Assignment 3 Programming Proofs

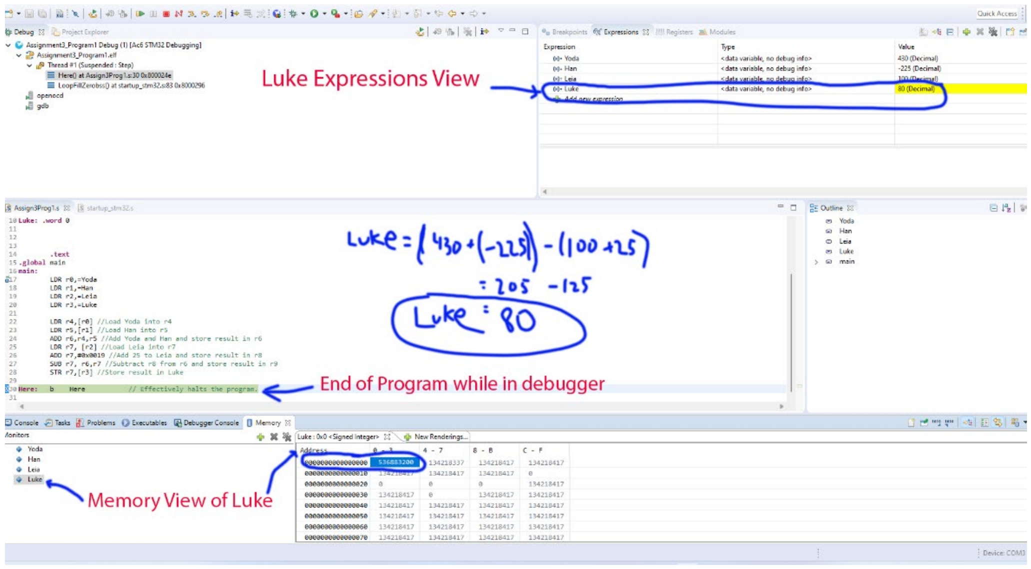
Jonathan Elder

CPSC 3300

Program 1

Source Code

```
             .data
             .global Yoda
             .global Han
             .global Leia
             .global Luke

Yoda: .word 0
Han: .word 0
Leia: .word 0
Luke: .word 0



             .text
.global main
main:
             LDR r0,=Yoda
             LDR r1,=Han
             LDR r2,=Leia
             LDR r3,=Luke

             LDR r4,[r0] //Load Yoda into r4
             LDR r5,[r1] //Load Han into r5
             ADD r6,r4,r5 //Add Yoda and Han and store result in r6
             LDR r7, [r2] //Load Leia into r7
             ADD r7,#0x0019 //Add 25 to Leia and store result in r8
             SUB r7, r6,r7 //Subtract r8 from r6 and store result in r9
             STR r7,[r3] //Store result in Luke

            Here:   b    Here           // Effectively halts the program.
```



```
Assignment3_Program1 Debug (1) [Ac6 STM32 Debugging]
  Assignment3_Program1.elf
    Thread #1 (Suspended : Step)
      Here() at Assign3Prog1.s:30 0x800024e
      LoopFillZerobss() at startup_stm32.s:83 0x8000296
  openocd
  gdb
```

Expression	Type	Value
Yoda	<data variable, no debug info>	430 (Decimal)
Han	<data variable, no debug info>	-225 (Decimal)
Leia	<data variable, no debug info>	100 (Decimal)
Luke	<data variable, no debug info>	80 (Decimal)
Add new expression		

Assign3Prog1.s   startup_stm32.s

```
Luke: .word 0



        .text
.global main
main:
        LDR r0,=Yoda
        LDR r1,=Han
        LDR r2,=Leia
        LDR r3,=Luke

        LDR r4,[r0] //Load Yoda into r4
        LDR r5,[r1] //Load Han into r5
        ADD r6,r4,r5 //Add Yoda and Han and store result in r6
        LDR r7, [r2] //Load Leia into r7
        ADD r7,#0x0019 //Add 25 to Leia and store result in r8
        SUB r7, r6,r7 //Subtract r8 from r6 and store result in r9
        STR r7,[r3] //Store result in Luke

Here:   b    Here           // Effectively halts the program.
```

$$\text{Luke} = (430 + (-225)) - (100 + 25)$$
$$= 205 \quad -125$$
$$\boxed{\text{Luke} = 80}$$

