**Thực hành kiến trúc máy tính**

**Assignment 1:**

Code:

.data

A: .word -2, 6, -1, 3, -2

.text

main: la $a0,A

li $a1,5

j mspfx

nop

continue:

lock: j lock

nop

end\_of\_main:

# @param[out] v1 the max sum of a certain sub-array

#-----------------------------------------------------------------

#Procedure mspfx

#function: find the maximum-sum prefix in a list of integers

#the base address of this list(A) in $a0 and the number of

#elements is stored in a1

mspfx: addi $v0,$zero,0 #initialize length in $v0 to 0

addi $v1,$zero,0 #initialize max sum in $v1 to 0

addi $t0,$zero,0 #initialize index i in $t0 to 0

addi $t1,$zero,0 #initialize running sum in $t1 to 0

loop: add $t2,$t0,$t0 #put 2i in $t2

add $t2,$t2,$t2 #put 4i in $t2

add $t3,$t2,$a0 #put 4i+A (address of A[i]) in $t3

lw $t4,0($t3) #load A[i] from mem(t3) into $t4

add $t1,$t1,$t4 #add A[i] to running sum in $t1

slt $t5,$v1,$t1 #set $t5 to 1 if max sum < new sum

bne $t5,$zero,mdfy #if max sum is less, modify results

j test #done?

mdfy: addi $v0,$t0,1 #new max-sum prefix has length i+1

addi $v1,$t1,0 #new max sum is the running sum

test: addi $t0,$t0,1 #advance the index i

slt $t5,$t0,$a1 #set $t5 to 1 if i<n

bne $t5,$zero,loop #repeat if i<n

done: j continue

mspfx\_end:

Kết quả:



⦁ Max-sum prefix: $v1 = -2 + 6 - 1 + 3 = 6

⦁ Độ dài của prefix: $v0 = 4

**Assignment 2:**

**Code:**

.data

A: .word 7, -2, 5, 1, 5, 6, 7, 3, 6, 8, 8, 59, 5

Aend: .word

.text

main: la $a0,A #$a0 = Address(A[0])

la $a1,Aend

addi $a1,$a1,-4

j sort #sort

after\_sort: li $v0, 10 #exit

syscall

end\_main:

#--------------------------------------------------------------

#procedure sort (ascending selection sort using pointer)

#register usage in sort program

#$a0 pointer to the first element in unsorted part

#$a1 pointer to the last element in unsorted part

#$t0 temporary place for value of last element

#$v0 pointer to max element in unsorted part

#$v1 value of max element in unsorted part

#--------------------------------------------------------------

sort: beq $a0,$a1,done #single element list is sorted

j max #call the max procedure

after\_max: lw $t0,0($a1) #load last element into $t0

sw $t0,0($v0) #copy last element to max location

sw $v1,0($a1) #copy max value to last element

addi $a1,$a1,-4 #decrement pointer to last element

j sort #repeat sort for smaller list

done: j after\_sort

#-------------------------------------------------------------------

#Procedure max

#function: fax the value and address of max element in the list

#$a0 pointer to first element

#$a1 pointer to last element

#-------------------------------------------------------------------

max:

addi $v0,$a0,0 #init max pointer to first element

lw $v1,0($v0) #init max value to first value

addi $t0,$a0,0 #init next pointer to first

loop:

beq $t0,$a1,ret #if next=last, return

addi $t0,$t0,4 #advance to next element

lw $t1,0($t0) #load next element into $t1

slt $t2,$t1,$v1 #(next)<(max) ?

