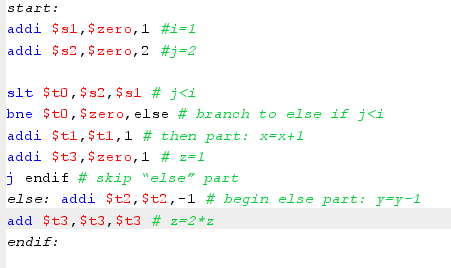
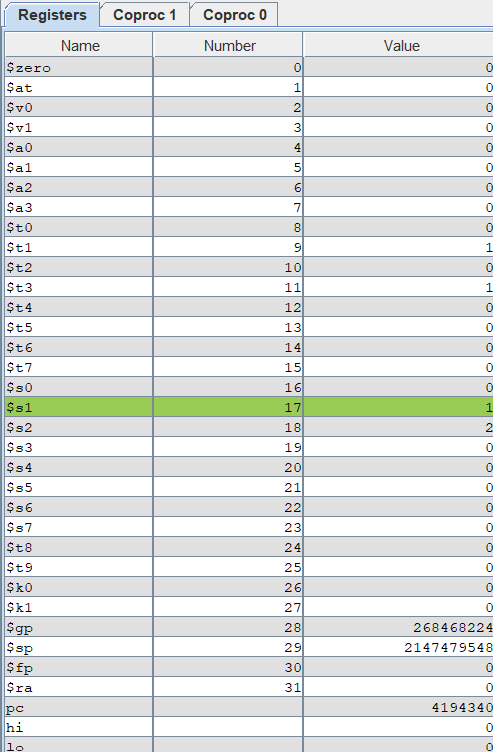
**Assignment 1**

* Code



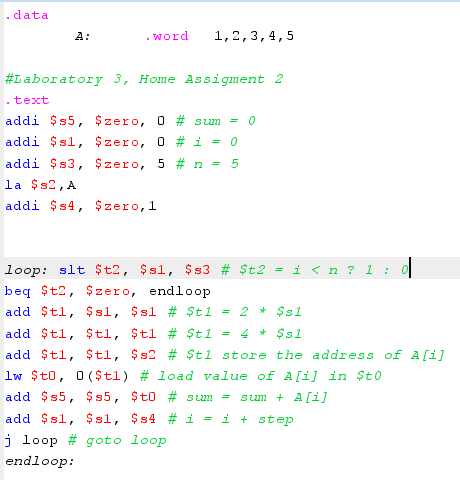
* Register



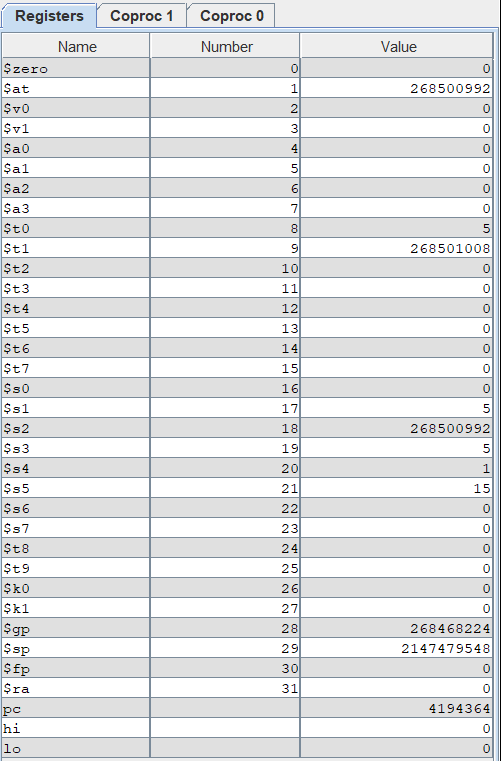
* Lệnh addi $s1,$zero,1
* $s1=1 (i=1)
* Lệnh addi $S2,,$zero,2
* $s2 = 2 (j=2)
* Lệnh slt $t0,$s2,$s1
* $t0 = 0 (vì j<i false )
* Lệnh bne $t0,$zero,else
* Lệnh addi $t1 ,$t1,1
* $t1 = 1 ( x = x+1)
* Lệnh addi $t3, $zero,1
* $t3 = 1 (z = 1)
* j endif
* skip “else” part