Outline: Yoda, Han, Leia, Luke, main

Monitors: Yoda, Han, Leia, Luke

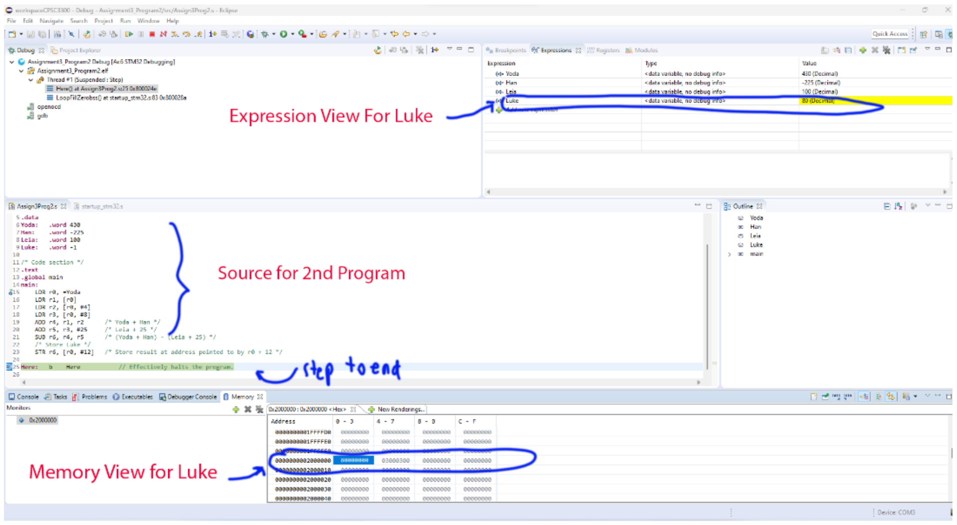
Luke : 0x0 <Signed Integer>

Address	0 - 3	4 - 7	8 - B	C - F
0000000000000000	536883200	134218337	134218417	134218417
0000000000000010	134218417	134218417	134218417	0
0000000000000020	0	0	0	134218417
0000000000000030	134218417	0	134218417	134218417
0000000000000040	134218417	134218417	134218417	134218417
0000000000000050	134218417	134218417	134218417	134218417
0000000000000060	134218417	134218417	134218417	134218417
0000000000000070	134218417	134218417	134218417	134218417

Program 2

Source Code

```
.syntax unified

// Define data section and allocate space for variables
/* Data section */
.data
Yoda:    .word 430
Han:     .word -225
Leia:    .word 100
Luke:    .word -1

/* Code section */
.text
.global main
main:
    LDR r0, =Yoda
    LDR r1, [r0]
    LDR r2, [r0, #4]
    LDR r3, [r0, #8]
    ADD r4, r1, r2      /* Yoda + Han */
    ADD r5, r3, #25     /* Leia + 25 */
    SUB r6, r4, r5      /* (Yoda + Han) - (Leia + 25) */
    /* Store Luke */
    STR r6, [r0, #12]   /* Store result at address pointed to by r0 + 12 */

Here:    b    Here           // Effectively halts the program.
```



