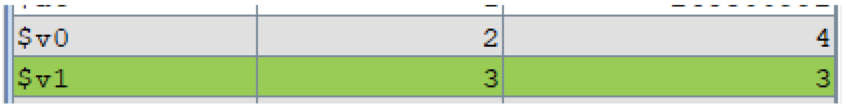
# Thực hành báo cáo Kiến trúc máy tính

**Assignment 1:**

```
.data
A:          .word -5, 6, -1, 3, -2
.text
main:
            la $a0,A
            li $a1,5
            j mspfx
            nop
continue:
lock:
            j lock
            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])
        lw      $t4, 0($t3)               # load A[i] into $t4
        add     $t1, $t1, $t4     # $t1 = sum += A[i]
        slt     $t5, $v1, $t1
        bne     $t5, $zero, mdfy          # if max sum < new sun
        j       test
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:**



|  |  |  |
|---|---|---|
| $v0 | 2 | 4 |
| $v1 | 3 | 3 |

**A = {-5, 6, -1, 3, -2}**

**Length = $v0 = 4**

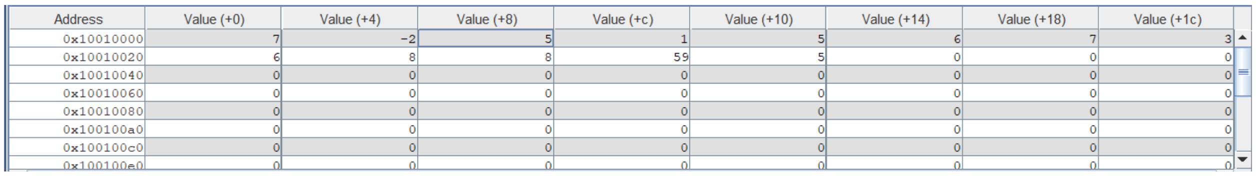
**Khi đó: max_sum = $v1 = -5 + 6 -1 +3 = 3**

**Assignment 2:**

**Code:**

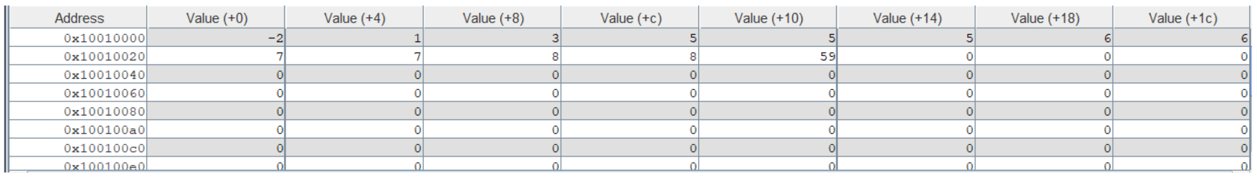
```
.data

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 #$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
```

**Mảng A trước khi sắp xếp:**



| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | 7 | -2 | 5 | 1 | 5 | 6 | 7 | 3 |
| 0x10010020 | 6 | 8 | 8 | 59 | 5 | 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 |

**Mảng A sau khi sắp xếp:**



| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | -2 | 1 | 3 | 5 | 5 | 5 | 6 | 6 |
| 0x10010020 | 7 | 7 | 8 | 8 | 59 | 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 |

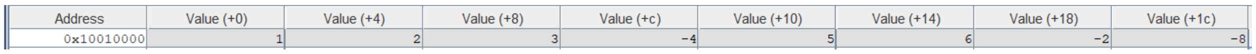
**Assignment 3: Bubble Sort**

**a) Sắp xếp tăng dần:**

```
.data
A:      .word 1,2,3,-4,5,6,-2,-8
.text
main:
   la $a0,A                 #$a0 = Address(A[0])
   li $s0, 8                        #length of array $s0,  length
   j sort                   #sort
end_sort:
   li $v0, 10               #exit
   syscall
end_main:

#Bubble sort algorithm
sort:
        li $t0, 0                               # $t0 = i = 0
loop1:
        slt $v0, $t0, $s0               # set $v0 = 1 when i < length
        beq $v0, $zero, end_sort        # end loop when i >= length
        li      $t1, 0                  # $t1 = j = 0
loop2:
        addi    $t2, $s0, -1
        sub     $t2, $t2, $t0           # $t2 = temp = n-i-1
        slt     $v0, $t1, $t2           # set $v0 = 1 when j < temp
        beq     $v0, $zero, end_loop2
if:
        sll     $t5, $t1, 2                     # $t5 = 4*j
        add     $t5, $t5, $a0           # $t5 is address A[j]
        lw      $t3, 0($t5)             # Load $t3 = A[j]
        lw      $t4, 4($t5)                     # $t4 = A[j+1]
        sgt     $v0, $t3, $t4           # set $v0 = 1 when A[j] > A[j+1]
        beq     $v0, $zero, end_if      # End_if if A[j] <= A[j+1]
        sw      $t4, 0($t5)
        sw      $t3, 4($t5)
end_if:
        addi    $t1, $t1, 1                     # j++
        j       loop2
end_loop2:
        addi    $t0, $t0, 1                     # i++
        j       loop1
```

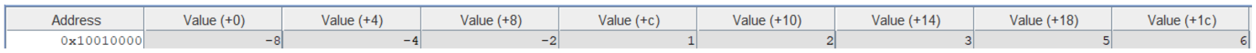
**Kết quả:**

**Trước**



| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | 1 | 2 | 3 | -4 | 5 | 6 | -2 | -8 |

**Sau:**



| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | -8 | -4 | -2 | 1 | 2 | 3 | 5 | 6 |

**b) Sắp xếp giảm dần:**

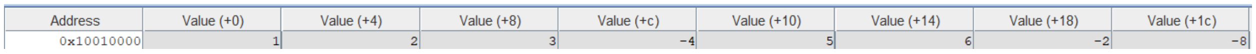
**Code:**

```
.data

A:   .word 1,2,3,-4,5,6,-2,-8

.text

main:

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

  li $s0, 8                 #length of array $s0,  length

  j sort          #sort

end_sort:

  li $v0, 10      #exit

  syscall

end_main:


#Bubble sort algorithm

sort:

    li $t0, 0                           # $t0 = i = 0
```

```
loop1:
    slt $v0, $t0, $s0               # set $v0 = 1 when i < length
    beq $v0, $zero, end_sort        # end loop when i >= length
    li      $t1, 0                  # $t1 = j = 0
loop2:
    addi    $t2, $s0, -1
    sub     $t2, $t2, $t0           # $t2 = temp = n-i-1
    slt     $v0, $t1, $t2           # set $v0 = 1 when j < temp
    beq     $v0, $zero, end_loop2
if:
    sll     $t5, $t1, 2                     # $t5 = 4*j
    add     $t5, $t5, $a0           # $t5 is address A[j]
    lw      $t3, 0($t5)             # Load $t3 = A[j]
    lw      $t4, 4($t5)                     # $t4 = A[j+1]
    sub     $v0, $t3, $t4           # v0 = $t3 - $t4 =  A[j] - A[j+1]
    slt     $v0, $v0,$zero          # v0 = 1 if A[j] - A[j+1]  < 0

    beq     $v0, $zero, end_if      # End_if if A[j] > A[j+1]
    sw      $t4, 0($t5)
    sw      $t3, 4($t5)
end_if:
    addi    $t1, $t1, 1                     # j++
    j       loop2
end_loop2:
    addi    $t0, $t0, 1                     # i++
    j       loop1
```

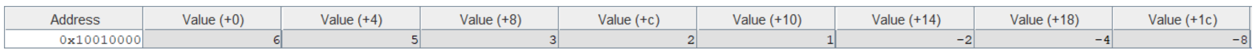
**Kết quả:**

**Trước:**



| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | 1 | 2 | 3 | -4 | 5 | 6 | -2 | -8 |

**Sau:**



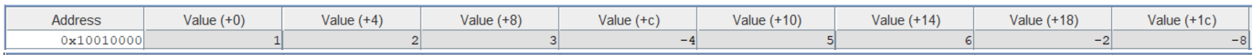
| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | 6 | 5 | 3 | 2 | 1 | -2 | -4 | -8 |

**Assignment 4:**

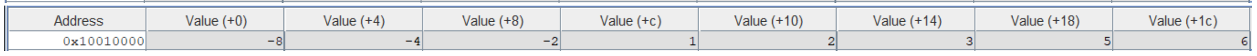
a) **Sắp xếp tăng dần**
**Code:**

```
.data
   A: .word 1,2,3,-4,5,6,-2,-8
.text
main:
   la $a0,A    #$a0 = Address(A[0])
   li $s0, 8  #length of array $s0,  length
   j sort      #sort
end_sort:
   li $v0, 10 #exit
   syscall
end_main:



#Insertion sort
sort:
   li $t0, 1   # $t0, i = 1
   li $t1, 0   # $t1, key = 0
   li $t2, 0   # $t2, j = 0
loop:
    slt $v0, $t0, $s0                  # set $v0 = 1 when i < length
    beq $v0, $zero, end_sort         # end loop when i >= length
    sll   $t3, $t0, 2                            # $t3 = 4*i
    add $t3, $t3, $a0        # $t3 is address A[i]
    lw $t1, 0($t3)                     # key = A[i]
    add $t2, $t0, -1                   # j = i - 1
while:
   slt $v0, $t2, $zero                # set $v0 = 1 when j < 0
   bne $v0, $zero, end_while
   sll   $t3, $t2, 2                             # $t3 = 4*j
   add $t3, $t3, $a0         # $t3 is address A[j]
   lw $t4, 0($t3)                      # Load A[j]
   slt $v0, $t1, $t4                   # set $v0 = 1 when A[j] > key
   beq $v0, $zero, end_while   # End while if key >=  A[j]
   sw $t4, 4($t3)                      # A[j+1] = A[j]
   add $t2, $t2, -1                    # j = j - 1
   j while
end_while:
   add $t3, $t2, $t2        # $t3 = 2*j
   add $t3, $t3, $t3        # $t3 = 4*j
   add $t3, $t3, $a0        # $t3 is address A[j]
   sw $t1, 4($t3)                      # A[j+1] = key
   add $t0, $t0, 1                     # i = i + 1
   j loop
```

**Kết quả:**



| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | 1 | 2 | 3 | -4 | 5 | 6 | -2 | -8 |



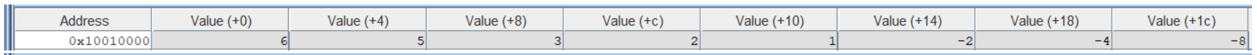
| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | -8 | -4 | -2 | 1 | 2 | 3 | 5 | 6 |

b) **Sắp xếp giảm dần**

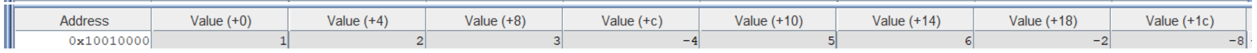
**Code:**

```
.data
   A: .word 1,2,3,-4,5,6,-2,-8
.text
main:
   la $a0,A    #$a0 = Address(A[0])
   li $s0, 8  #length of array $s0,  length
   j sort      #sort
end_sort:
   li $v0, 10 #exit
   syscall
end_main:



#Insertion sort
sort:
   li $t0, 1   # $t0, i = 1
   li $t1, 0   # $t1, key = 0
   li $t2, 0   # $t2, j = 0
loop:
    slt $v0, $t0, $s0                        # set $v0 = 1 when i < length
    beq $v0, $zero, end_sort                 # end loop when i >= length
    sll       $t3, $t0, 2                              # $t3 = 4*i
    add $t3, $t3, $a0               # $t3 is address A[i]
    lw $t1, 0($t3)                           # key = A[i]
    add $t2, $t0, -1                         # j = i - 1
while:
   slt $v0, $t2, $zero                      # set $v0 = 1 when j < 0
   bne $v0, $zero, end_while
   sll      $t3,$t2,2                               # $t3 = 4*j
   add $t3, $t3, $a0                # $t3 is address A[j]
   lw $t4, 0($t3)                           # Load A[j]
   slt $v0, $t4, $t1                        # set $v0 = 1 when A[j] < key
   beq $v0, $zero, end_while                # End while if key <=  A[j]
   sw $t4, 4($t3)                           # A[j+1] = A[j]
   add $t2, $t2, -1                         # j = j - 1
   j while
end_while:
   sll      $t3,$t2,2                               # $t3 = 4*j
   add $t3, $t3, $a0                # $t3 is address A[j]
   sw $t1, 4($t3)                           # A[j+1] = key
   add $t0, $t0, 1                          # i = i + 1
   j loop
```

Kết quả:



| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | 6 | 5 | 3 | 2 | 1 | -2 | -4 | -8 |



| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|---|---|---|---|---|---|---|---|---|
| 0x10010000 | 1 | 2 | 3 | -4 | 5 | 6 | -2 | -8 |