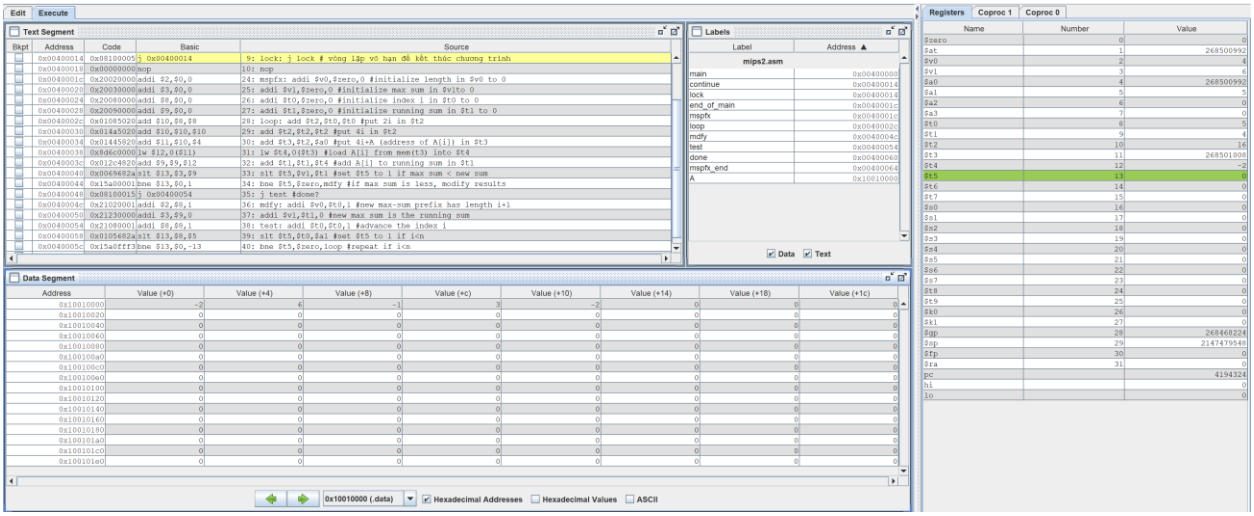
LABORATORY 6

Họ tên: Vũ Quốc Bảo
MSSV: 20225694

Assignment 1:

```
.data
A: .word -2, 6, -1, 3, -2 # khai báo mảng
.text
main: la $a0,A # load địa chỉ mảng A vào thanh ghi $a0
li $a1,5 # load số phần tử của mảng A vào thanh ghi $a1
j mspfx
nop
continue:
lock: j lock # vòng lặp vô hạn để kết thúc chương trình
nop
end_of_main:
#---------------------------------------------------------------
#Procedure mspfx
# @brief find the maximum-sum prefix in a list of integers
# @param[in] a0 the base address of this list(A) need to be processed
# @param[in] a1 the number of elements in list(A)
# @param[out] v0 the length of sub-array of A in which max sum reachs.
# @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 $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
```

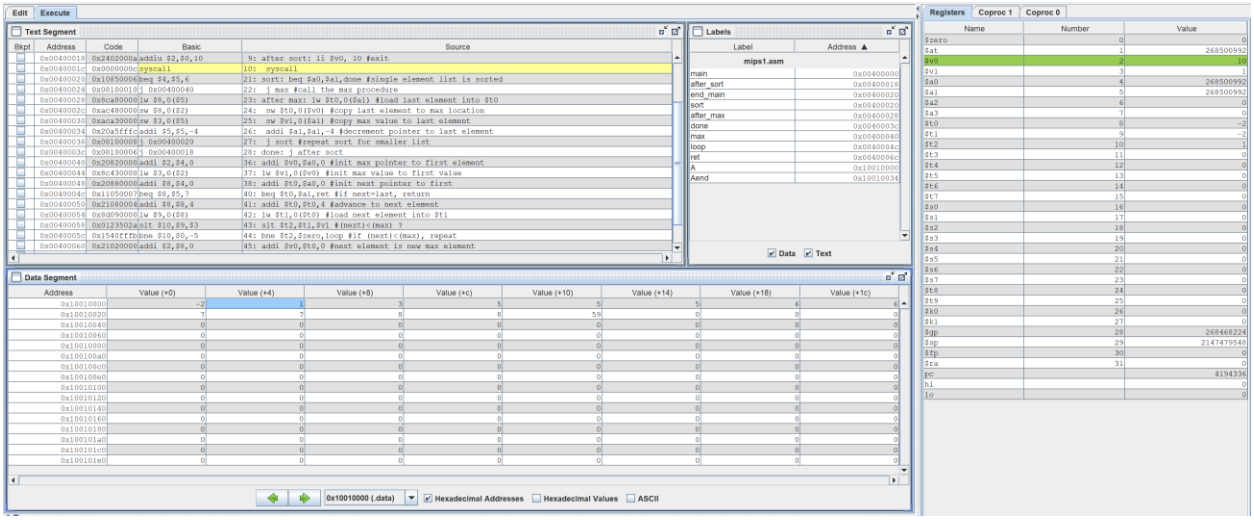
```
bne $t5,$zero,loop #repeat if i<n
done: j continue
mspfx_end:
```