bne $t2,$zero,loop #if (next)<(max), repeat

addi $v0,$t0,0 #next element is new max element

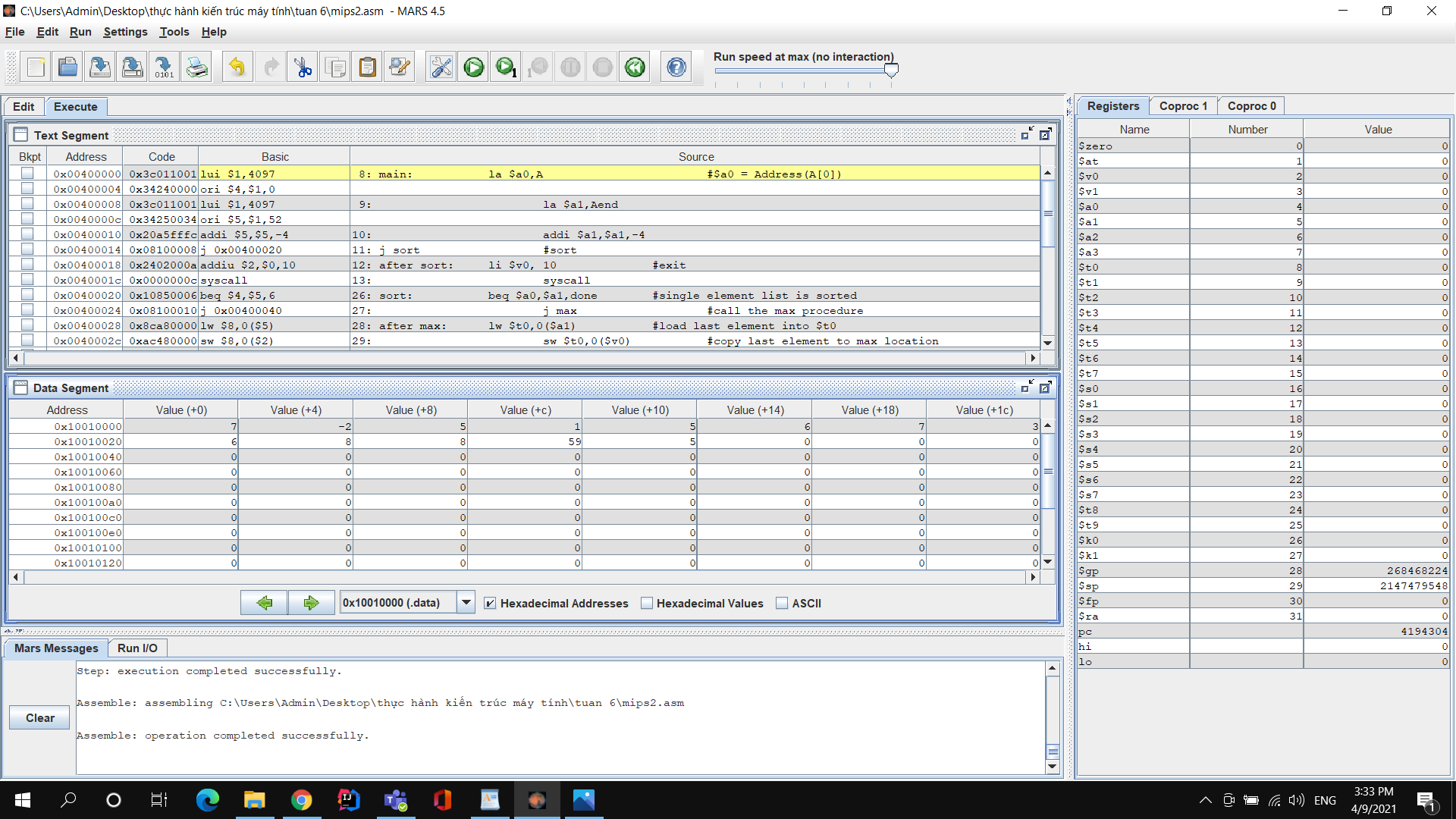
addi $v1,$t1,0 #next value is new max value

