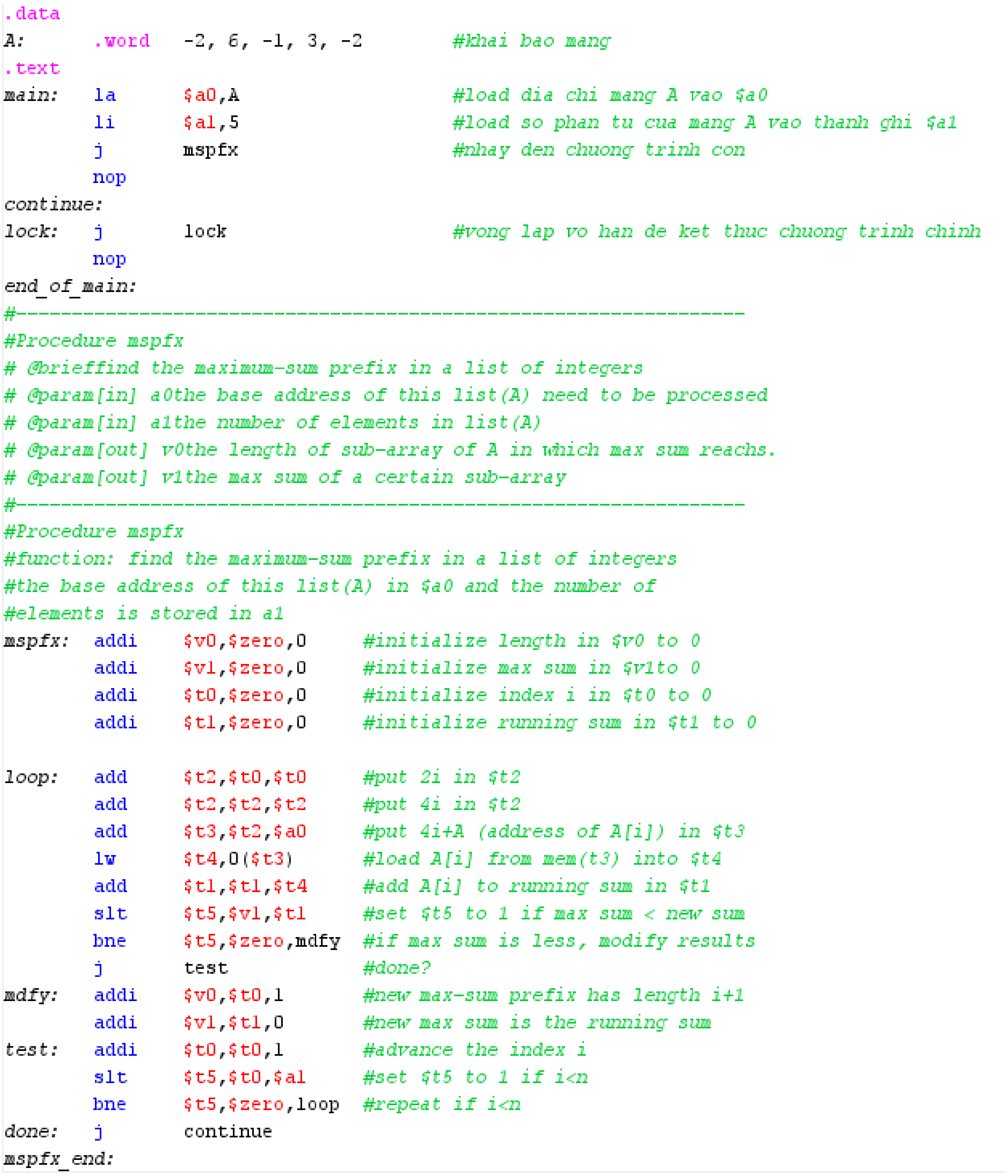
BÁO CÁO TUẦN 6

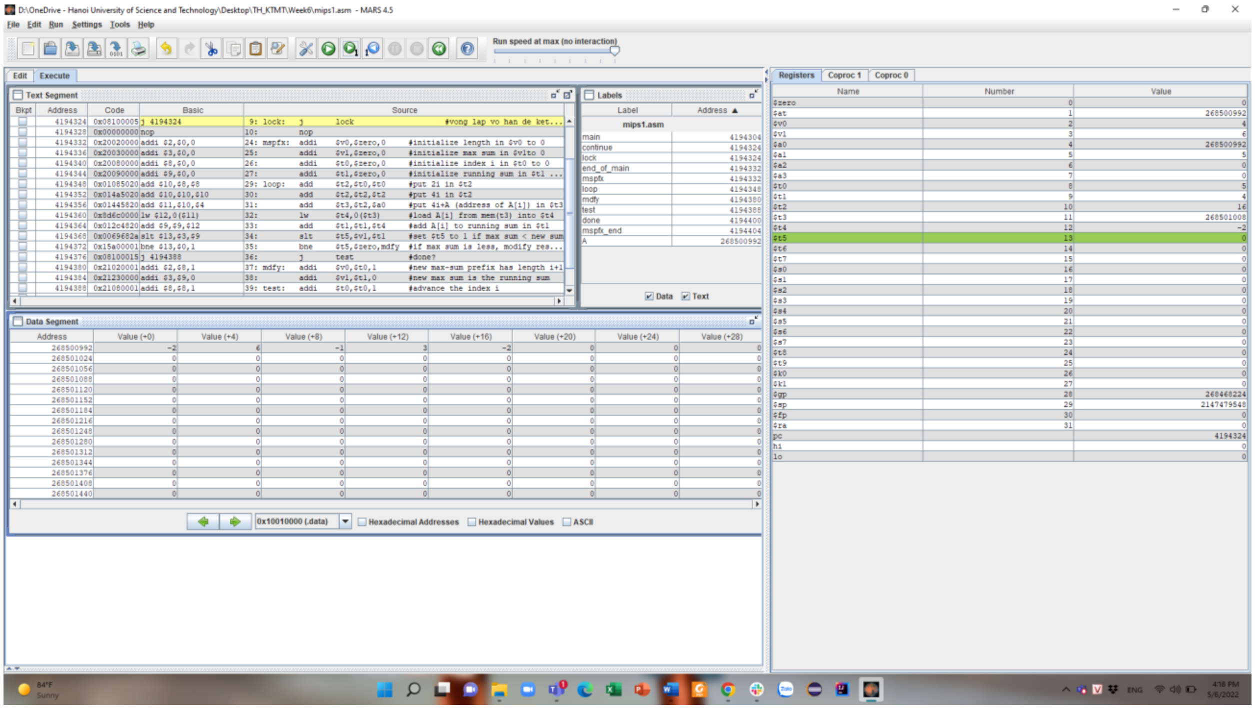
Exercise 1:

