**Mainline Algorithm**

Initialize R1 to index value (16)

Point R2 to Fahrenheit\_Array

Point R3 to Celsius\_Array

Point R4 to Fahrenheit\_Ave

Point R5 to Celsius\_Ave

Jump to Convert function

Move R2 to R0, passing Fahrenheit array to average procedure

Jump to Average procedure, which stores average in R0

Store R0 in R4

Move R3 to R0, passing Celsius array to Average procedure

Jump to Average procedure

Store R0 in R5

**Average Procedure**

**In English:**

Push registers to stack

Set Count to zero

Loop through array, adding up values in Total

IF Total is negative

Total = -Total

incrementing value = -1

ELSE

incrementing value = 1

WHILE Total > Index

Total -= Index

Count += incrementing value

IF Total >= Index/2

Count += incrementing value

Restore registers

Return

**The same algorithm, written in something closer to assembly:**

I’m using the registers thusly:

R0 has pointer to array

R1 is index value

R2 is accumulator

R3 stores index copy

R4 is the incrementation value

R5 is zero, for R2 = |R2|

R6 is stores individual array values

R0 becomes division counter

Push R1-R5, R14 to the stack

Initialize R2 to zero

Copy R1 to R3

DO

Add value at R0 to R2 (R2 is the array total)

Decrement R1

WHILE (R1 > 0)

Compare R2 to zero

IF (R2 < 0)

R5 = 0

R2 = R5 - R2

R4 = -1

ELSE

R4 = 1

WHILE (R2 > R3)

R2 - R3

R0 += R4

R3 >> 1 (Divide R3 by 2)

IF (R2 > R3)

R0 += R4

Pop registers

Return

**Conversion Algorithm**

For conversion of a single value from Fahrenheit to Celsius

INC = 0

COUNT = 0

FAHR -= 32

FAHR \*= 5

IF (FAHR < 0)

FAHR = 0 - FAHR

INC = -1

ELSE

INC = 1

WHILE (FAHR >= 9)

FAHR -= 9

COUNT += INC

**For the conversion of the whole array of values:**

INDEX = 16

DO:

Load Fahrenheit value from array into register

Increment Fahrenheit array pointer

[Insert above algorithm here]

Save Celsius value to array

Increment Celsius array pointer

--INDEX

WHILE (INDEX > 0)