j loop #change completed; now repeat

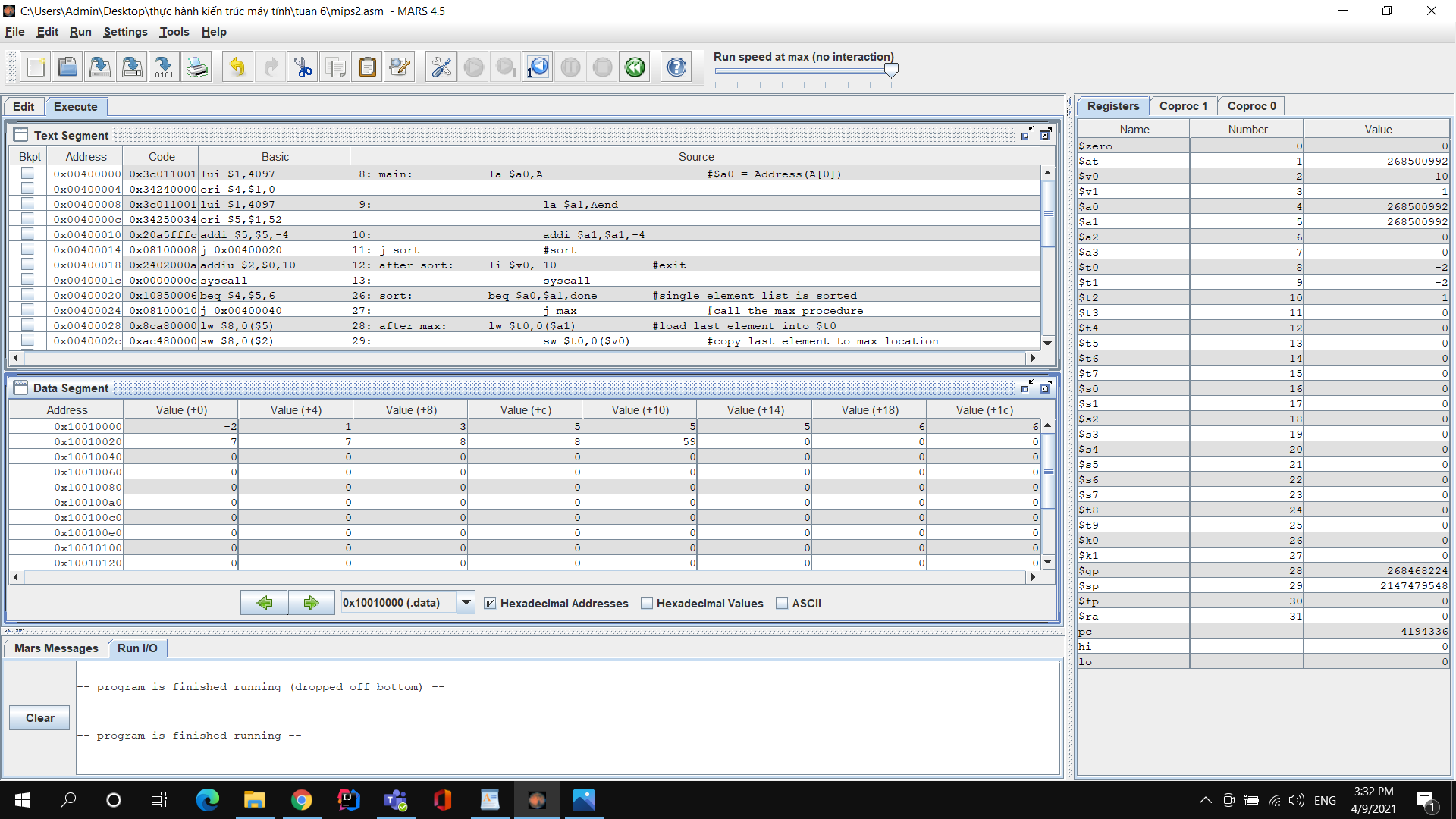
ret:

j after\_max

**kết quả trước khi sắp xếp:**

****

**kết quả sau khi sắp xếp:**

****

**Assignment 3:**

**Code:**

.data

A: .word 7, -2, 5, 1, 5, 6, 7, 3, 6, 8, 8, 59, 5

Aend: .word

.text

main: la $a0,A #$a0 = Address(A[0])

la $a1,Aend

addi $a1,$a1,-4 #$a1 = Address(A[n-1])

j sort #sort

after\_sort: li $v0, 10 #exit

syscall

end\_main:

sort: beq $a0,$a1,done #single element list is sorted

j max #call the max procedure

after\_loop:

addi $a1,$a1,-4 #decrement pointer to last element

j sort #repeat sort for smaller list

done: j after\_sort

max:

addi $t0,$a0,0 #init pointer to first

loop:

beq $t0,$a1,after\_loop #if j = last or n - i -1, return

addi $t1,$t0,0 # init j

addi $t0,$t0,4 # init j + 1

lw $t3,0($t1) #load j element into $t3

lw $t4,0($t0) #load j+1 element into $t4

slt $t5,$t4,$t3 #(A[j+1])<(A[j]) ?

