**Lab 1 Report**

**COMP3350**

**Adam Biggs – 8/29/2024**

The purpose of this lab was to swap the position of 2 arbitrary elements in an array. Using the template provided I made modifications for readability and functionality. The following is a breakdown of my code and thought process.

**Code Breakdown:**

Establish Variables. I created an array and 2 integers (word). I also assigned their values. The words n, and m, are arbitrary and can be changed.

.data

A: .word 7, 42, 0, 27, 16, 8, 4, 15, 31, 45

n: .word 3

m: .word 6

Setup Main. This is just setup so the following code runs in main.

.text

.globl main

main:

Load Variables into registers. This just makes it so operations can be performed on the variables.

la $a0, A

lw $a1, n

lw $a2, m

Calculate offset for N. This takes the binary of word n and shifts its 1’s left 2 spaces. And then it takes this new offset and adds it to the address of the array to find a specific element in the array. ($a1 = n)

sll $t1, $a1, 2

add $t1, $a0, $t1

Calculate offset for M. This does the same thing for m. It just finds the position of the input in the array. ($a2 = m)

sll $t2, $a2, 2

add $t2, $a0, $t2

Load the actual number of addresses in array. This converts and stores the address back into the actual number in it. For purposes of swapping them.

lw $t4, 0($t1)

lw $t5, 0($t2)

Swap the elements. This literally takes the actual number we just set and swaps with the previous positions where the address was stored.

sw $t4, 0($t2)

sw $t5, 0($t1)