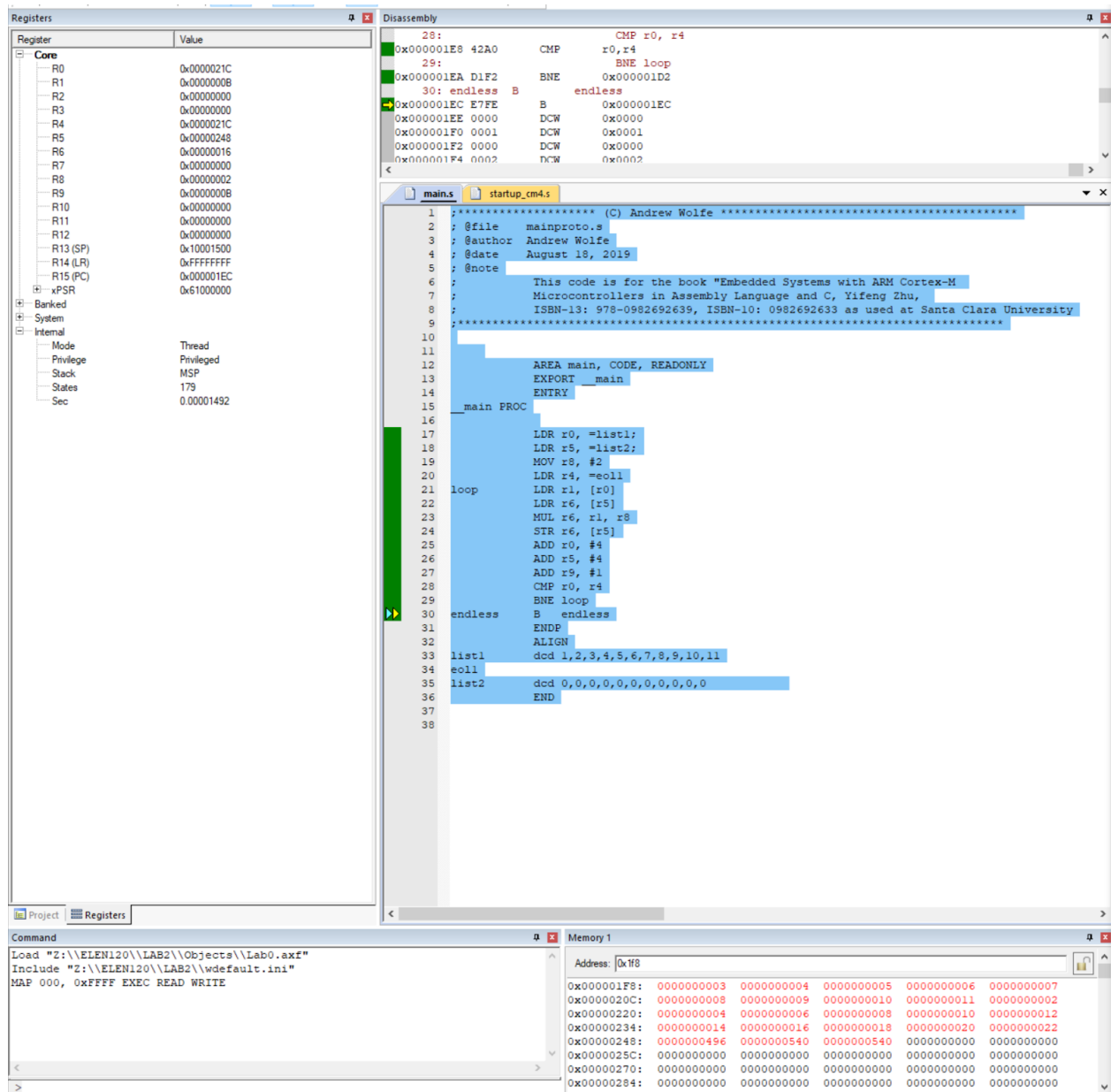


1a.

```
***** (C) Andrew Wolfe *****  
;  
; @file mainproto.s  
; @author Andrew Wolfe  
; @date August 18, 2019  
; @note  
; This code is for the book "Embedded Systems with ARM Cortex-M  
; Microcontrollers in Assembly Language and C, Yifeng Zhu,  
; ISBN-13: 978-0982692639, ISBN-10: 0982692633 as used at Santa Clara University  
*****  
;
```

```
                AREA main, CODE, READONLY  
                EXPORT __main  
                ENTRY  
  
__main PROC  
  
                LDR r0, =list1;  
                LDR r5, =list2;  
                MOV r8, #2  
                LDR r4, =eol1  
loop            LDR r1, [r0]  
                LDR r6, [r5]  
                MUL r6, r1, r8  
                STR r6, [r5]  
                ADD r0, #4  
                ADD r5, #4  
                ADD r9, #1  
                CMP r0, r4  
                BNE loop  
endless        B      endless  
                ENDP  
                ALIGN  
list1          dcd 1,2,3,4,5,6,7,8,9,10,11  
eol1  
list2          dcd 0,0,0,0,0,0,0,0,0,0,0  
                END
```

Screenshot:



1b.