- *Chương trình:*



```
.data
A:      .word   -2, 6, -1, 3, -2        #khai bao mang
.text
main:   la      $a0,A                   #load dia chi mang A vao $a0
        li      $a1,5                   #load so phan tu cua mang A vao thanh ghi $a1
        j       mspfx                   #nhay den chuong trinh con
        nop
continue:
lock:   j       lock                    #vong lap vo han de ket thuc chuong trinh chinh
        nop
end_of_main:
#-----------------------------------------------------------------
#Procedure mspfx
# @brieffind the maximum-sum prefix in a list of integers
# @param[in] a0the base address of this list(A) need to be processed
# @param[in] a1the number of elements in list(A)
# @param[out] v0the length of sub-array of A in which max sum reachs.
# @param[out] v1the 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 $v1to 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:*



D:\OneDrive - Hanoi University of Science and Technology\Desktop\TH_KTMT\Week6\mips1.asm - MARS 4.5

File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Labels
Data Segment
Registers Coproc 1 Coproc 0

Bkpt	Address	Code	Basic	Source
□	4194324	0x08100005	j 4194324	9: lock: j lock #vong lap vo han de ket...
□	4194328	0x00000000	nop	10: nop
□	4194332	0x20020000	addi $2,$0,0	24: mspfx: addi $v0,$zero,0 #initialize length in $v0 to 0
□	4194336	0x20030000	addi $3,$0,0	25: addi $v1,$zero,0 #initialize max sum in $v1to 0
□	4194340	0x20080000	addi $8,$0,0	26: addi $t0,$zero,0 #initialize index i in $t0 to 0
□	4194344	0x20090000	addi $9,$0,0	27: addi $t1,$zero,0 #initialize running sum in $t1 ...
□	4194348	0x01085020	add $10,$8,$8	29: loop: add $t2,$t0,$t0 #put 2i in $t2
□	4194352	0x014a5020	add $10,$10,$10	30: add $t2,$t2,$t2 #put 4i in $t2
□	4194356	0x01445820	add $11,$10,$4	31: add $t3,$t2,$a0 #put 4i+A (address of A[i]) in $t3
□	4194360	0x8d6c0000	lw $12,0($11)	32: lw $t4,0($t3) #load A[i] from mem(t3) into $t4
□	4194364	0x012c4820	add $9,$9,$12	33: add $t1,$t1,$t4 #add A[i] to running sum in $t1
□	4194368	0x0069682a	slt $13,$3,$9	34: slt $t5,$v1,$t1 #set $t5 to 1 if max sum < new sum
□	4194372	0x15a00001	bne $13,$0,1	35: bne $t5,$zero,mdfy #if max sum is less, modify res...
□	4194376	0x08100015	j 4194388	36: j test #done?
□	4194380	0x21020001	addi $2,$8,1	37: mdfy: addi $v0,$t0,1 #new max-sum prefix has length i+1
□	4194384	0x21230000	addi $3,$9,0	38: addi $v1,$t1,0 #new max sum is the running sum
□	4194388	0x21080001	addi $8,$8,1	39: test: addi $t0,$t0,1 #advance the index i

Label	Address ▲
mips1.asm	
main	4194304
continue	4194324
lock	4194324
end_of_main	4194332
mspfx	4194332
loop	4194348
mdfy	4194380
test	4194388
done	4194400
mspfx_end	4194404
A	268500992

☑ Data ☑ Text

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	-2	6	-1	3	-2	0	0	0
268501024	0	0	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0
268501440	0	0	0	0	0	0	0	0

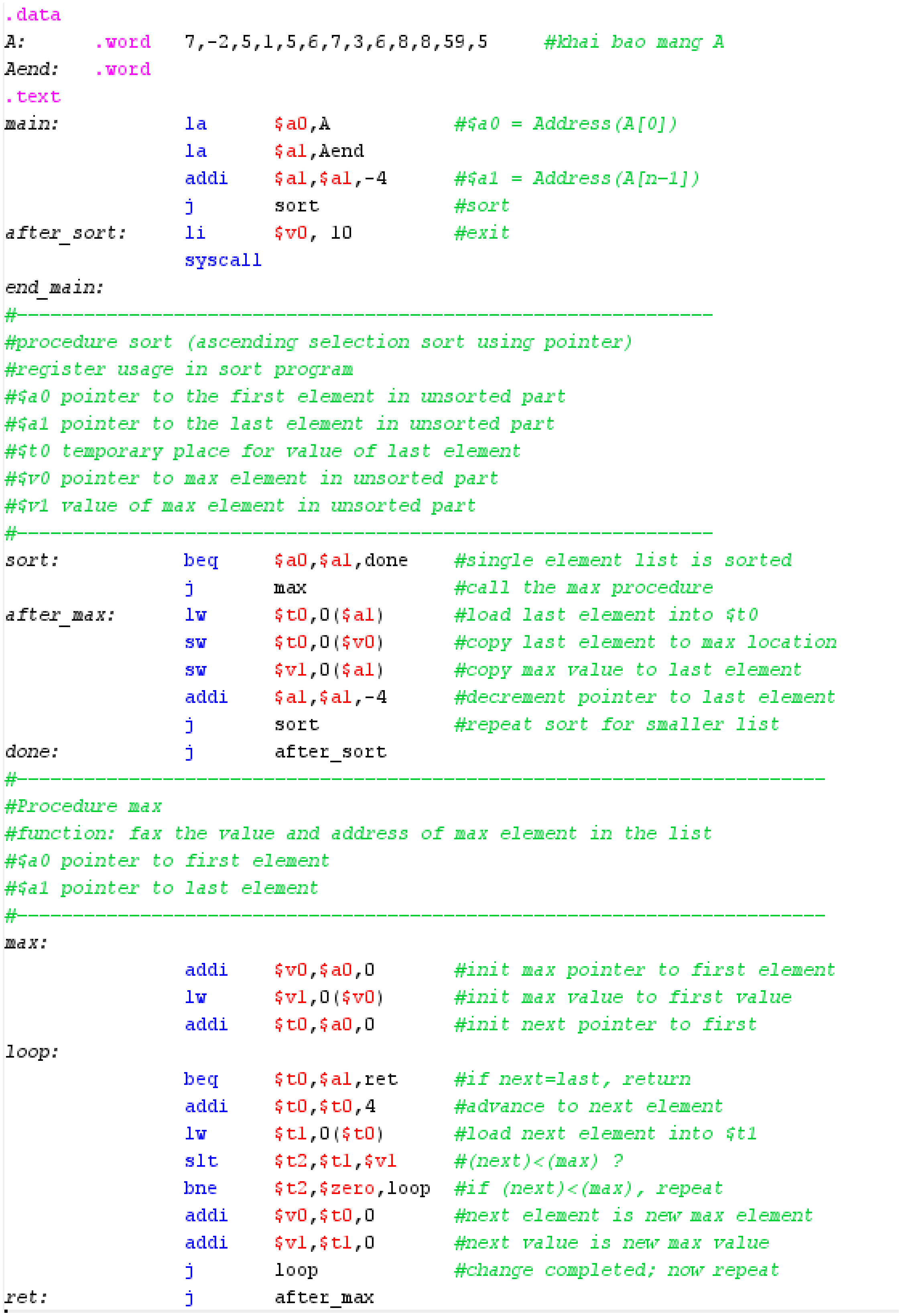
0x10010000 (.data) □ Hexadecimal Addresses □ Hexadecimal Values □ ASCII

Name	Number	Value
$zero	0	0
$at	1	268500992
$v0	2	4
$v1	3	6
$a0	4	268500992
$a1	5	5
$a2	6	0
$a3	7	0
$t0	8	5
$t1	9	4
$t2	10	16
$t3	11	268501008
$t4	12	-2
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	0
$s2	18	0
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194324
hi		0
lo		0

Exercise 2:

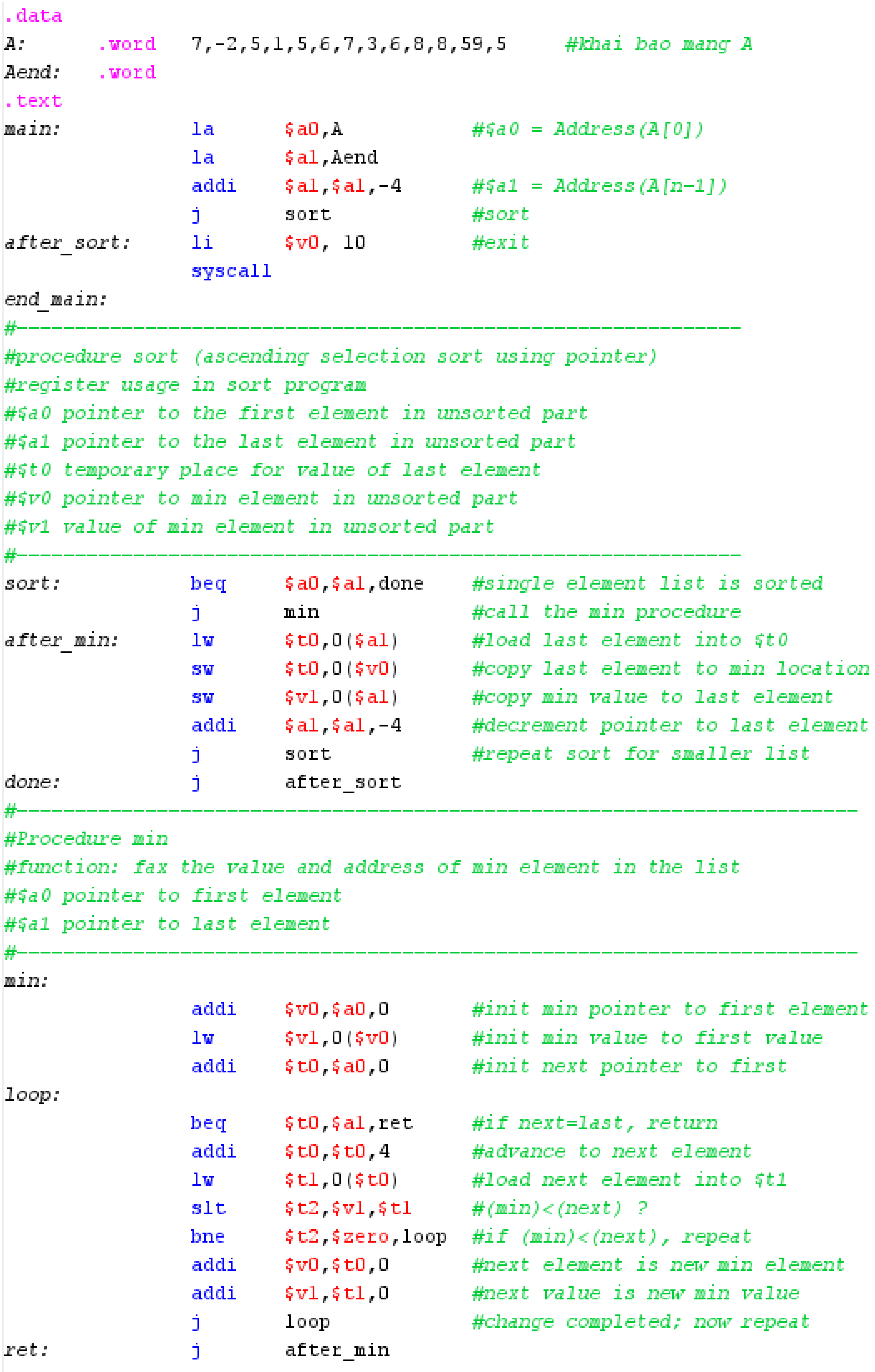
- *Chương trình:*

  + Sắp xếp tăng dần:



```
.data
A:      .word   7,-2,5,1,5,6,7,3,6,8,8,59,5     #khai bao mang A
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      $v0, 10         #exit
                syscall
end_main:
#---------------------------------------------------------------
#procedure sort (ascending selection sort using pointer)
#register usage in sort program
#$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
```

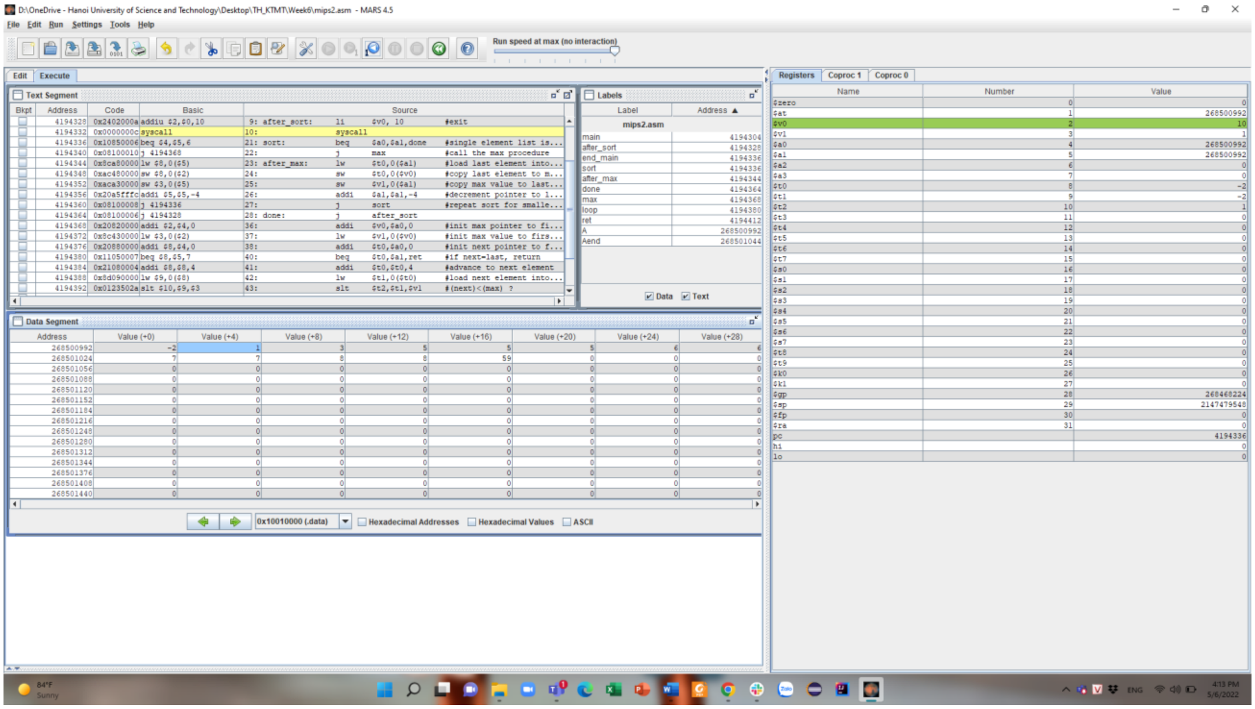
+ Sắp xếp giảm dần:



