**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.

Graphical user interface, application

Description automatically generated

**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.

Graphical user interface, application

Description automatically generated

**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

Graphical user interface

Description automatically generated with medium confidence

**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.

Graphical user interface, application

Description automatically generated

**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

Graphical user interface, application

Description automatically generated