bne $t5,$zero,swap #if A[j+1])<(A[j]), repeat

j loop #if not, repeat

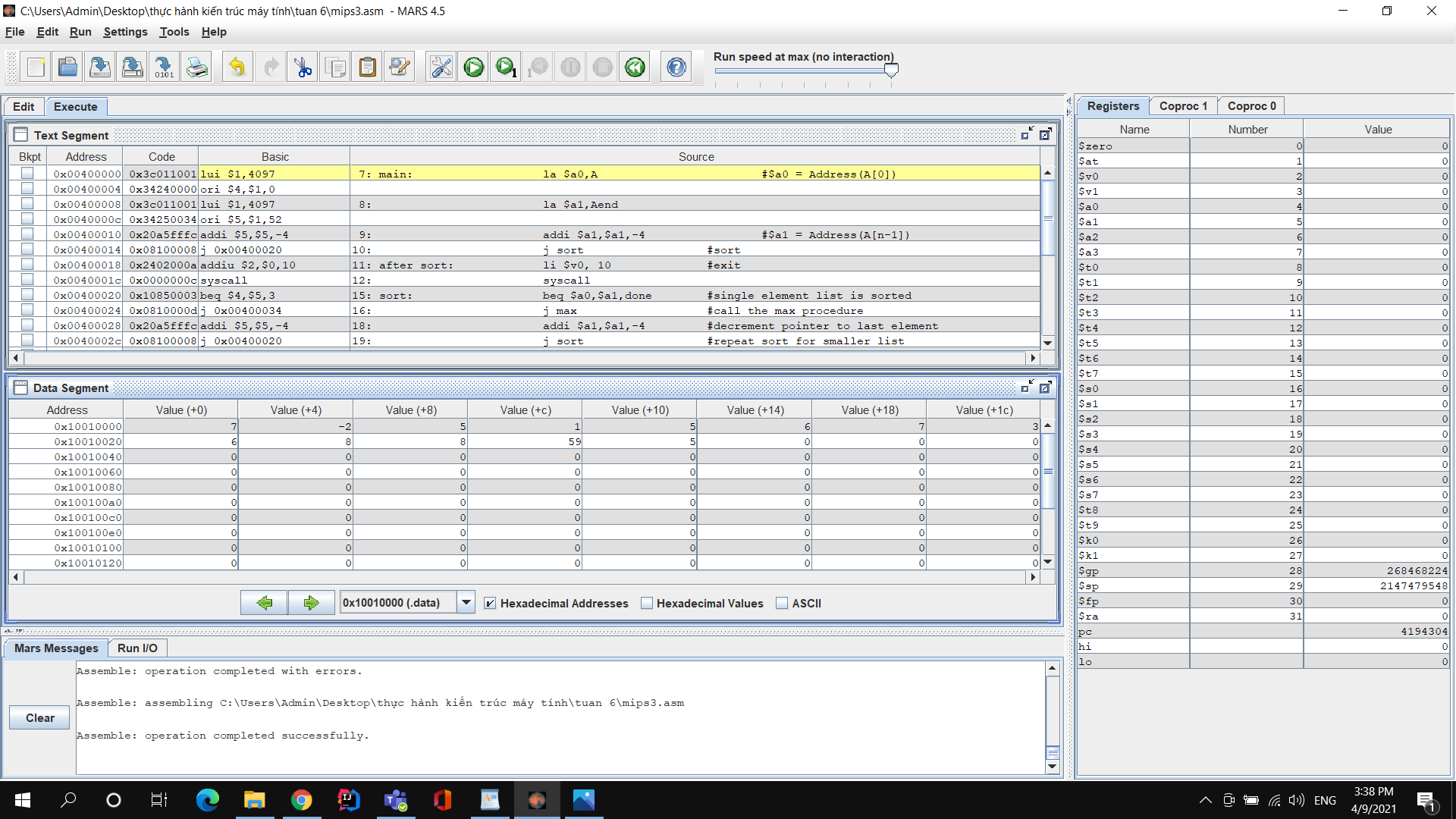
swap:

sw $t3,0($t0)