```
.data
A:      .word   7,-2,5,1,5,6,7,3,6,8,8,59,5     #khai bao mang A
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      $v0, 10         #exit
                syscall
end_main:
#---------------------------------------------------------------
#procedure sort (ascending selection sort using pointer)
#register usage in sort program
#$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
#$v0 pointer to min element in unsorted part
#$v1 value of min element in unsorted part
#---------------------------------------------------------------
sort:           beq     $a0,$a1,done    #single element list is sorted
                j       min             #call the min procedure
after_min:      lw      $t0,0($a1)      #load last element into $t0
                sw      $t0,0($v0)      #copy last element to min location
                sw      $v1,0($a1)      #copy min 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 min
#function: fax the value and address of min element in the list
#$a0 pointer to first element
#$a1 pointer to last element
#-------------------------------------------------------------------------
min:
                addi    $v0,$a0,0       #init min pointer to first element
                lw      $v1,0($v0)      #init min 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     $t2,$v1,$t1     #(min)<(next) ?
                bne     $t2,$zero,loop  #if (min)<(next), repeat
                addi    $v0,$t0,0       #next element is new min element
                addi    $v1,$t1,0       #next value is new min value
                j       loop            #change completed; now repeat
ret:            j       after_min
```

- *Kết quả:*

  + Sắp xếp tăng dần:



D:\OneDrive - Hanoi University of Science and Technology\Desktop\TH_KTMT\Week6\mips2.asm - MARS 4.5

File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source			
□	4194328	0x2402000a	addiu $2,$0,10	9: after_sort:	li	$v0, 10	#exit
□	4194332	0x0000000c	syscall	10:	syscall		
□	4194336	0x10850006	beq $4,$5,6	21: sort:	beq	$a0,$a1,done	#single element list is...
□	4194340	0x08100010	j 4194368	22:	j	max	#call the max procedure
□	4194344	0x8ca80000	lw $8,0($5)	23: after_max:	lw	$t0,0($a1)	#load last element into...
□	4194348	0xac480000	sw $8,0($2)	24:	sw	$t0,0($v0)	#copy last element to m...
□	4194352	0xaca30000	sw $3,0($5)	25:	sw	$v1,0($a1)	#copy max value to last...
□	4194356	0x20a5fffc	addi $5,$5,-4	26:	addi	$a1,$a1,-4	#decrement pointer to l...
□	4194360	0x08100008	j 4194336	27:	j	sort	#repeat sort for smalle...
□	4194364	0x08100006	j 4194328	28: done:	j	after_sort	
□	4194368	0x20820000	addi $2,$4,0	36:	addi	$v0,$a0,0	#init max pointer to fi...
□	4194372	0x8c430000	lw $3,0($2)	37:	lw	$v1,0($v0)	#init max value to firs...
□	4194376	0x20880000	addi $8,$4,0	38:	addi	$t0,$a0,0	#init next pointer to f...
□	4194380	0x11050007	beq $8,$5,7	40:	beq	$t0,$a1,ret	#if next=last, return
□	4194384	0x21080004	addi $8,$8,4	41:	addi	$t0,$t0,4	#advance to next element
□	4194388	0x8d090000	lw $9,0($8)	42:	lw	$t1,0($t0)	#load next element into...
□	4194392	0x0123502a	slt $10,$9,$3	43:	slt	$t2,$t1,$v1	#(next)<(max) ?