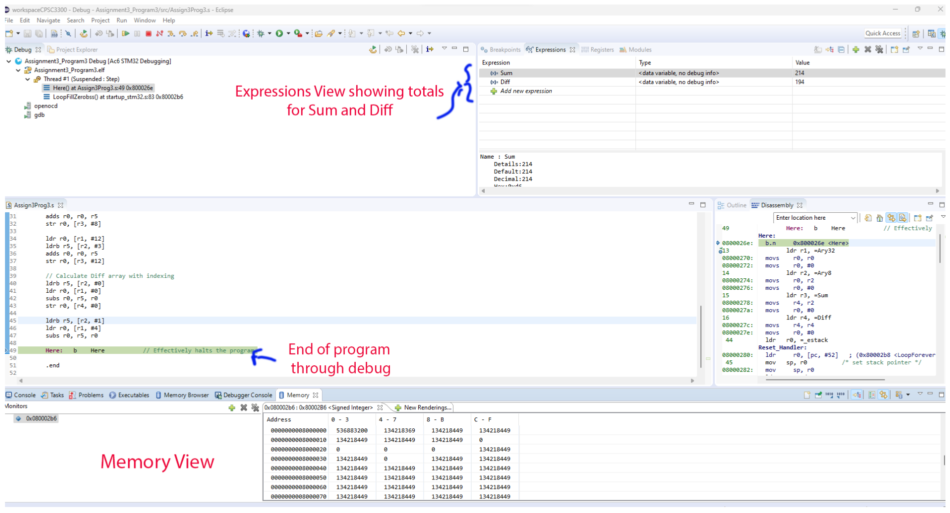
Expression	Type	Value
Yoda	<data variable, no debug info>	430 (Decimal)
Han	<data variable, no debug info>	-225 (Decimal)
Leia	<data variable, no debug info>	100 (Decimal)
Luke	<data variable, no debug info>	80 (Decimal)

```
.data
Yoda:   .word 430
Han:    .word -225
Leia:   .word 100
Luke:   .word -1

/* Code section */
.text
.global main
main:
    LDR r0, =Yoda
    LDR r1, [r0]
    LDR r2, [r0, #4]
    LDR r3, [r0, #8]
    ADD r4, r1, r2      /* Yoda + Han */
    ADD r5, r3, #25     /* Leia + 25 */
    SUB r6, r4, r5      /* (Yoda + Han) - (Leia + 25) */
    /* Store Luke */
    STR r6, [r0, #12]   /* Store result at address pointed to by r0 + 12 */

Here:   b    Here          // Effectively halts the program.
```

Program 3

Source Code

```
.syntax unified

    .data
    Ary32:  .word  10, -5, 0xFFFFFFF4, 0x77777777
    Ary8:   .byte  0xCC, 12, -3, 0xB
    Sum:    .word  0, 0, 0, 0
    Diff:   .word  0, 0, 0, 0

    .text
    .global main
main:
        // Initialize pointers to arrays
        ldr r1, =Ary32
        ldr r2, =Ary8
        ldr r3, =Sum
        ldr r4, =Diff

        // Calculate Sum array using indexing
        ldr r0, [r1, #0]
        ldrb r5, [r2, #0]
        adds r0, r0, r5
        str r0, [r3, #0]

        ldr r0, [r1, #4]
        ldrb r5, [r2, #1]
        adds r0, r0, r5
        str r0, [r3, #4]
        ldr r0, [r1, #8]
        ldrb r5, [r2, #2]
        adds r0, r0, r5
        str r0, [r3, #8]

        ldr r0, [r1, #12]
        ldrb r5, [r2, #3]
        adds r0, r0, r5
        str r0, [r3, #12]

        // Calculate Diff array with indexing
        ldrb r5, [r2, #0]
        ldr r0, [r1, #0]
        subs r0, r5, r0
        str r0, [r4, #0]

        ldrb r5, [r2, #1]
        ldr r0, [r1, #4]
        subs r0, r5, r0

Here:   b     Here            // Effectively halts the program.

            .end
```



```
Assignment3_Program3 Debug [Ac6 STM32 Debugging]
  Assignment3_Program3.elf
    Thread #1 (Suspended : Step)
      Here() at Assign3Prog3.s:49 0x800026e
      LoopFillZerobss() at startup_stm32.s:83 0x80002b6
  openocd
  gdb
```

Expression	Type	Value
Sum	<data variable, no debug info>	214
Diff	<data variable, no debug info>	194
Add new expression		

```
Name : Sum
    Details:214
    Default:214
    Decimal:214
```

Assign3Prog3.s

```
        adds r0, r0, r5
        str r0, [r3, #8]

        ldr r0, [r1, #12]
        ldrb r5, [r2, #3]
        adds r0, r0, r5
        str r0, [r3, #12]

        // Calculate Diff array with indexing
        ldrb r5, [r2, #0]
        ldr r0, [r1, #0]
        subs r0, r5, r0
        str r0, [r4, #0]

        ldrb r5, [r2, #1]
        ldr r0, [r1, #4]
        subs r0, r5, r0

        Here:   b    Here         // Effectively halts the program

        .end
```

```
49                Here:   b    Here          // Effectively
          Here:
0800026e:   b.n      0x800026e <Here>
13                ldr r1, =Ary32
08000270:   movs     r0, r0
08000272:   movs     r0, #0
14                ldr r2, =Ary8
08000274:   movs     r0, r2
08000276:   movs     r0, #0
15                ldr r3, =Sum
08000278:   movs     r4, r2
0800027a:   movs     r0, #0
16                ldr r4, =Diff
0800027c:   movs     r4, r4
0800027e:   movs     r0, #0
 44         ldr    r0, =_estack
          Reset_Handler:
08000280:   ldr      r0, [pc, #52]   ; (0x80002b8 <LoopForever
 45         mov    sp, r0          /* set stack pointer */
08000282:   mov      sp, r0
```

0x080002b6

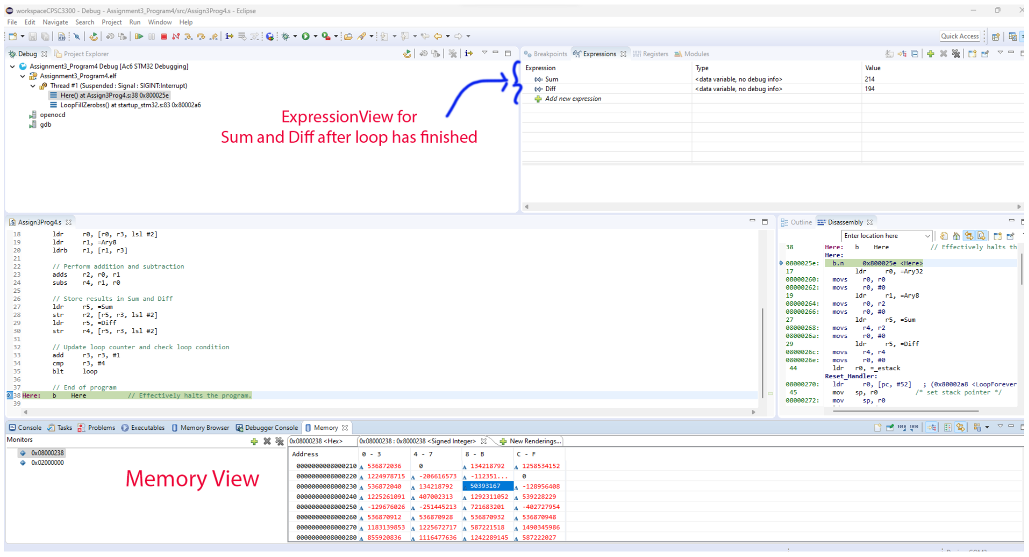
0x080002b6 : 0x80002B6 <Signed Integer>

```
Address           0 - 3      4 - 7      8 - B      C - F
0000000008000000  536883200  134218369  134218449  134218449
0000000008000010  134218449  134218449  134218449  0
0000000008000020  0          0          0          134218449
0000000008000030  134218449  0          134218449  134218449
0000000008000040  134218449  134218449  134218449  134218449
0000000008000050  134218449  134218449  134218449  134218449
0000000008000060  134218449  134218449  134218449  134218449
0000000008000070  134218449  134218449  134218449  134218449
```

Program 4

Source Code

```
        .syntax unified

        .data
Ary32:  .word  10, -5, 0xFFFFFFF4, 0x77777777
Ary8:   .byte  0xCC, 12, -3, 0xB
Sum:    .word  0, 0, 0, 0
Diff:   .word  0, 0, 0, 0

        .text
        .global main
main:
        // Initialize loop counter
        mov     r3, #0

loop:
        // Load values from Ary32 and Ary8 into registers
        ldr     r0, =Ary32
        ldr     r0, [r0, r3, lsl #2]
        ldr     r1, =Ary8
        ldrb    r1, [r1, r3]

        // Perform addition and subtraction
        adds    r2, r0, r1
        subs    r4, r1, r0

        // Store results in Sum and Diff
        ldr     r5, =Sum
        str     r2, [r5, r3, lsl #2]
        ldr     r5, =Diff
        str     r4, [r5, r3, lsl #2]

        // Update loop counter and check loop condition
        add     r3, r3, #1
        cmp     r3, #4
        blt     loop

        // End of program
Here:   b    Here           // Effectively halts the program.
```



```
Assignment3_Program4 Debug [Ac6 STM32 Debugging]
  Assignment3_Program4.elf
    Thread #1 (Suspended : Signal : SIGINT:Interrupt)
      Here() at Assign3Prog4.s:38 0x800025e
      LoopFillZerobss() at startup_stm32.s:83 0x80002a6
  openocd
  gdb
```

Expression	Type	Value
Sum	<data variable, no debug info>	214
Diff	<data variable, no debug info>	194
Add new expression		

```
        ldr     r0, [r0, r3, lsl #2]
        ldr     r1, =Ary8
        ldrb    r1, [r1, r3]

        // Perform addition and subtraction
        adds    r2, r0, r1
        subs    r4, r1, r0

        // Store results in Sum and Diff
        ldr     r5, =Sum
        str     r2, [r5, r3, lsl #2]
        ldr     r5, =Diff
        str     r4, [r5, r3, lsl #2]

        // Update loop counter and check loop condition
        add     r3, r3, #1
        cmp     r3, #4
        blt     loop

        // End of program
Here:   b    Here           // Effectively halts the program.
```

```
38          Here:   b    Here         // Effectively halts th
            Here:
0800025e:     b.n     0x800025e <Here>
17                  ldr     r0, =Ary32
08000260:     movs    r0, r0
08000262:     movs    r0, #0
19                  ldr     r1, =Ary8
08000264:     movs    r0, r2
08000266:     movs    r0, #0
27                  ldr     r5, =Sum
08000268:     movs    r4, r2
0800026a:     movs    r0, #0
29                  ldr     r5, =Diff
0800026c:     movs    r4, r4
0800026e:     movs    r0, #0
 44           ldr   r0, =_estack
            Reset_Handler:
08000270:     ldr     r0, [pc, #52]   ; (0x80002a8 <LoopForever
 45           mov   sp, r0          /* set stack pointer */
08000272:     mov     sp, r0
```

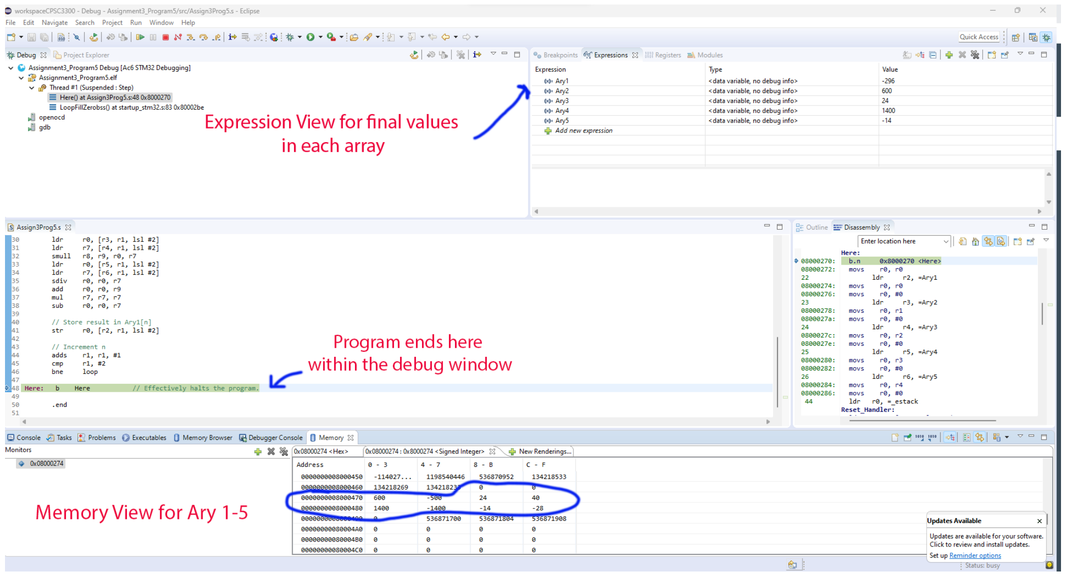
```
0x08000238
0x02000000

0x08000238 <Hex>    0x08000238 : 0x8000238 <Signed Integer>

Address             0 - 3        4 - 7        8 - B        C - F
0000000008000210    536872036    0            134218792    1258534152
0000000008000220    1224978715   -206616573   -112351...   0
0000000008000230    536872040    134218792    50393167     -128956408
0000000008000240    1225261091   407002313    1292311052   539228229
0000000008000250    -129676026   -251445213   721683201    -402727954
0000000008000260    536870912    536870928    536870932    536870948
0000000008000270    1183139853   1225672717   587221518    1490345986
0000000008000280    855920836    1116477636   1242289145   587222027
```

Program 5

Source Code

```
        .syntax unified

        .data

        // Define the arrays
        Ary1:   .word   0, 0
        Ary2:   .word   600, -500
        Ary3:   .word   24, 40
        Ary4:   .word   1400, -1400
        Ary5:   .word   -14, -28

        .text
        .align

        .global main
        .type main, %function

main:

        // Initialize variables
        movs    r1, #0 // Variable N
        ldr     r2, =Ary1
        ldr     r3, =Ary2
        ldr     r4, =Ary3
        ldr     r5, =Ary4
        ldr     r6, =Ary5
```

```
loop:
        //  Compute expression for N
        ldr     r0, [r3, r1, lsl #2]
        ldr     r7, [r4, r1, lsl #2]
        smull   r8, r9, r0, r7
        ldr     r0, [r5, r1, lsl #2]
        ldr     r7, [r6, r1, lsl #2]
        sdiv    r0, r0, r7
        add     r0, r0, r9
        mul     r7, r7, r7
        sub     r0, r0, r7

        // Store result in Ary1[n]
        str     r0, [r2, r1, lsl #2]

        // Increment n
        adds    r1, r1, #1
        cmp     r1, #2
        bne     loop

 Here:    b     Here            // Effectively halts the program.

              .end
```



Assignment3_Program5 Debug [Ac6 STM32 Debugging]
Assignment3_Program5.elf
Thread #1 (Suspended : Step)
Here() at Assign3Prog5.s:48 0x8000270
LoopFillZerobss() at startup_stm32.s:83 0x80002be
openocd
gdb

Expression	Type	Value
Ary1	<data variable, no debug info>	-296
Ary2	<data variable, no debug info>	600
Ary3	<data variable, no debug info>	24
Ary4	<data variable, no debug info>	1400
Ary5	<data variable, no debug info>	-14
Add new expression		

```
        ldr     r0, [r3, r1, lsl #2]
        ldr     r7, [r4, r1, lsl #2]
        smull   r8, r9, r0, r7
        ldr     r0, [r5, r1, lsl #2]
        ldr     r7, [r6, r1, lsl #2]
        sdiv    r0, r0, r7
        add     r0, r0, r9
        mul     r7, r7, r7
        sub     r0, r0, r7

        // Store result in Ary1[n]
        str     r0, [r2, r1, lsl #2]

        // Increment n
        adds    r1, r1, #1
        cmp     r1, #2
        bne     loop

Here:   b       Here            // Effectively halts the program.

        .end
```

```
                 Here:
08000270:          b.n     0x8000270 <Here>
08000272:          movs    r0, r0
22                      ldr     r2, =Ary1
08000274:          movs    r0, r0
08000276:          movs    r0, #0
23                      ldr     r3, =Ary2
08000278:          movs    r0, r1
0800027a:          movs    r0, #0
24                      ldr     r4, =Ary3
0800027c:          movs    r0, r2
0800027e:          movs    r0, #0
25                      ldr     r5, =Ary4
08000280:          movs    r0, r3
08000282:          movs    r0, #0
26                      ldr     r6, =Ary5
08000284:          movs    r0, r4
08000286:          movs    r0, #0
 44                ldr   r0, =_estack
                 Reset_Handler:
```

0x08000274 <Hex>
0x08000274 : 0x8000274 <Signed Integer>
New Renderings...

```
Address            0 - 3        4 - 7        8 - B        C - F
0000000008000450   -114027...   1198540446   536870952    134218533
0000000008000460   134218269    13421823[illegible]  0    0
0000000008000470   600          -500         24           40
0000000008000480   1400         -1400        -14          -28
0000000008000490   0            536871700    536871804    536871908
00000000080004A0   0            0            0            0
00000000080004B0   0            0            0            0
00000000080004C0   0            0            0            0
```

Updates are available for your software. Click to review and install updates.
Set up Reminder options