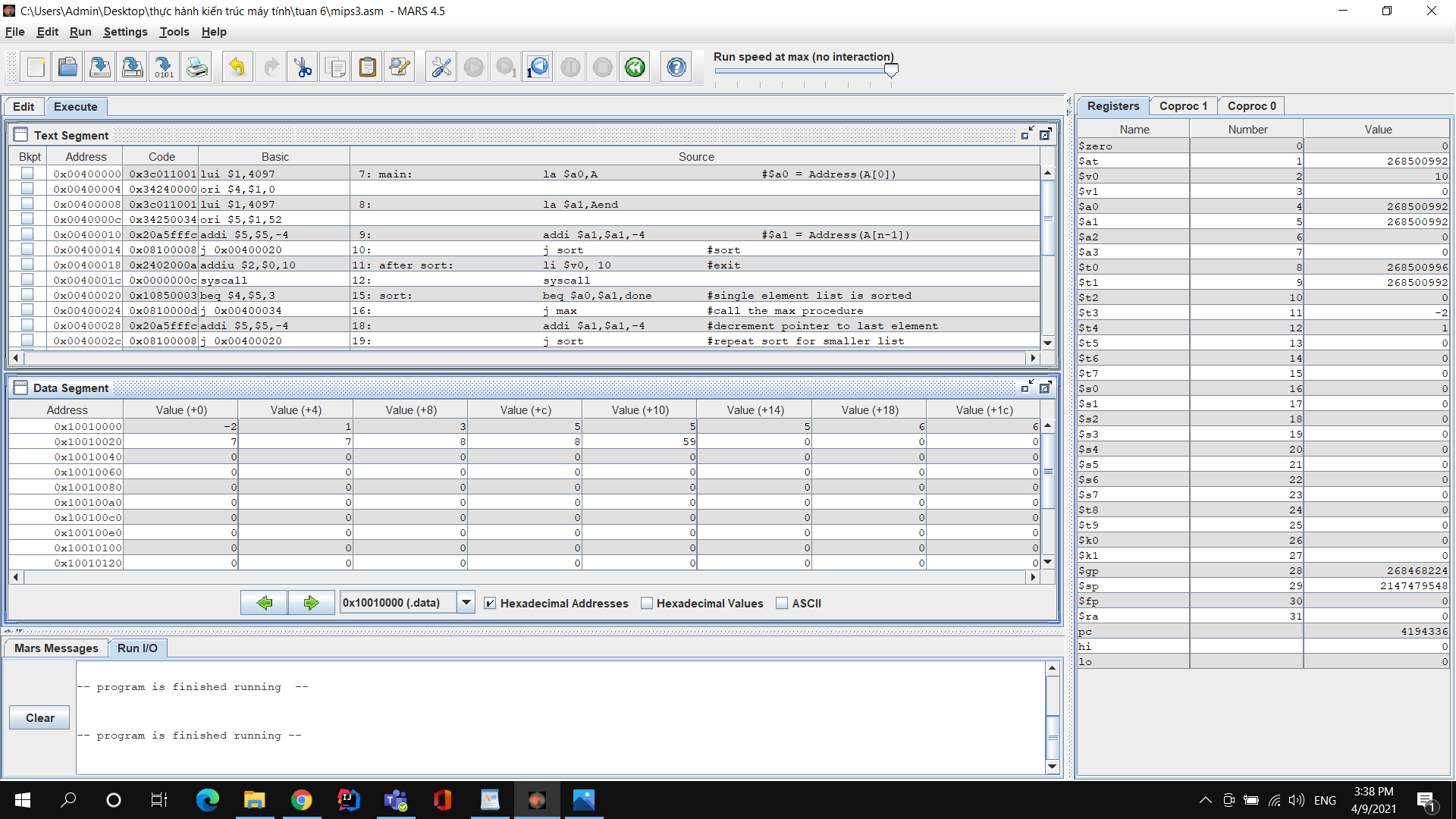
sw $t4,0($t1)

j loop

**kết quả trước khi sắp xếp:**

****

**kết quả sau khi sắp xếp:**

****

**Assignment 4:**

**Code:**

.data

A: .word 7, -2, 5, 1, 5, 6, 7, 3, 6, 8, 8, 59, 5

Aend: .word

.text

main: la $a0,A #$a0 = Address(A[0])

la $a1,Aend # see sort

addi $a2,$a0,4 #$a2 = i = A[1] cuz i run from 1

j sort #sort

after\_sort: li $v0, 10 #exit

syscall

end\_main:

sort: beq $a2,$a1,done # i = n ? or i > n - 1

j loop #call the loop procedure

after\_loop:

addi $a2,$a2,4 #increment pointer of i

j sort #repeat sort

done: j after\_sort

loop:

addi $t0,$a2,-4 #j = i -1

lw $t1,0($a2) # $t1 = key = A[i]

condition1:

slt $t2,$t0,$a0

bne $t2,$0,ret

condition2:

lw $t3,0($t0) #$t3 = A[j]

slt $t2,$t1,$t3 #key < A[j]

beq $t2,$0,ret # if not, ret

sw $t3,4($t0) # A[j+1] = A[j]

