped Time:00:00:00.0353650
Instructions per second:169
```