**Assignment 2**

* code



* Register



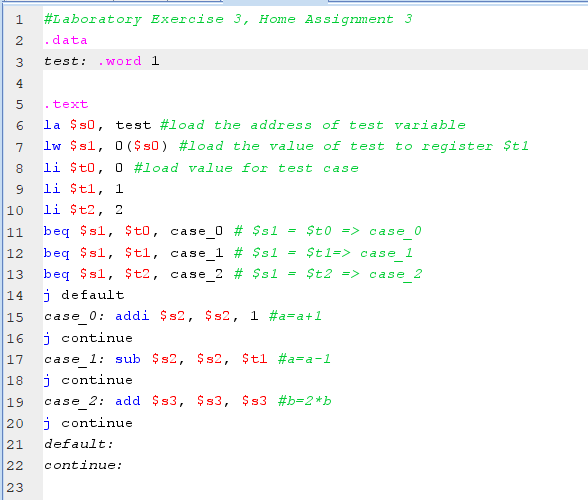
* Lệnh addi $s5, $zero, 0
* $s5 = 0 ( sum = 0 )
* Lệnh addi $s1, $zero, 0
* $s1 = 0 ( i = 0 )
* Lệnh addi $s3, $zero, 5
* $s3 = 5 (# n = 5)
* Lệnh la $s2,A
* $s2 = 0x10010000
* Lệnh addi $s4, $zero,1
* $s4 =1 (step =1)
* Lệnh loop: slt $t2, $s1, $s3
* $t2 =1 (# $t2 = i < n ? 1 : 0)
* Lệnh beq $t2, $zero, endloop
* $t2 != $zero => tiếp tục vòng lặp
* Lệnh add $t1, $s1, $s1

add $t1, $t1, $t1 ( $t1 = 4$s1 )

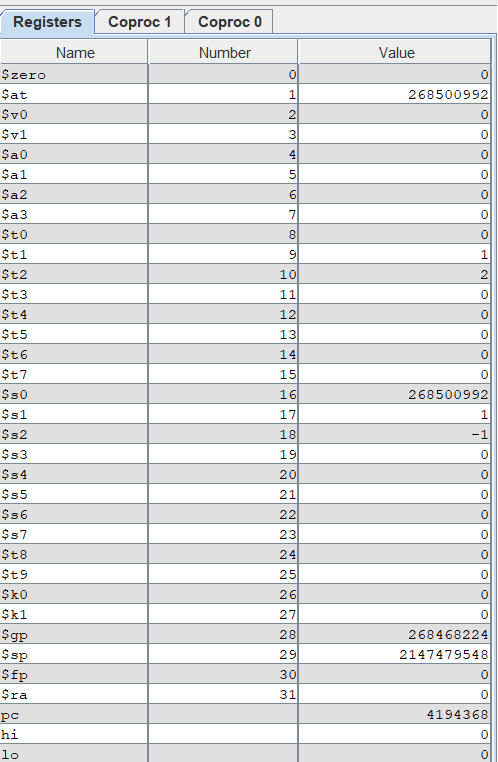
* $t1 = 0
* Lệnh add $t1, $t1, $s2
* $t1 = 0x10010000 ( # $t1 store the address of A[i] )
* Lệnh lw $t0, 0($t1)
* $t0 = 1 ( # load value of A[i] in $t0)
* Lệnh add $s5, $s5, $t0
* $s5 = 1 (# sum = sum + A[i])
* Lệnh add $s1, $s1, $s4
* $s1 =1 (# i = i + step)
* j loop # goto loop
* tiến hành lặp lại vòng lặp cho tới khi $s1>= $s3 ( I >= n )