Code:

```
***** (C) Andrew Wolfe *****
; @file mainproto.s
; @author Andrew Wolfe
; @date August 18, 2019
; @note
; This code is for the book "Embedded Systems with ARM Cortex-M
```

; Microcontrollers in Assembly Language and C, Yifeng Zhu,
; ISBN-13: 978-0982692639, ISBN-10: 0982692633 as used at Santa Clara University
.,*****
,

```
                AREA main, CODE, READONLY
                EXPORT __main
                ENTRY

__main PROC

                LDR r0, =list1;
                LDR r5, =list2;
                MOV r8, #2
                LDR r4, =eol1
loop            LDR r1, [r0]
                LDR r6, [r5]
                MUL r6, r1, r8
                STR r6, [r5]
                ADD r0, #4
                ADD r5, #4
                ADD r9, #1
                CMP r0, r4
                BNE loop
endless        B      endless
                ENDP
                ALIGN
list1          dcd 1,2,3,4,5,6,7,8,9,10,11,12
eol1
list2          dcd 0,0,0,0,0,0,0,0,0,0,0,0
                END
```

Screenshot:

Registers

Register	Value
Core	
R0	0x0000220
R1	0x000000C
R2	0x0000000
R3	0x0000000
R4	0x0000220
R5	0x0000250
R6	0x0000018
R7	0x0000000
R8	0x0000002
R9	0x000000C
R10	0x0000000
R11	0x0000000
R12	0x0000000
R13 (SP)	0x10001500
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00001EC
xPSR	0x61000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	190
Sec	0.00001583

Disassembly

```

28:      CMP r0, r4
0x000001E8 42A0    CMP    r0,r4
29:      BNE loop
0x000001EA D1F2    BNE    0x000001D2
30:      endless B
0x000001EC E7FE    B      0x000001EC
0x000001EE 0000    DCW    0x0000
0x000001F0 0001    DCW    0x0001
0x000001F2 0000    DCW    0x0000
0x000001F4 0002    DCW    0x0002

```

main.s

startup_cm4.s

```

1  ;***** (C) Andrew Wolfe *****
2  ; @file    mainproto.s
3  ; @author  Andrew Wolfe
4  ; @date    August 18, 2019
5  ; @note
6  ;         This code is for the book "Embedded Systems with ARM Cortex-M
7  ;         Microcontrollers in Assembly Language and C, Yifeng Zhu,
8  ;         ISBN-13: 978-0982692639, ISBN-10: 0982692633 as used at Santa Clara University
9  ;*****
10
11
12      AREA main, CODE, READONLY
13      EXPORT __main
14      ENTRY __main
15
16      __main PROC
17
18          LDR r0, =list1;
19          LDR r5, =list2;
20          MOV r8, #2
21          LDR r4, =eoll
22      loop
23          LDR r1, [r0]
24          LDR r6, [r5]
25          MUL r6, r1, r8
26          STR r6, [r5]
27          ADD r0, #4
28          ADD r5, #4
29          ADD r9, #1
30          CMP r0, r4
31          BNE loop
32      endless B endless
33      ENDP
34      ALIGN
35      list1 dcd 1,2,3,4,5,6,7,8,9,10,11,12
36      eoll
37      list2 dcd 0,0,0,0,0,0,0,0,0,0,0,0
38      END

```

Project

Registers

Command

```

Load "Z:\ELEN120\LAB2\Objects\Lab0.axf"
Include "Z:\ELEN120\LAB2\wdefault.ini"
MAP 000, 0xFFFF EXEC READ WRITE

```

Memory 1

Address: 0x1F8					
0x000001F8:	0000000003	0000000004	0000000005	0000000006	0000000007
0x0000020C:	0000000008	0000000009	0000000010	0000000011	0000000012
0x00000220:	0000000002	0000000004	0000000006	0000000008	0000000010
0x00000234:	0000000012	0000000014	0000000016	0000000018	0000000020
0x00000248:	0000000022	0000000024	0000000046	0000000054	0000000054
0x0000025C:	0000000000	0000000000	0000000000	0000000000	0000000000
0x00000270:	0000000000	0000000000	0000000000	0000000000	0000000000
0x00000284:	0000000000	0000000000	0000000000	0000000000	0000000000

2.

