**REPORT LAB 6**

**Assignment 1:**

Graphical user interface, application

Description automatically generated

In the mspfx function:

First, initialize all the parameters that we need to use while performing the algorithms: length, index, running sum and max sum. In the loop, we increase the value of $t2 to 4 unit corresponding to the increment of i. Then we load the value of $t3 to $t4, add that value to the running sum (stored in the register $t5).

We consider 2 cases:

+ If the running sum is greater than the max sum, increase value of length by 1 and store the running sum to $v1. After that we jump to test and check if the value of i is less than n. If not, end the loop

+ If the running sum is not greater than the max sum, run test and continue

The final result is:

$v0 – the length of max-prefix sum: 4

$v1 – the max prefix-sum: 6

**Assignment 2:**

Graphical user interface, table

Description automatically generated

First: At main, we load address of A and Aend, store the A[0] address and A[n-1] address. The array will be stored between those memory bound.

The sort function will stop when there is only 1 remainder element.

The max function init pointers and value to be considered.

The loop function:

+ Check the value of next is equal to last. If true, jump to ret

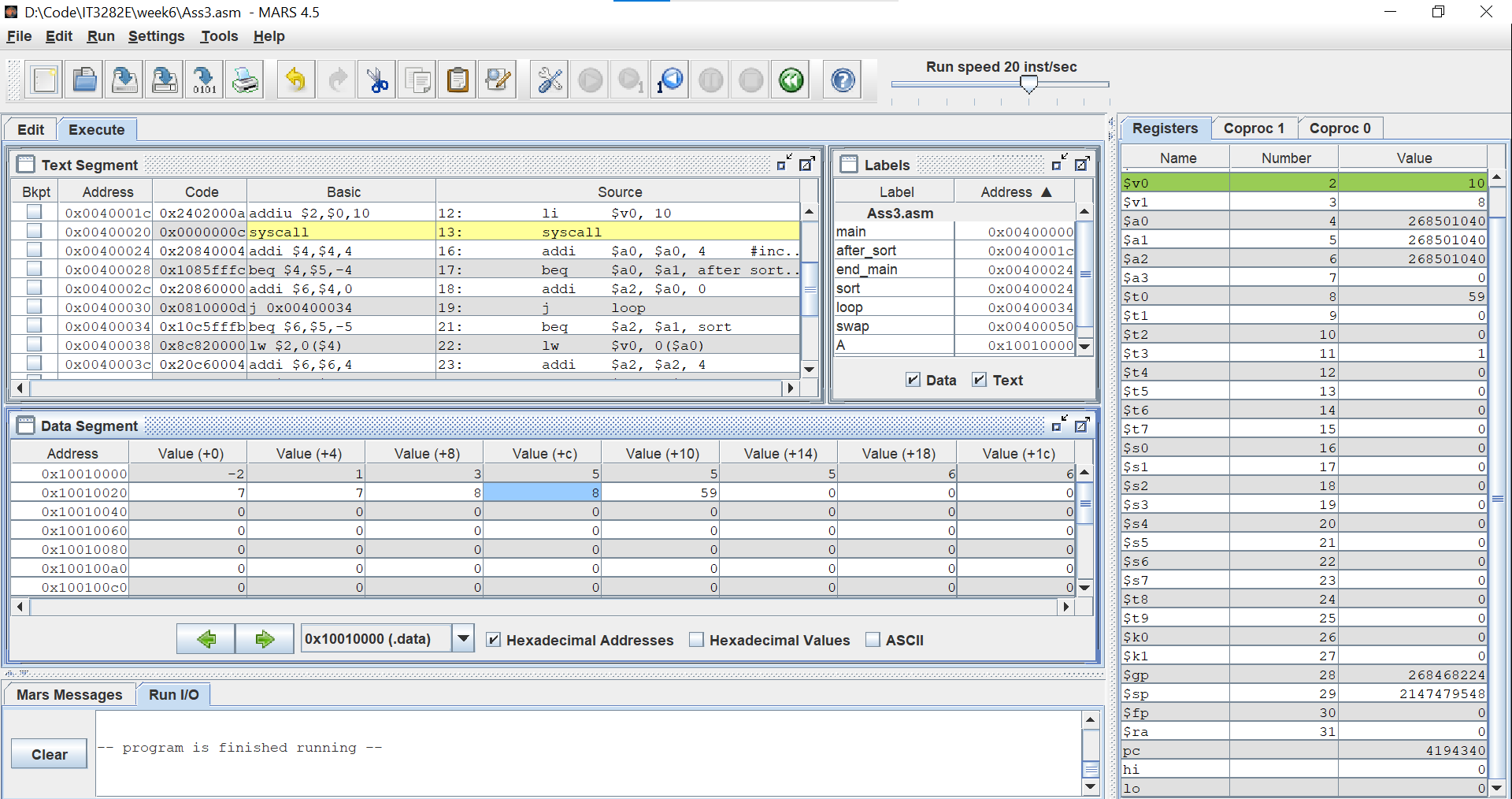
+ Move to next element and load it into $s1. Do it till next value is greater than current max value and assign that value to max value.

The ret function call the after\_max function:

+ Load last element to a temporary register $t0, copy its address and assign max value to that.

+ Then we decrease the pointer to 4 unit to sort a smaller array.

**Assginment 3:**



All of the implementation is the loop. Here we use 2 count variable as $a2 and $a0, and loop until the value at $a2 < $a0($v1 < $v0), then we swap 2 value at those registers.

**Assginment 4:**

Graphical user interface, table

Description automatically generated