**Assignment 3**

* Code



* Register



* Lệnh la $s0, test
* $s0 = 268500992 ( #load the address of test variable )
* Lệnh lw $s1, 0($s0)
* $s1 = 1 ( #load the value of test to register $t1 )
* Lệnh li $t0 0

li $t1, 1

li $t2, 2

* $t0 = 0, $t1 = 1, $t2 = 2
* beq $s1, $t0, case\_0
* $s1 != $t0 =>không lọt vào case\_0
* Lệnh beq $s1, $t1, case\_1
* $s1 = $t1 => case\_1
* Lệnh case\_1: sub $s2, $s2, $t1
* $s2 =-1 ( #a=a-1 )

**Assignment 4**

1. **i<j**

start:

addi $s1,$zero,1 #i=1

addi $s2,$zero,2 #j=2

slt $t0,$s1,$s2 # i<j 1

bne $t0,$zero,else # branch to else if

addi $t2,$t2,-1 # begin else part: y=y-1

add $t3,$t3,$t3 # z=2\*z

j endif # skip “else” part

else:

addi $t1,$t1,1 # then part: x=x+1

addi $t3,$zero,1 # z=1

endif:

1. i>= j

start:

addi $s1,$zero,1 #i=1

addi $s2,$zero,2 #j=2

slt $t0,$s1,$s2 # i < j

bne $t0,$zero,else # branch to else if

addi $t1,$t1,1 # then part: x=x+1

addi $t3,$zero,1 # z=1

j endif # skip “else” part

else: addi $t2,$t2,-1 # begin else part: y=y-1

add $t3,$t3,$t3 # z=2\*z

endif:

1. **i + j<=0**

start:

addi $s1,$zero,1 #i=1

addi $s2,$zero,-2 #j=-2

add $s3,$1,$s2 # i+j

slt $t0,$zero,$s3 # i +j >0

bne $t0,$zero,else # branch to else if

addi $t1,$t1,1 # then part: x=x+1

addi $t3,$zero,1 # z=1

j endif # skip “else” part

else: addi $t2,$t2,-1 # begin else part: y=y-1

add $t3,$t3,$t3 # z=2\*z

endif:

1. **i + j < m + n**

start:

addi $s1,$zero,1 #i=1

addi $s2,$zero,-2 #j=-2

add $s3,$1,$s2 # i+j

addi $s4,$zero,4 # m=4

addi $s5,$zero,5 # n=5

add $s6 ,$s4,$s5 # m+n

slt $t0,$s3,$s6 # i +j < m + n

bne $t0,$zero,else # branch to else if

addi $t2,$t2,-1 # begin else part: y=y-1

add $t3,$t3,$t3 # z=2\*z

j endif # skip “else” part

else:

addi $t1,$t1,1 # then part: x=x+1

addi $t3,$zero,1 # z=1

endif:

**Assignment 5**

1. **i < n**

.data

A: .word 1,2,3,4,5

#Laboratory 3, Home Assigment 2

.text

addi $s5, $zero, 0 # sum = 0

addi $s1, $zero, 5 # i = 0

addi $s3, $zero, 5 # n = 5

la $s2,A

addi $s4, $zero,1

loop: slt $t2, $s1, $s3 # $t2 = i < n ? 1 : 0

bne $t2, $zero, endloop

add $t1, $s1, $s1 # $t1 = 2 \* $s1

add $t1, $t1, $t1 # $t1 = 4 \* $s1

add $t1, $t1, $s2 # $t1 store the address of A[i]

lw $t0, 0($t1) # load value of A[i] in $t0

add $s5, $s5, $t0 # sum = sum + A[i]

add $s1, $s1, $s4 # i = i + step

j loop # goto loop

endloop:

1. **i <= n**

.data

A: .word 1,2,3,4,5

#Laboratory 3, Home Assigment 2

.text

addi $s5, $zero, 0 # sum = 0

addi $s1, $zero, 0 # i = 0

addi $s3, $zero, 5 # n = 5

la $s2,A

addi $s4, $zero,1

loop: slt $t2, $s3, $s1 # $t2 = n < i ? 1 : 0

beq $t2, $zero, endloop

add $t1, $s1, $s1 # $t1 = 2 \* $s1

add $t1, $t1, $t1 # $t1 = 4 \* $s1

add $t1, $t1, $s2 # $t1 store the address of A[i]

lw $t0, 0($t1) # load value of A[i] in $t0

add $s5, $s5, $t0 # sum = sum + A[i]

add $s1, $s1, $s4 # i = i + step

j loop # goto loop

endloop:

1. **sum >= 0**

.data

A: .word -1,-2,-3,-4,5,6,7

#Laboratory 3, Home Assigment 2

.text

addi $s5, $zero, 0 # sum = 0

addi $s1, $zero, 0 # i = 0

addi $s3, $zero, 7 # n = 5

la $s2,A

addi $s4, $zero,1

loop:

slt $t2, $s1, $s3 # $t2 = i < n ? 1 : 0

beq $t2, $zero, endloop

add $t1, $s1, $s1 # $t1 = 2 \* $s1

add $t1, $t1, $t1 # $t1 = 4 \* $s1

add $t1, $t1, $s2 # $t1 store the address of A[i]

lw $t0, 0($t1) # load value of A[i] in $t0

add $s5, $s5, $t0 # sum = sum + A[i]

add $s1, $s1, $s4 # i = i + step

slt $t3 ,$s5,$zero # $t3 = sum < 0 ? 1 : 0

beq $t3, $zero, endloop # $t3 = 0 => sum>=0 => end loop

j loop # goto loop

endloop:

1. **A[i] == 0**

.data

A: .word 1,2,3,4,5,6,0,7

#Laboratory 3, Home Assigment 2

.text

addi $s5, $zero, 0 # sum = 0

addi $s1, $zero, 0 # i = 0

addi $s3, $zero, 7 # n = 5

la $s2,A

addi $s4, $zero,1

loop:

slt $t2, $s1, $s3 # $t2 = i < n ? 1 : 0

beq $t2, $zero, endloop

add $t1, $s1, $s1 # $t1 = 2 \* $s1

add $t1, $t1, $t1 # $t1 = 4 \* $s1

add $t1, $t1, $s2 # $t1 store the address of A[i]

lw $t0, 0($t1) # load value of A[i] in $t0

beq $t0, $zero, endloop # $t0 == 0 ( A[i] ==0 ) => end loop

add $s5, $s5, $t0 # sum = sum + A[i]

add $s1, $s1, $s4 # i = i + step

j loop # goto loop

endloop:

**Assignment 6**

.data

NhapN: .asciiz "Nhap so phan tu mang: "

Xuat\_Mang: .asciiz "Mang vua nhap la: \n"

Xuat: .asciiz "Nhap phan tu thu "

Xuat\_Max: .asciiz "Phan tu co gia tri tuyet doi lon nhat la : "

tong: .word 0

a: .word 0:100

.text

.globl main

main:

la $a0, NhapN # in NhapN

li $v0, 4

syscall

li $v0, 5 # nhap n va luu vao $s0

syscall

move $s0, $v0

li $t0, 0 # khoi tao vong lap , i = 0

la $a1, a #load a vao $a1

li $s1,0 # khoi tao phan tu max =0

Enter:

blt $t0, $s0, enter\_Element # Neu i < n thi enter\_Element

j out

enter\_Element:

la $a0, Xuat # print string Xuat

li $v0, 4

syscall

li $v0, 1 # print index

move $a0, $t0

syscall

li $v0, 5 # nhap so nguyen va luu vao mang

syscall

sw $v0, ($a1)

addi $t0, $t0, 1 #Tang chi so

addi $a1, $a1, 4

abs $s2,$v0 # bien ve gia tri tuyet doi: $s2 = |v0=a[i]|

# so sanh voi p tu max

slt $t2,$s1,$s2 # $t2 = $s1 < $s2 ( max <a[i]) ? 1: 0

beq $t2,$zero,Enter # t2=0 ( max>=a[i] ) => quay lai enter

add $s1,$s2,$zero # t2=1 ( max < a[i] ) => gan $s1=$s2 ( max = a[i] )

j Enter

out:

la $a0,Xuat\_Max #print string Xuat\_Max

li $v0,4

syscall

li $v0, 1 # print phan tu max

move $a0, $s1

syscall