addi $t0,$t0,-4 # j = j -1

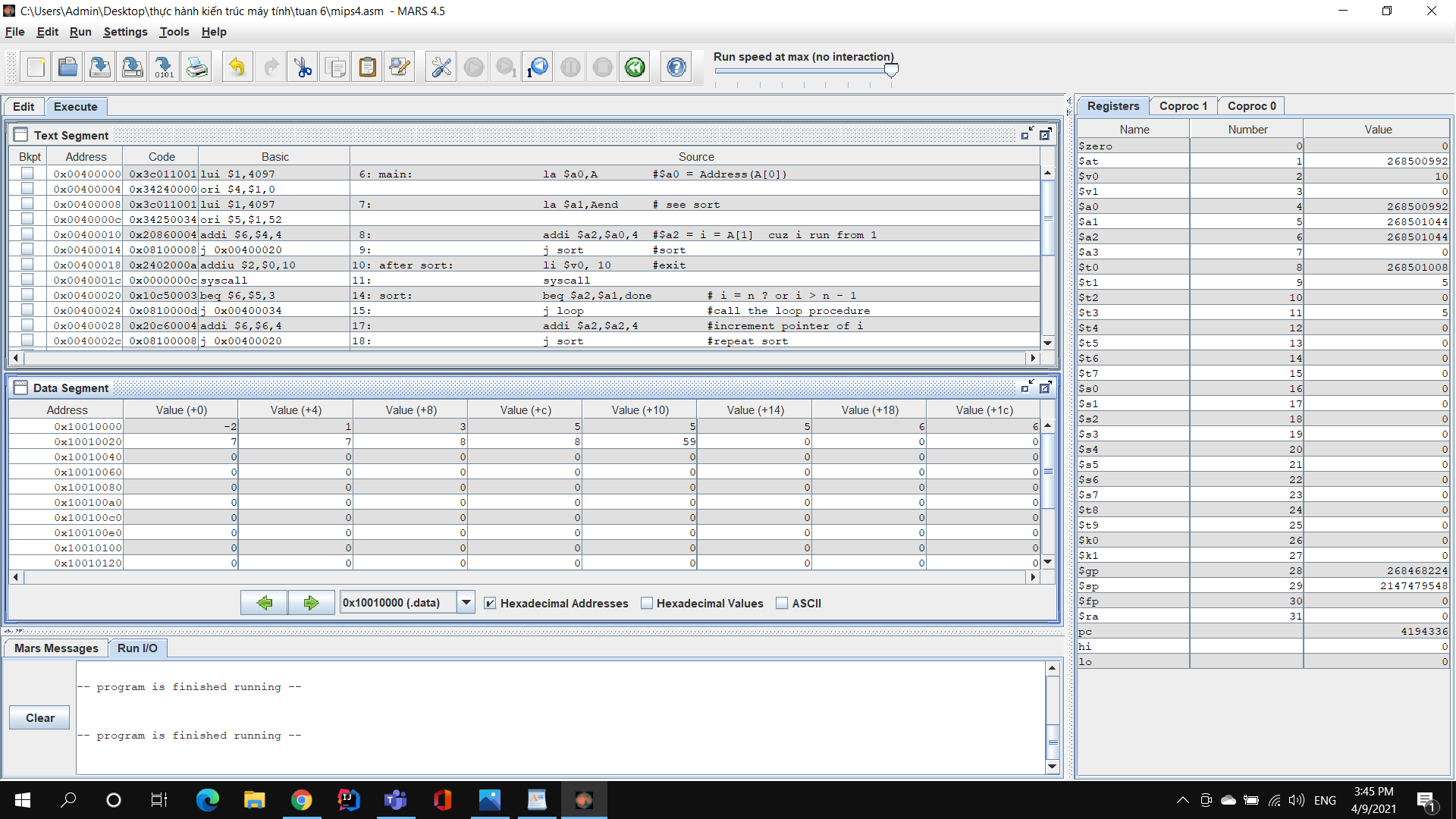
j condition1 #repeat while loop

ret:

sw $t1,4($t0) # A[j+1] = key

j after\_loop

**kết quả sau khi sắp xếp :**

****

**Insertion Sort**

⦁ $a0: địa chỉ mảng A

⦁ $a1: địa chỉ sau phần tử cuối mảng A

⦁ $a2: giá trị i

⦁ $t1: địa chỉ A[i]

⦁ $t3: địa chỉ A[j]