Labels

Label	Address ▲
mips2.asm	
main	4194304
after_sort	4194328
end_main	4194336
sort	4194336
after_max	4194344
done	4194364
max	4194368
loop	4194380
ret	4194412
A	268500992
Aend	268501044

☑ Data ☑ Text

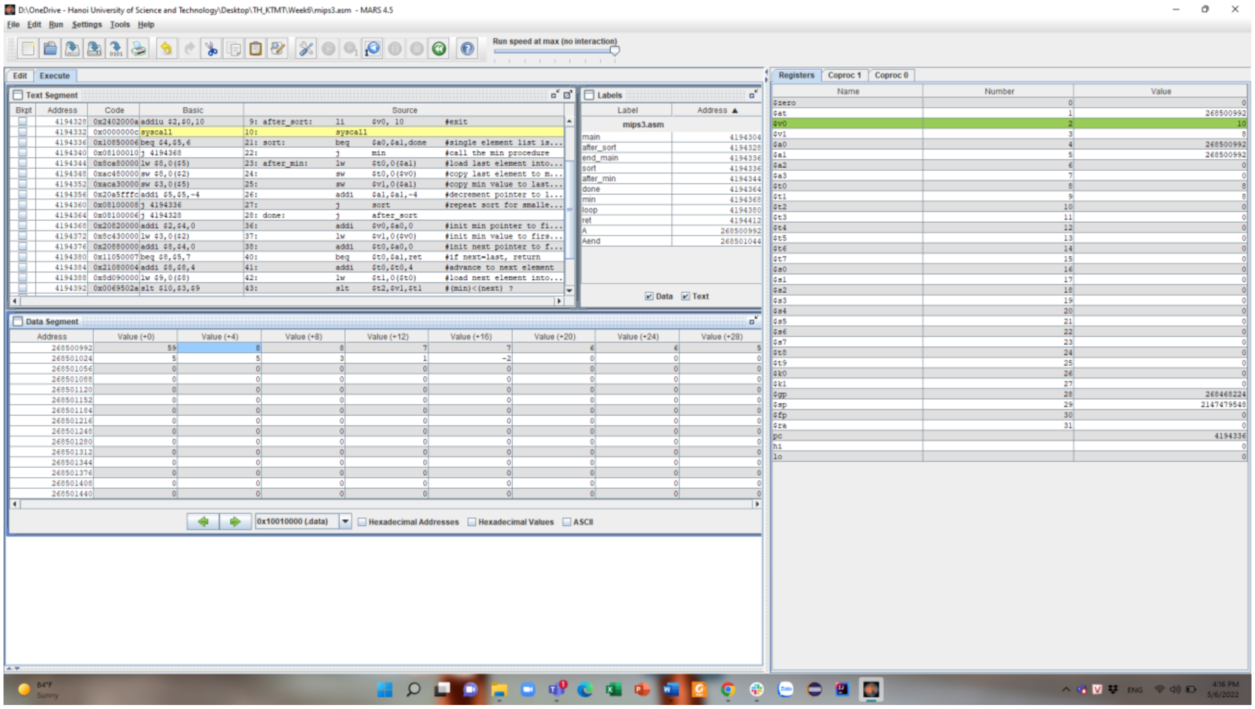
Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	-2	1	3	5	5	5	6	6
268501024	7	7	8	8	59	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0
268501440	0	0	0	0	0	0	0	0

0x10010000 (.data) □ Hexadecimal Addresses □ Hexadecimal Values □ ASCII

Name	Number	Value
$zero	0	0
$at	1	268500992
$v0	2	10
$v1	3	1
$a0	4	268500992
$a1	5	268500992
$a2	6	0
$a3	7	0
$t0	8	-2
$t1	9	-2
$t2	10	1
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	0
$s2	18	0
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194336
hi		0
lo		0

  + Sắp xếp giảm dần:



D:\OneDrive - Hanoi University of Science and Technology\Desktop\TH_KTMT\Week6\mips3.asm - MARS 4.5

File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source			
□	4194328	0x2402000a	addiu $2,$0,10	9: after_sort:	li	$v0, 10	#exit
□	4194332	0x0000000c	syscall	10:	syscall		
□	4194336	0x10850006	beq $4,$5,6	21: sort:	beq	$a0,$a1,done	#single element list is...
□	4194340	0x08100010	j 4194368	22:	j	min	#call the min procedure
□	4194344	0x8ca80000	lw $8,0($5)	23: after_min:	lw	$t0,0($a1)	#load last element into...
□	4194348	0xac480000	sw $8,0($2)	24:	sw	$t0,0($v0)	#copy last element to m...
□	4194352	0xaca30000	sw $3,0($5)	25:	sw	$v1,0($a1)	#copy min value to last...
□	4194356	0x20a5fffc	addi $5,$5,-4	26:	addi	$a1,$a1,-4	#decrement pointer to l...
□	4194360	0x08100008	j 4194336	27:	j	sort	#repeat sort for smalle...
□	4194364	0x08100006	j 4194328	28: done:	j	after_sort	
□	4194368	0x20820000	addi $2,$4,0	36:	addi	$v0,$a0,0	#init min pointer to fi...
□	4194372	0x8c430000	lw $3,0($2)	37:	lw	$v1,0($v0)	#init min value to firs...
□	4194376	0x20880000	addi $8,$4,0	38:	addi	$t0,$a0,0	#init next pointer to f...
□	4194380	0x11050007	beq $8,$5,7	40:	beq	$t0,$a1,ret	#if next=last, return
□	4194384	0x21080004	addi $8,$8,4	41:	addi	$t0,$t0,4	#advance to next element
□	4194388	0x8d090000	lw $9,0($8)	42:	lw	$t1,0($t0)	#load next element into...
□	4194392	0x0069502a	slt $10,$3,$9	43:	slt	$t2,$v1,$t1	#(min)<(next) ?

Labels

Label	Address ▲
mips3.asm	
main	4194304
after_sort	4194328
end_main	4194336
sort	4194336
after_min	4194344
done	4194364
min	4194368
loop	4194380
ret	4194412
A	268500992
Aend	268501044

☑ Data ☑ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	59	8	8	7	7	6	6	5
268501024	5	5	3	1	-2	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0
268501440	0	0	0	0	0	0	0	0

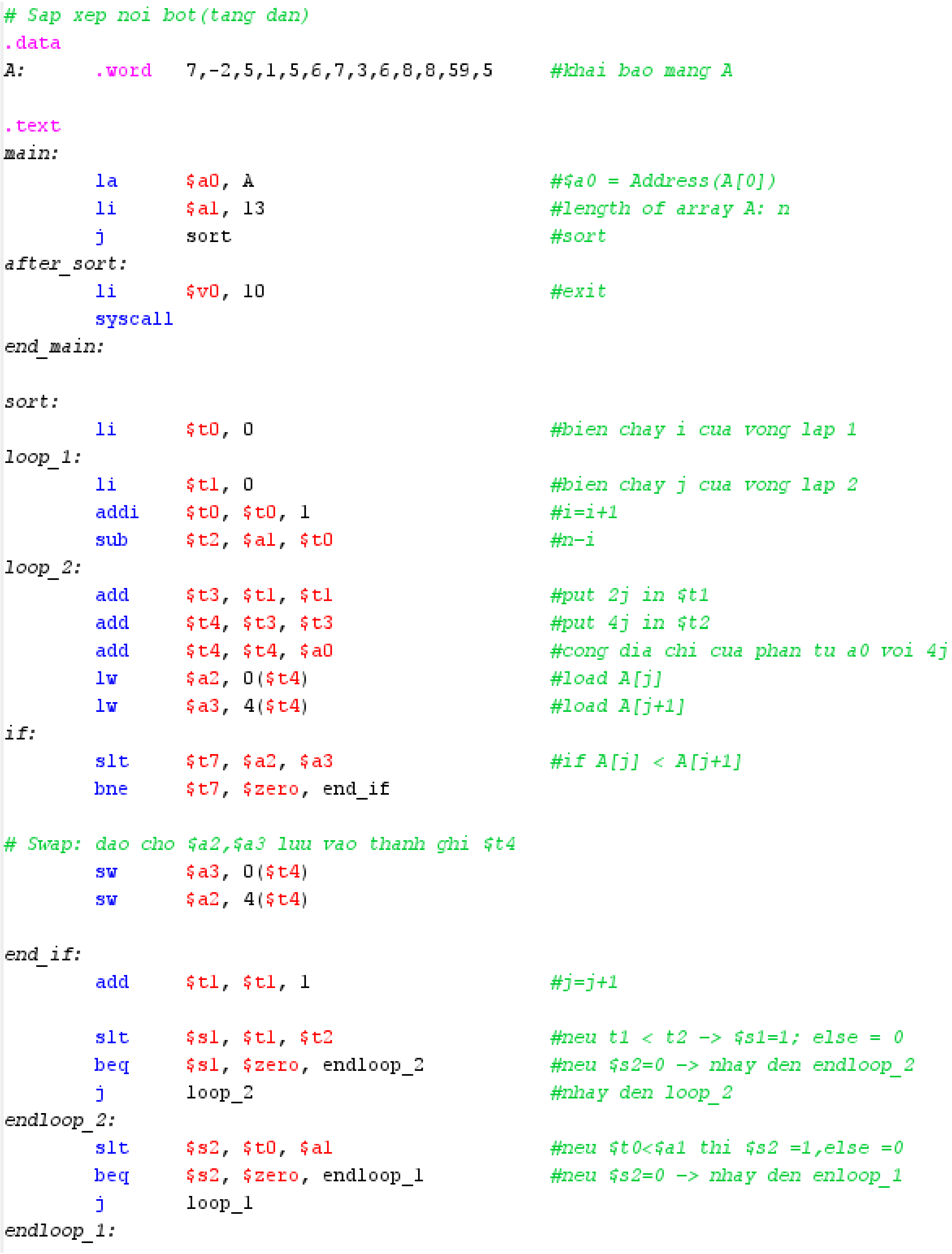
0x10010000 (.data) □ Hexadecimal Addresses □ Hexadecimal Values □ ASCII

Name	Number	Value
$zero	0	0
$at	1	268500992
$v0	2	10
$v1	3	8
$a0	4	268500992
$a1	5	268500992
$a2	6	0
$a3	7	0
$t0	8	8
$t1	9	8
$t2	10	0
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	0
$s2	18	0
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194336
hi		0
lo		0

Exercise 3:

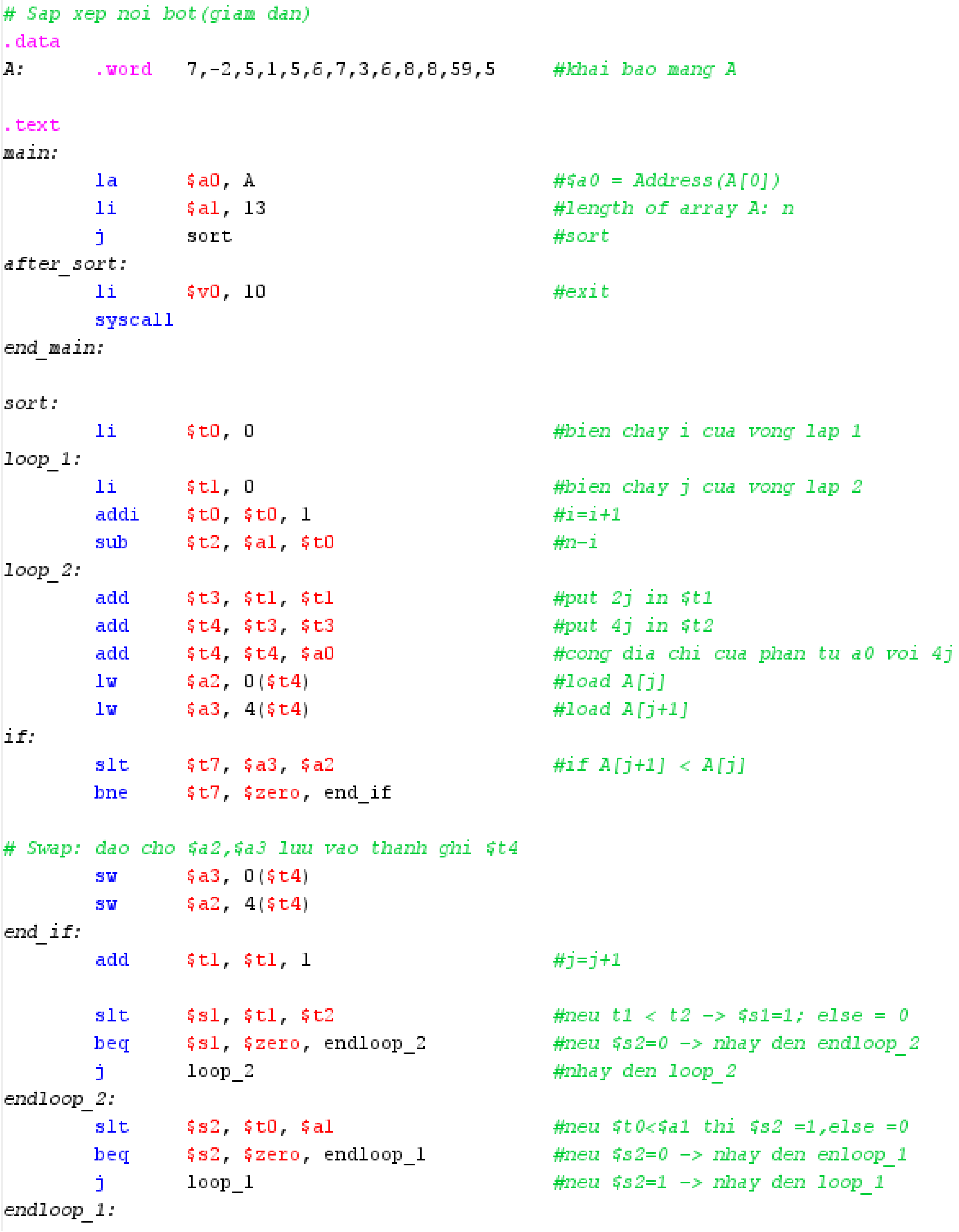
- *Chương trình:*

  + Sắp xếp tăng dần:



```
# Sap xep noi bot(tang dan)
.data
A:      .word   7,-2,5,1,5,6,7,3,6,8,8,59,5     #khai bao mang A

.text
main:
        la      $a0, A                          #$a0 = Address(A[0])
        li      $a1, 13                         #length of array A: n
        j       sort                            #sort
after_sort:
        li      $v0, 10                         #exit
        syscall
end_main:

sort:
        li      $t0, 0                          #bien chay i cua vong lap 1
loop_1:
        li      $t1, 0                          #bien chay j cua vong lap 2
        addi    $t0, $t0, 1                     #i=i+1
        sub     $t2, $a1, $t0                   #n-i
loop_2:
        add     $t3, $t1, $t1                   #put 2j in $t1
        add     $t4, $t3, $t3                   #put 4j in $t2
        add     $t4, $t4, $a0                   #cong dia chi cua phan tu a0 voi 4j
        lw      $a2, 0($t4)                     #load A[j]
        lw      $a3, 4($t4)                     #load A[j+1]
if:
        slt     $t7, $a2, $a3                   #if A[j] < A[j+1]
        bne     $t7, $zero, end_if

# Swap: dao cho $a2,$a3 luu vao thanh ghi $t4
        sw      $a3, 0($t4)
        sw      $a2, 4($t4)

end_if:
        add     $t1, $t1, 1                     #j=j+1

        slt     $s1, $t1, $t2                   #neu t1 < t2 -> $s1=1; else = 0
        beq     $s1, $zero, endloop_2           #neu $s2=0 -> nhay den endloop_2
        j       loop_2                          #nhay den loop_2
endloop_2:
        slt     $s2, $t0, $a1                   #neu $t0<$a1 thi $s2 =1,else =0
        beq     $s2, $zero, endloop_1           #neu $s2=0 -> nhay den enloop_1
        j       loop_1
endloop_1:
```

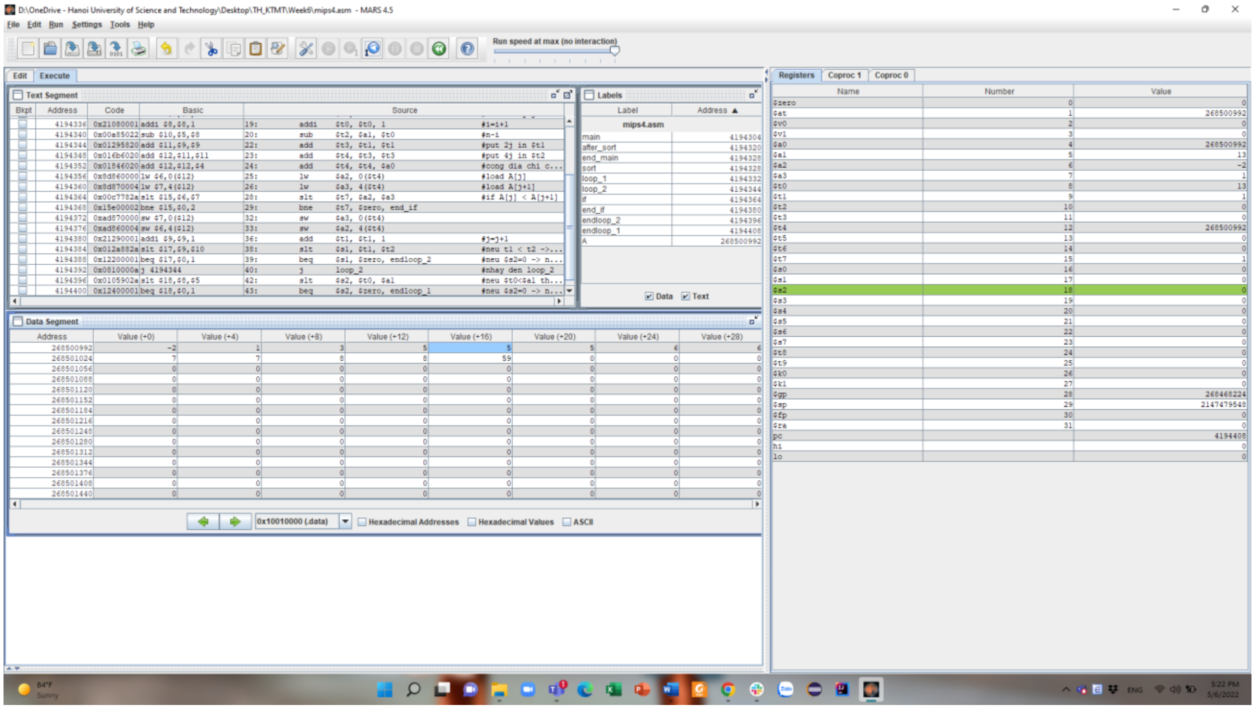
+ Sắp xếp giảm dần:



```
# Sap xep noi bot(giam dan)
.data
A:      .word   7,-2,5,1,5,6,7,3,6,8,8,59,5     #khai bao mang A

.text
main:
        la      $a0, A                          #$a0 = Address(A[0])
        li      $a1, 13                         #length of array A: n
        j       sort                            #sort
after_sort:
        li      $v0, 10                         #exit
        syscall
end_main:

sort:
        li      $t0, 0                          #bien chay i cua vong lap 1
loop_1:
        li      $t1, 0                          #bien chay j cua vong lap 2
        addi    $t0, $t0, 1                     #i=i+1
        sub     $t2, $a1, $t0                   #n-i
loop_2:
        add     $t3, $t1, $t1                   #put 2j in $t1
        add     $t4, $t3, $t3                   #put 4j in $t2
        add     $t4, $t4, $a0                   #cong dia chi cua phan tu a0 voi 4j
        lw      $a2, 0($t4)                     #load A[j]
        lw      $a3, 4($t4)                     #load A[j+1]
if:
        slt     $t7, $a3, $a2                   #if A[j+1] < A[j]
        bne     $t7, $zero, end_if

# Swap: dao cho $a2,$a3 luu vao thanh ghi $t4
        sw      $a3, 0($t4)
        sw      $a2, 4($t4)
end_if:
        add     $t1, $t1, 1                     #j=j+1

        slt     $s1, $t1, $t2                   #neu t1 < t2 -> $s1=1; else = 0
        beq     $s1, $zero, endloop_2           #neu $s2=0 -> nhay den endloop_2
        j       loop_2                          #nhay den loop_2
endloop_2:
        slt     $s2, $t0, $a1                   #neu $t0<$a1 thi $s2 =1,else =0
        beq     $s2, $zero, endloop_1           #neu $s2=0 -> nhay den enloop_1
        j       loop_1                          #neu $s2=1 -> nhay den loop_1
endloop_1:
```

- *Kết quả:*

  + Sắp xếp tăng dần:



D:\OneDrive - Hanoi University of Science and Technology\Desktop\TH_KTMT\Week6\mips4.asm - MARS 4.5

File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Labels
Data Segment
Registers Coproc 1 Coproc 0

```
Bkpt  Address  Code        Basic                 Source
      4194336  0x21080001  addi $8,$8,1          19:   addi  $t0, $t0, 1             #i=i+1
      4194340  0x00a85022  sub $10,$5,$8         20:   sub   $t2, $a1, $t0           #n-i
      4194344  0x01295820  add $11,$9,$9         22:   add   $t3, $t1, $t1           #put 2j in $t1
      4194348  0x016b6020  add $12,$11,$11       23:   add   $t4, $t3, $t3           #put 4j in $t2
      4194352  0x01846020  add $12,$12,$4        24:   add   $t4, $t4, $a0           #cong dia chi c...
      4194356  0x8d860000  lw $6,0($12)          25:   lw    $a2, 0($t4)             #load A[j]
      4194360  0x8d870004  lw $7,4($12)          26:   lw    $a3, 4($t4)             #load A[j+1]
      4194364  0x00c7782a  slt $15,$6,$7         28:   slt   $t7, $a2, $a3           #if A[j] < A[j+1]
      4194368  0x15e00002  bne $15,$0,2          29:   bne   $t7, $zero, end_if
      4194372  0xad870000  sw $7,0($12)          32:   sw    $a3, 0($t4)
      4194376  0xad860004  sw $6,4($12)          33:   sw    $a2, 4($t4)
      4194380  0x21290001  addi $9,$9,1          36:   add   $t1, $t1, 1             #j=j+1
      4194384  0x012a882a  slt $17,$9,$10        38:   slt   $s1, $t1, $t2           #neu t1 < t2 ->...
      4194388  0x12200001  beq $17,$0,1          39:   beq   $s1, $zero, endloop_2   #neu $s2=0 -> n...
      4194392  0x0810000a  j 4194344             40:   j     loop_2                  #nhay den loop_2
      4194396  0x0105902a  slt $18,$8,$5         42:   slt   $s2, $t0, $a1           #neu $t0<$a1 th...
      4194400  0x12400001  beq $18,$0,1          43:   beq   $s2, $zero, endloop_1   #neu $s2=0 -> n...
```

Label	Address ▲
mips4.asm	
main	4194304
after_sort	4194320
end_main	4194328
sort	4194328
loop_1	4194332
loop_2	4194344
if	4194364
end_if	4194380
endloop_2	4194396
endloop_1	4194408
A	268500992

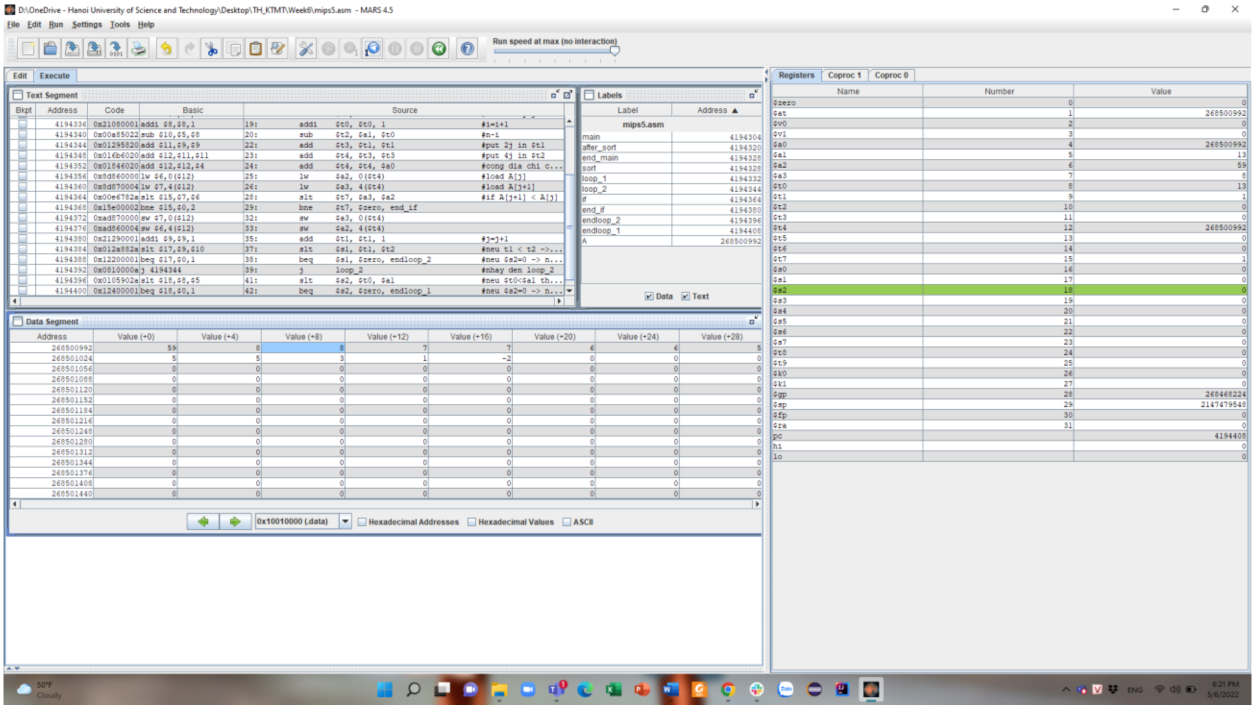
☑ Data ☑ Text

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	-2	1	3	5	5	5	6	6
268501024	7	7	8	8	59	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0
268501440	0	0	0	0	0	0	0	0

0x10010000 (.data) □ Hexadecimal Addresses □ Hexadecimal Values □ ASCII

Name	Number	Value
$zero	0	0
$at	1	268500992
$v0	2	0
$v1	3	0
$a0	4	268500992
$a1	5	13
$a2	6	-2
$a3	7	1
$t0	8	13
$t1	9	1
$t2	10	0
$t3	11	0
$t4	12	268500992
$t5	13	0
$t6	14	0
$t7	15	1
$s0	16	0
$s1	17	0
$s2	18	0
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194408
hi		0
lo		0

+ Sắp xếp giảm dần:



D:\OneDrive - Hanoi University of Science and Technology\Desktop\TH_KTMT\Week6\mips5.asm - MARS 4.5

File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Labels
Data Segment
Registers Coproc 1 Coproc 0