Assignment 2:

```
.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 $v0, 10 #exit
 syscall
end_main:
#------------------------------------------------------------
#procedure sort (ascending selection sort using pointer)
#register usage in sort program
#$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
```

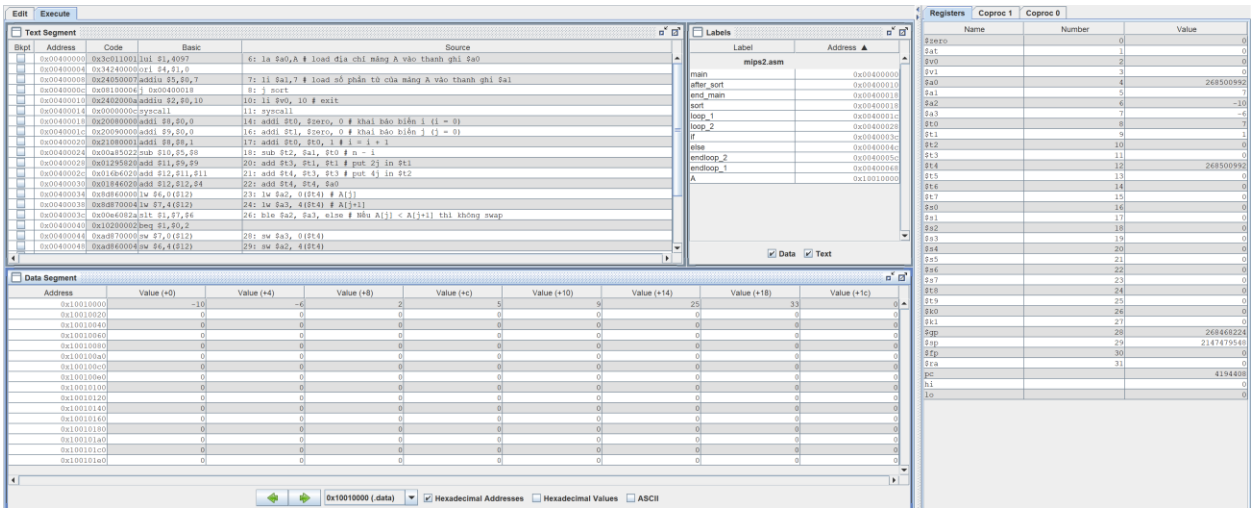
```
 addi $a1,$a1,-4 #decrement pointer to last element
 j sort #repeat sort for smaller list
done: j after_sort
#------------------------------------------------------------------
#Procedure max
#function: fax the value and address of max element in the list
#$a0 pointer to first element
#$a1 pointer 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
```



Assignment 3:

```
# sắp xếp nổi bọt tăng dần
.data
A: .word 5,9,-6,2,-10,33,25 # khai báo mảng bất kì
.text
```

```
main:
la $a0,A # load địa chỉ mảng A vào thanh ghi $a0
li $a1,7 # load số phần tử của mảng A vào thanh ghi $a1
j sort
after_sort:
li $v0, 10 # exit
syscall
end_main:
sort:
addi $t0, $zero, 0 # khai báo biến i (i = 0)
loop_1:
addi $t1, $zero, 0 # khai báo biến j (j = 0)
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
lw $a2, 0($t4) # A[j]
lw $a3, 4($t4) # A[j+1]
if:
ble $a2, $a3, else # Nếu A[j] < A[j+1] thì không swap
# Swap A[j] và A[j+1]
sw $a3, 0($t4)
sw $a2, 4($t4)
else:
addi $t1, $t1, 1 # Tăng j lên 1
slt $t5, $t1, $t2 # $t1 < $t2
beq $t5, $zero, endloop_2
j loop_2
endloop_2:
slt $t6, $t0, $a1 # $t0 < $t1
beq $t6, $zero, endloop_1
j loop_1
endloop_1:
```