Code:

```

AREA main, CODE, READONLY
EXPORT __main
ENTRY

```

```
__main PROC
```

```

ldr      r0,=data
ldr      r1,[r0]
cmp      r1,#0xffffffff
bne      fail
add      r0,#4
ldr      r1,[r0]
cmp      r1,#0xffffffff
bmi      fail
add      r0,#8
ldr      r1,[r0]
cmp      r1,#0x00000000
bls      fail
sub      r0,#4
ldr      r1,[r0]
cmn      r1,#0xffffffff
bne      fail
add      r0,#8
ldr      r1,[r0]
teq      r1,#0xffffffff
bne      fail
add      r0,#4
ldr      r1,[r0]
tst      r1,#0xffffffff
bne      fail
add      r0,#8
ldr      r1,[r0]
tst      r1,#0x02
beq      fail
sub      r0,#4
ldr      r1,[r0]
teq      r1,#0xffffffff
beq      fail

```

```

pass    b
fail    b

```

```

pass
fail

```

```

ENDP
ALIGN

```

```

data    dcd
eodata

```

```

-1,0,1,1,-1,0,0,-1

```

```

END

```

Screenshot:


```

                                EXPORT    __main
                                ENTRY
__main      PROC

                                ldr     r2,=listsize
                                ldr     r3,[r2]
                                ldr     r2,=list
                                mov     r0,#0
loop        ldrb    r1,[r2]
                                sxtb    r1,r1
                                cmp     r1,#0
                                addgt   r0,#1
                                add     r2,#1
                                subs    r3,#1
                                bne     loop

                                b        endless
                                endless

                                ENDP
                                ALIGN
list        dcb     -2, 0, 1, 5, -3, -5, 3, 9, 2, 12
                                ALIGN
listsize    dcd     10

                                END

```

Screenshot:

Registers

Register	Value
Core	
R0	0x00000006
R1	0x0000000C
R2	0x000001F2
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x10001500
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00001E4
xPSR	0x61000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	168
Sec	0.00001400

Disassembly

```

0x000001E2 D1F6 BNE 0x000001D2
28: endless b
0x000001E4 E7FE B 0x000001E4 endless
0x000001E6 0000 DCW 0x0000
0x000001E8 00FE DCW 0x00FE
0x000001EA 0501 DCW 0x0501
0x000001EC FBFD DCW 0xFBFD
0x000001EE 0903 DCW 0x0903
0x000001F0 0C02 DCW 0xC02
0x000001F2 0000 DCW 0x0000

```

main.s

startup_cm4.s

```

1
2
3 ***** (C) Andrew Wolfe *****
4 @file HW4 Problem 3
5 @author Andrew Wolfe
6 @date Aug. 13, 2023
7
8
9
10 AREA main, CODE, READONLY
11 EXPORT __main
12 ENTRY
13
14 __main PROC
15
16     ldr r2,=listsize
17     ldr r3,[r2]
18     ldr r2,=list
19     mov r0,#0
20 loop    ldrb r1,[r2]
21         sxtb r1,r1
22         cmp r1,#0
23         addgt r0,#1
24         add r2,#1
25         subs r3,#1
26         bne loop
27
28 endless b endless
29
30 ENDP
31 ALIGN
32 list    dcb -2, 0, 1, 5, -3, -5, 3, 9, 2, 12
33 ALIGN
34 listsize dcd 10
35
36 END

```

Command

```

Load "Z:\\ELEN120\\LAB2\\Objects\\Lab0.axf"
Include "Z:\\ELEN120\\LAB2\\wdefault.ini"
MAP 000, 0xFFFF EXEC READ WRITE

```

Memory 1

Address	0x10
0x000001F0:	0000003074 0000000010 0000000500 0000000485 0000000000
0x00000204:	0000000000 0000000000 0000000000 0000000000 0000000000
0x00000218:	0000000000 0000000000 0000000000 0000000000 0000000000
0x0000022C:	0000000000 0000000000 0000000000 0000000000 0000000000
0x00000240:	0000000000 0000000000 0000000000 0000000000 0000000000
0x00000254:	0000000000 0000000000 0000000000 0000000000 0000000000
0x00000268:	0000000000 0000000000 0000000000 0000000000 0000000000
0x0000027C:	0000000000 0000000000 0000000000 0000000000 0000000000