```
Bkpt  Address  Code        Basic                 Source
      4194336  0x21080001  addi $8,$8,1          19:   addi  $t0, $t0, 1             #i=i+1
      4194340  0x00a85022  sub $10,$5,$8         20:   sub   $t2, $a1, $t0           #n-i
      4194344  0x01295820  add $11,$9,$9         22:   add   $t3, $t1, $t1           #put 2j in $t1
      4194348  0x016b6020  add $12,$11,$11       23:   add   $t4, $t3, $t3           #put 4j in $t2
      4194352  0x01846020  add $12,$12,$4        24:   add   $t4, $t4, $a0           #cong dia chi c...
      4194356  0x8d860000  lw $6,0($12)          25:   lw    $a2, 0($t4)             #load A[j]
      4194360  0x8d870004  lw $7,4($12)          26:   lw    $a3, 4($t4)             #load A[j+1]
      4194364  0x00e6782a  slt $15,$7,$6         28:   slt   $t7, $a3, $a2           #if A[j+1] < A[j]
      4194368  0x15e00002  bne $15,$0,2          29:   bne   $t7, $zero, end_if
      4194372  0xad870000  sw $7,0($12)          32:   sw    $a3, 0($t4)
      4194376  0xad860004  sw $6,4($12)          33:   sw    $a2, 4($t4)
      4194380  0x21290001  addi $9,$9,1          35:   add   $t1, $t1, 1             #j=j+1
      4194384  0x012a882a  slt $17,$9,$10        37:   slt   $s1, $t1, $t2           #neu t1 < t2 ->...
      4194388  0x12200001  beq $17,$0,1          38:   beq   $s1, $zero, endloop_2   #neu $s2=0 -> n...
      4194392  0x0810000a  j 4194344             39:   j     loop_2                  #nhay den loop_2
      4194396  0x0105902a  slt $18,$8,$5         41:   slt   $s2, $t0, $a1           #neu $t0<$a1 th...
      4194400  0x12400001  beq $18,$0,1          42:   beq   $s2, $zero, endloop_1   #neu $s2=0 -> n...
```

Label	Address ▲
mips5.asm	
main	4194304
after_sort	4194320
end_main	4194328
sort	4194328
loop_1	4194332
loop_2	4194344
if	4194364
end_if	4194380
endloop_2	4194396
endloop_1	4194408
A	268500992

☑ Data ☑ Text

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	59	8	8	7	7	6	6	5
268501024	5	5	3	1	-2	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0
268501440	0	0	0	0	0	0	0	0

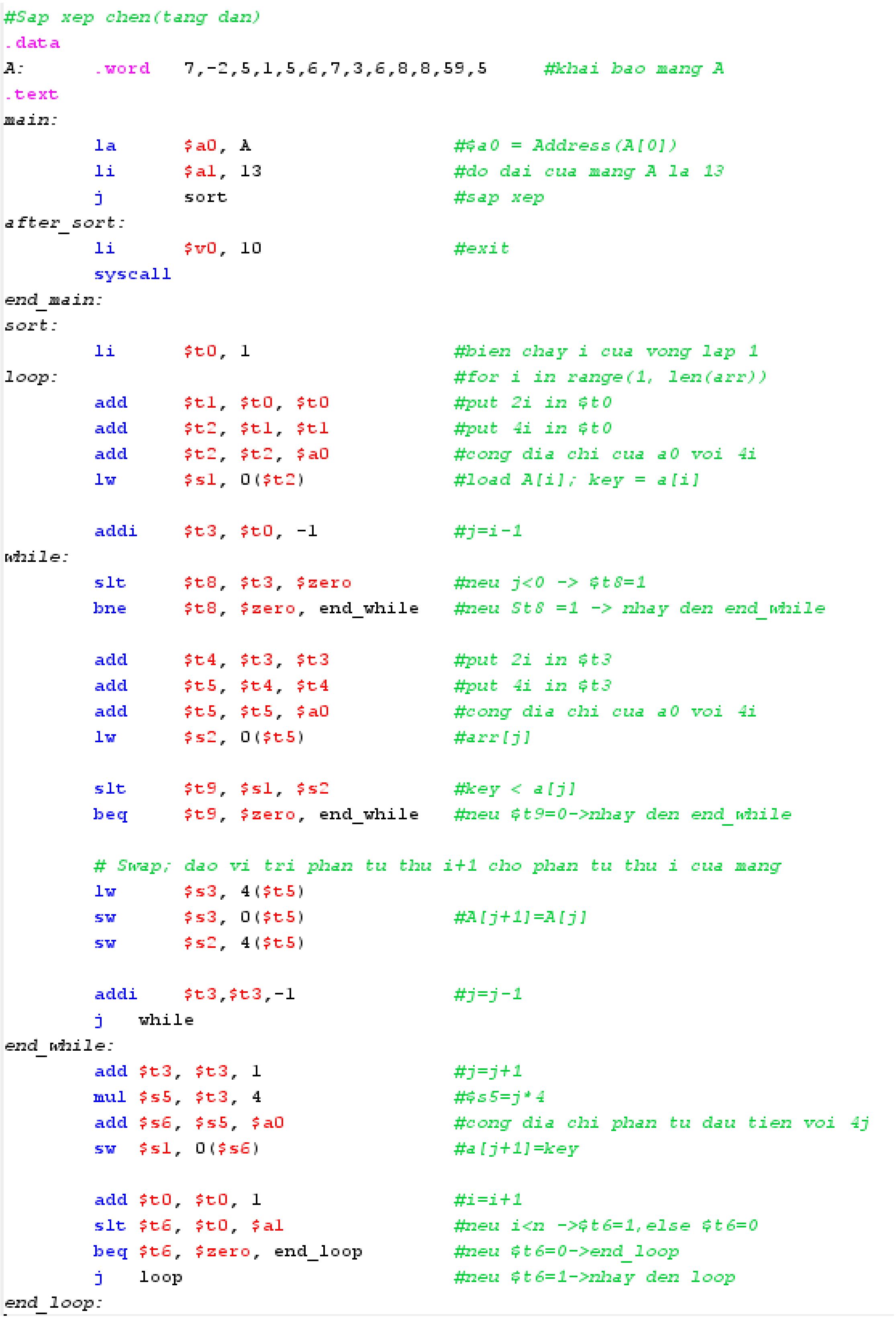
0x10010000 (.data) □ Hexadecimal Addresses □ Hexadecimal Values □ ASCII

Name	Number	Value
$zero	0	0
$at	1	268500992
$v0	2	0
$v1	3	0
$a0	4	268500992
$a1	5	13
$a2	6	59
$a3	7	8
$t0	8	13
$t1	9	1
$t2	10	0
$t3	11	0
$t4	12	268500992
$t5	13	0
$t6	14	0
$t7	15	1
$s0	16	0
$s1	17	0
$s2	18	0
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194408
hi		0
lo		0

Exercise 4:

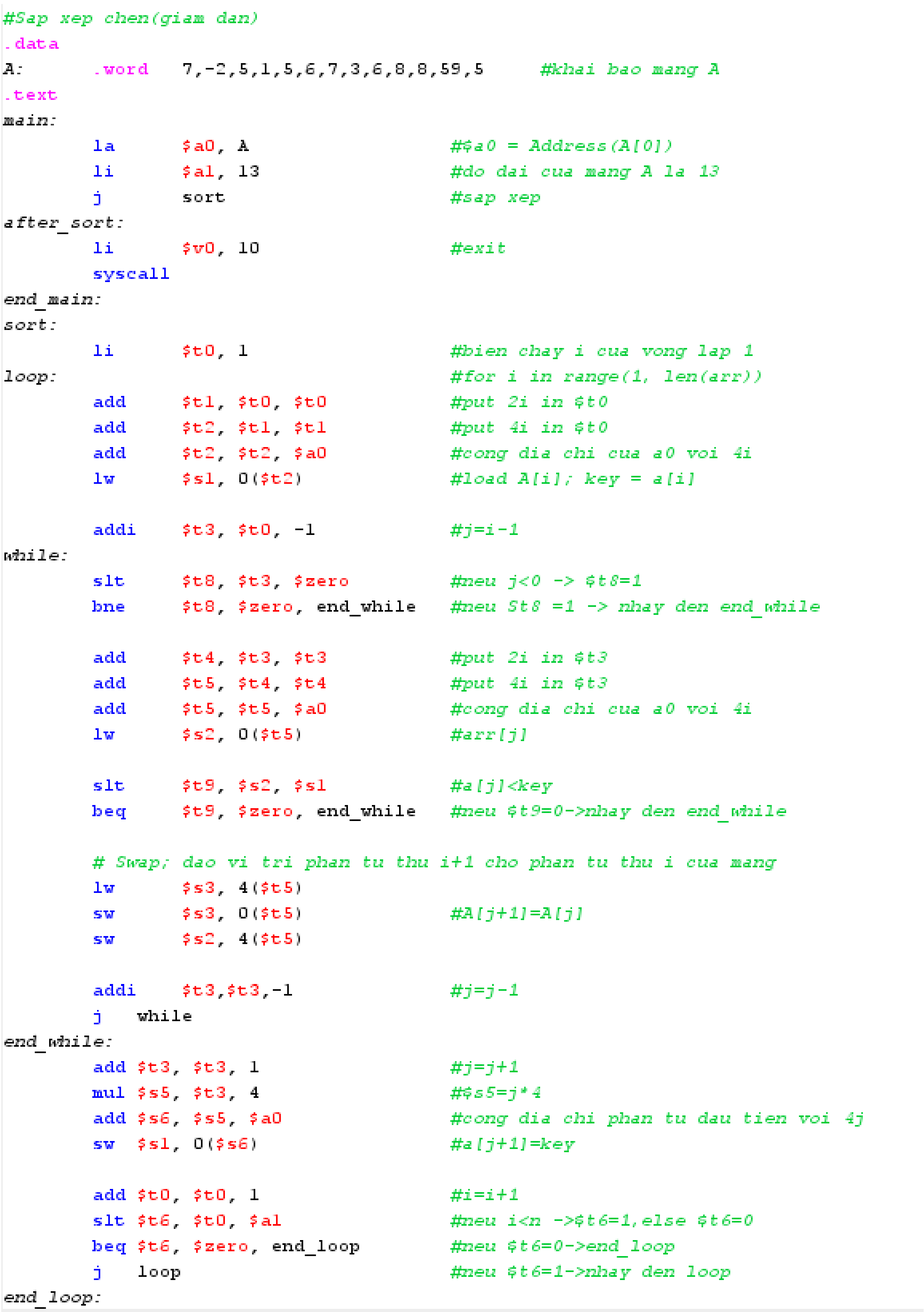
- *Chương trình:*

+ Sắp xếp tăng dần:



```
#Sap xep chen(tang dan)
.data
A:      .word   7,-2,5,1,5,6,7,3,6,8,8,59,5     #khai bao mang A
.text
main:
        la      $a0, A                  #$a0 = Address(A[0])
        li      $a1, 13                 #do dai cua mang A la 13
        j       sort                    #sap xep
after_sort:
        li      $v0, 10                 #exit
        syscall
end_main:
sort:
        li      $t0, 1                  #bien chay i cua vong lap 1
loop:                                   #for i in range(1, len(arr))
        add     $t1, $t0, $t0           #put 2i in $t0
        add     $t2, $t1, $t1           #put 4i in $t0
        add     $t2, $t2, $a0           #cong dia chi cua a0 voi 4i
        lw      $s1, 0($t2)             #load A[i]; key = a[i]

        addi    $t3, $t0, -1            #j=i-1
while:
        slt     $t8, $t3, $zero         #neu j<0 -> $t8=1
        bne     $t8, $zero, end_while   #neu St8 =1 -> nhay den end_while

        add     $t4, $t3, $t3           #put 2i in $t3
        add     $t5, $t4, $t4           #put 4i in $t3
        add     $t5, $t5, $a0           #cong dia chi cua a0 voi 4i
        lw      $s2, 0($t5)             #arr[j]

        slt     $t9, $s1, $s2           #key < a[j]
        beq     $t9, $zero, end_while   #neu $t9=0->nhay den end_while

        # Swap; dao vi tri phan tu thu i+1 cho phan tu thu i cua mang
        lw      $s3, 4($t5)
        sw      $s3, 0($t5)             #A[j+1]=A[j]
        sw      $s2, 4($t5)

        addi    $t3,$t3,-1              #j=j-1
        j   while
end_while:
        add $t3, $t3, 1                 #j=j+1
        mul $s5, $t3, 4                 #$s5=j*4
        add $s6, $s5, $a0               #cong dia chi phan tu dau tien voi 4j
        sw  $s1, 0($s6)                 #a[j+1]=key

        add $t0, $t0, 1                 #i=i+1
        slt $t6, $t0, $a1               #neu i<n ->$t6=1,else $t6=0
        beq $t6, $zero, end_loop        #neu $t6=0->end_loop
        j   loop                        #neu $t6=1->nhay den loop
end_loop:
```

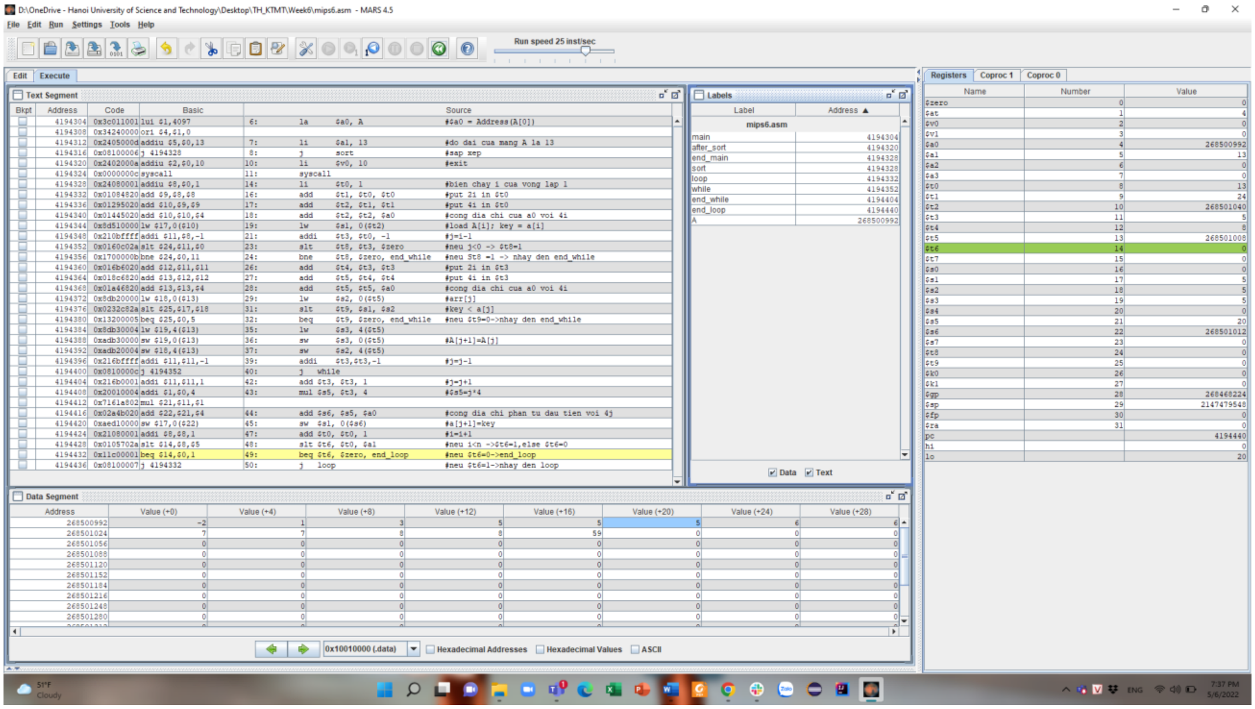
+ Sắp xếp giảm dần:



```
#Sap xep chen(giam dan)
.data
A:      .word   7,-2,5,1,5,6,7,3,6,8,8,59,5     #khai bao mang A
.text
main:
        la      $a0, A                  #$a0 = Address(A[0])
        li      $a1, 13                 #do dai cua mang A la 13
        j       sort                    #sap xep
after_sort:
        li      $v0, 10                 #exit
        syscall
end_main:
sort:
        li      $t0, 1                  #bien chay i cua vong lap 1
loop:                                   #for i in range(1, len(arr))
        add     $t1, $t0, $t0           #put 2i in $t0
        add     $t2, $t1, $t1           #put 4i in $t0
        add     $t2, $t2, $a0           #cong dia chi cua a0 voi 4i
        lw      $s1, 0($t2)             #load A[i]; key = a[i]

        addi    $t3, $t0, -1            #j=i-1
while:
        slt     $t8, $t3, $zero         #neu j<0 -> $t8=1
        bne     $t8, $zero, end_while   #neu St8 =1 -> nhay den end_while

        add     $t4, $t3, $t3           #put 2i in $t3
        add     $t5, $t4, $t4           #put 4i in $t3
        add     $t5, $t5, $a0           #cong dia chi cua a0 voi 4i
        lw      $s2, 0($t5)             #arr[j]

        slt     $t9, $s2, $s1           #a[j]<key
        beq     $t9, $zero, end_while   #neu $t9=0->nhay den end_while

        # Swap; dao vi tri phan tu thu i+1 cho phan tu thu i cua mang
        lw      $s3, 4($t5)
        sw      $s3, 0($t5)             #A[j+1]=A[j]
        sw      $s2, 4($t5)

        addi    $t3,$t3,-1              #j=j-1
        j   while
end_while:
        add $t3, $t3, 1                 #j=j+1
        mul $s5, $t3, 4                 #$s5=j*4
        add $s6, $s5, $a0               #cong dia chi phan tu dau tien voi 4j
        sw  $s1, 0($s6)                 #a[j+1]=key