Edit
Execute
Text Segment
Bkpt Address Code Basic Source
0x00400000 0x3c011001 lui $1,4097 6: la $a0,A # load địa chỉ mảng A vào thanh ghi $a0
0x00400004 0x34240000 ori $4,$1,0
0x00400008 0x24050007 addiu $5,$0,7 7: li $a1,7 # load số phần tử của mảng A vào thanh ghi $a1
0x0040000c 0x08100006 j 0x00400018 8: j sort
0x00400010 0x2402000a addiu $2,$0,10 10: li $v0, 10 # exit
0x00400014 0x0000000c syscall 11: syscall
0x00400018 0x20080000 addi $8,$0,0 14: addi $t0, $zero, 0 # khai báo biến i (i = 0)
0x0040001c 0x20090000 addi $9,$0,0 16: addi $t1, $zero, 0 # khai báo biến j (j = 0)
0x00400020 0x21080001 addi $8,$8,1 17: addi $t0, $t0, 1 # i = i + 1
0x00400024 0x00a85022 sub $10,$5,$8 18: sub $t2, $a1, $t0 # n - i
0x00400028 0x01295820 add $11,$9,$9 20: add $t3, $t1, $t1 # put 2j in $t1
0x0040002c 0x016b4020 add $12,$11,$11 21: add $t4, $t3, $t3 # put 4j in $t2
0x00400030 0x01846020 add $12,$12,$4 22: add $t4, $t4, $a0
0x00400034 0x8d860000 lw $6,0($12) 23: lw $a2, 0($t4) # A[j]
0x00400038 0x8d870004 lw $7,4($12) 24: lw $a3, 4($t4) # A[j+1]
0x0040003c 0x00e6082a slt $1,$7,$6 26: ble $a2, $a3, else # Nếu A[j] < A[j+1] thì không swap
0x00400040 0x10200002 beq $1,$0,2
0x00400044 0xad870000 sw $7,0($12) 28: sw $a3, 0($t4)
0x00400048 0xad860004 sw $6,4($12) 29: sw $a2, 4($t4)
Labels
Label Address
mips2.asm
main 0x00400000
after_sort 0x00400010
end_main 0x00400018
sort 0x00400018
loop_1 0x0040001c
loop_2 0x00400028
if 0x0040003c
else 0x00400048
endloop_2 0x0040005c
endloop_1 0x00400068
A 0x10010000
Data
Text
Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+c) Value (+10) Value (+14) Value (+18) Value (+1c)
0x10010000 -10 -6 2 5 9 25 33 0
0x10010020 0 0 0 0 0 0 0 0
0x10010040 0 0 0 0 0 0 0 0
0x10010060 0 0 0 0 0 0 0 0
0x10010080 0 0 0 0 0 0 0 0
0x100100a0 0 0 0 0 0 0 0 0
0x100100c0 0 0 0 0 0 0 0 0
0x100100e0 0 0 0 0 0 0 0 0
0x10010100 0 0 0 0 0 0 0 0
0x10010120 0 0 0 0 0 0 0 0
0x10010140 0 0 0 0 0 0 0 0
0x10010160 0 0 0 0 0 0 0 0
0x10010180 0 0 0 0 0 0 0 0
0x100101a0 0 0 0 0 0 0 0 0
0x100101c0 0 0 0 0 0 0 0 0
0x100101e0 0 0 0 0 0 0 0 0
0x10010000 (.data)
Hexadecimal Addresses
Hexadecimal Values
ASCII
Registers
Coproc 1
Coproc 0
Name Number Value
$zero 0 0
$at 1 0
$v0 2 0
$v1 3 0
$a0 4 268500992
$a1 5 7
$a2 6 -10
$a3 7 -6
$t0 8 7
$t1 9 1
$t2 10 0
$t3 11 0
$t4 12 268500992
$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 4194408
hi 0
lo 0