        add $t0, $t0, 1                 #i=i+1
        slt $t6, $t0, $a1               #neu i<n ->$t6=1,else $t6=0
        beq $t6, $zero, end_loop        #neu $t6=0->end_loop
        j   loop                        #neu $t6=1->nhay den loop
end_loop:
```

- *Kết quả:*

  + Sắp xếp tăng dần:



D:\OneDrive - Hanoi University of Science and Technology\Desktop\TH_KTMT\Week6\mips6.asm - MARS 4.5

Bkpt	Address	Code	Basic		Source
☐	4194304	0x3c011001	lui $1,4097	6:	la $a0, A #$a0 = Address(A[0])
☐	4194308	0x34240000	ori $4,$1,0		
☐	4194312	0x2405000d	addiu $5,$0,13	7:	li $a1, 13 #do dai cua mang A la 13
☐	4194316	0x08100006	j 4194328	8:	j sort #sap xep
☐	4194320	0x2402000a	addiu $2,$0,10	10:	li $v0, 10 #exit
☐	4194324	0x0000000c	syscall	11:	syscall
☐	4194328	0x24080001	addiu $8,$0,1	14:	li $t0, 1 #bien chay i cua vong lap 1
☐	4194332	0x01084820	add $9,$8,$8	16:	add $t1, $t0, $t0 #put 2i in $t0
☐	4194336	0x01295020	add $10,$9,$9	17:	add $t2, $t1, $t1 #put 4i in $t0
☐	4194340	0x01445020	add $10,$10,$4	18:	add $t2, $t2, $a0 #cong dia chi cua a0 voi 4i
☐	4194344	0x8d510000	lw $17,0($10)	19:	lw $s1, 0($t2) #load A[i]; key = a[i]
☐	4194348	0x210bffff	addi $11,$8,-1	21:	addi $t3, $t0, -1 #j=i-1
☐	4194352	0x0160c02a	slt $24,$11,$0	23:	slt $t8, $t3, $zero #neu j<0 -> $t8=1
☐	4194356	0x1700000b	bne $24,$0,11	24:	bne $t8, $zero, end_while #neu St8 =1 -> nhay den end_while
☐	4194360	0x016b6020	add $12,$11,$11	26:	add $t4, $t3, $t3 #put 2i in $t3
☐	4194364	0x018c6820	add $13,$12,$12	27:	add $t5, $t4, $t4 #put 4i in $t3
☐	4194368	0x01a46820	add $13,$13,$4	28:	add $t5, $t5, $a0 #cong dia chi cua a0 voi 4i
☐	4194372	0x8db20000	lw $18,0($13)	29:	lw $s2, 0($t5) #arr[j]
☐	4194376	0x0232c82a	slt $25,$17,$18	31:	slt $t9, $s1, $s2 #key < a[j]
☐	4194380	0x13200005	beq $25,$0,5	32:	beq $t9, $zero, end_while #neu $t9=0->nhay den end_while
☐	4194384	0x8db30004	lw $19,4($13)	35:	lw $s3, 4($t5)
☐	4194388	0xadb30000	sw $19,0($13)	36:	sw $s3, 0($t5) #A[j+1]=A[j]
☐	4194392	0xadb20004	sw $18,4($13)	37:	sw $s2, 4($t5)
☐	4194396	0x216bffff	addi $11,$11,-1	39:	addi $t3,$t3,-1 #j=j-1
☐	4194400	0x0810000c	j 4194352	40:	j while
☐	4194404	0x216b0001	addi $11,$11,1	42:	add $t3, $t3, 1 #j=j+1
☐	4194408	0x20010004	addi $1,$0,4	43:	mul $s5, $t3, 4 #$s5=j*4
☐	4194412	0x7161a802	mul $21,$11,$1		
☐	4194416	0x02a4b020	add $22,$21,$4	44:	add $s6, $s5, $a0 #cong dia chi phan tu dau tien voi 4j
☐	4194420	0xaed10000	sw $17,0($22)	45:	sw $s1, 0($s6) #a[j+1]=key
☐	4194424	0x21080001	addi $8,$8,1	47:	add $t0, $t0, 1 #i=i+1
☐	4194428	0x0105702a	slt $14,$8,$5	48:	slt $t6, $t0, $a1 #neu i<n ->$t6=1,else $t6=0
☐	4194432	0x11c00001	beq $14,$0,1	49:	beq $t6, $zero, end_loop #neu $t6=0->end_loop
☐	4194436	0x08100007	j 4194332	50:	j loop #neu $t6=1->nhay den loop

Label	Address ▲
mips6.asm	
main	4194304
after_sort	4194320
end_main	4194328
sort	4194328
loop	4194332
while	4194352
end_while	4194404
end_loop	4194440
A	268500992

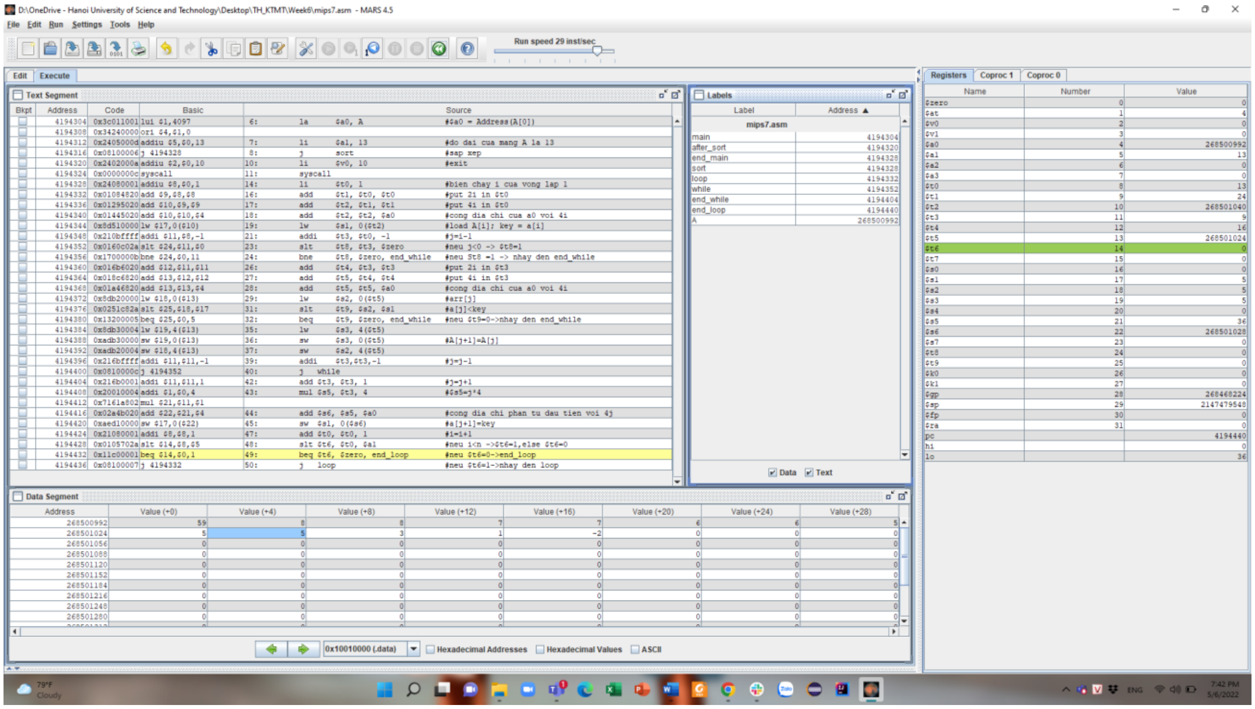
☑ Data ☑ Text

Name	Number	Value
$zero	0	0
$at	1	4
$v0	2	0
$v1	3	0
$a0	4	268500992
$a1	5	13
$a2	6	0
$a3	7	0
$t0	8	13
$t1	9	24
$t2	10	268501040
$t3	11	5
$t4	12	8
$t5	13	268501008
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	5
$s2	18	5
$s3	19	5
$s4	20	0
$s5	21	20
$s6	22	268501012
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194440
hi		0
lo		20

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	-2	1	3	5	5	5	6	6
268501024	7	7	8	8	59	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0

0x10010000 (.data) ☐ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

  + Sắp xếp giảm dần:



D:\OneDrive - Hanoi University of Science and Technology\Desktop\TH_KTMT\Week6\mips7.asm - MARS 4.5

Bkpt	Address	Code	Basic		Source
☐	4194304	0x3c011001	lui $1,4097	6:	la $a0, A #$a0 = Address(A[0])
☐	4194308	0x34240000	ori $4,$1,0		
☐	4194312	0x2405000d	addiu $5,$0,13	7:	li $a1, 13 #do dai cua mang A la 13
☐	4194316	0x08100006	j 4194328	8:	j sort #sap xep
☐	4194320	0x2402000a	addiu $2,$0,10	10:	li $v0, 10 #exit
☐	4194324	0x0000000c	syscall	11:	syscall
☐	4194328	0x24080001	addiu $8,$0,1	14:	li $t0, 1 #bien chay i cua vong lap 1
☐	4194332	0x01084820	add $9,$8,$8	16:	add $t1, $t0, $t0 #put 2i in $t0
☐	4194336	0x01295020	add $10,$9,$9	17:	add $t2, $t1, $t1 #put 4i in $t0
☐	4194340	0x01445020	add $10,$10,$4	18:	add $t2, $t2, $a0 #cong dia chi cua a0 voi 4i
☐	4194344	0x8d510000	lw $17,0($10)	19:	lw $s1, 0($t2) #load A[i]; key = a[i]
☐	4194348	0x210bffff	addi $11,$8,-1	21:	addi $t3, $t0, -1 #j=i-1
☐	4194352	0x0160c02a	slt $24,$11,$0	23:	slt $t8, $t3, $zero #neu j<0 -> $t8=1
☐	4194356	0x1700000b	bne $24,$0,11	24:	bne $t8, $zero, end_while #neu St8 =1 -> nhay den end_while
☐	4194360	0x016b6020	add $12,$11,$11	26:	add $t4, $t3, $t3 #put 2i in $t3
☐	4194364	0x018c6820	add $13,$12,$12	27:	add $t5, $t4, $t4 #put 4i in $t3
☐	4194368	0x01a46820	add $13,$13,$4	28:	add $t5, $t5, $a0 #cong dia chi cua a0 voi 4i
☐	4194372	0x8db20000	lw $18,0($13)	29:	lw $s2, 0($t5) #arr[j]
☐	4194376	0x0251c82a	slt $25,$18,$17	31:	slt $t9, $s2, $s1 #a[j]<key
☐	4194380	0x13200005	beq $25,$0,5	32:	beq $t9, $zero, end_while #neu $t9=0->nhay den end_while
☐	4194384	0x8db30004	lw $19,4($13)	35:	lw $s3, 4($t5)
☐	4194388	0xadb30000	sw $19,0($13)	36:	sw $s3, 0($t5) #A[j+1]=A[j]
☐	4194392	0xadb20004	sw $18,4($13)	37:	sw $s2, 4($t5)
☐	4194396	0x216bffff	addi $11,$11,-1	39:	addi $t3,$t3,-1 #j=j-1
☐	4194400	0x0810000c	j 4194352	40:	j while
☐	4194404	0x216b0001	addi $11,$11,1	42:	add $t3, $t3, 1 #j=j+1
☐	4194408	0x20010004	addi $1,$0,4	43:	mul $s5, $t3, 4 #$s5=j*4
☐	4194412	0x7161a802	mul $21,$11,$1		
☐	4194416	0x02a4b020	add $22,$21,$4	44:	add $s6, $s5, $a0 #cong dia chi phan tu dau tien voi 4j
☐	4194420	0xaed10000	sw $17,0($22)	45:	sw $s1, 0($s6) #a[j+1]=key
☐	4194424	0x21080001	addi $8,$8,1	47:	add $t0, $t0, 1 #i=i+1
☐	4194428	0x0105702a	slt $14,$8,$5	48:	slt $t6, $t0, $a1 #neu i<n ->$t6=1,else $t6=0
☐	4194432	0x11c00001	beq $14,$0,1	49:	beq $t6, $zero, end_loop #neu $t6=0->end_loop
☐	4194436	0x08100007	j 4194332	50:	j loop #neu $t6=1->nhay den loop

Label	Address ▲
mips7.asm	
main	4194304
after_sort	4194320
end_main	4194328
sort	4194328
loop	4194332
while	4194352
end_while	4194404
end_loop	4194440
A	268500992

☑ Data ☑ Text

Name	Number	Value
$zero	0	0
$at	1	4
$v0	2	0
$v1	3	0
$a0	4	268500992
$a1	5	13
$a2	6	0
$a3	7	0
$t0	8	13
$t1	9	24
$t2	10	268501040
$t3	11	9
$t4	12	16
$t5	13	268501024
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	5
$s2	18	5
$s3	19	5
$s4	20	0
$s5	21	36
$s6	22	268501028
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194440
hi		0
lo		36

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	59	8	8	7	7	6	6	5
268501024	5	5	3	1	-2	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0

0x10010000 (.data) ☐ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII