THỰC HÀNH KIẾN TRÚC MÁY TÍNH

IT-3280

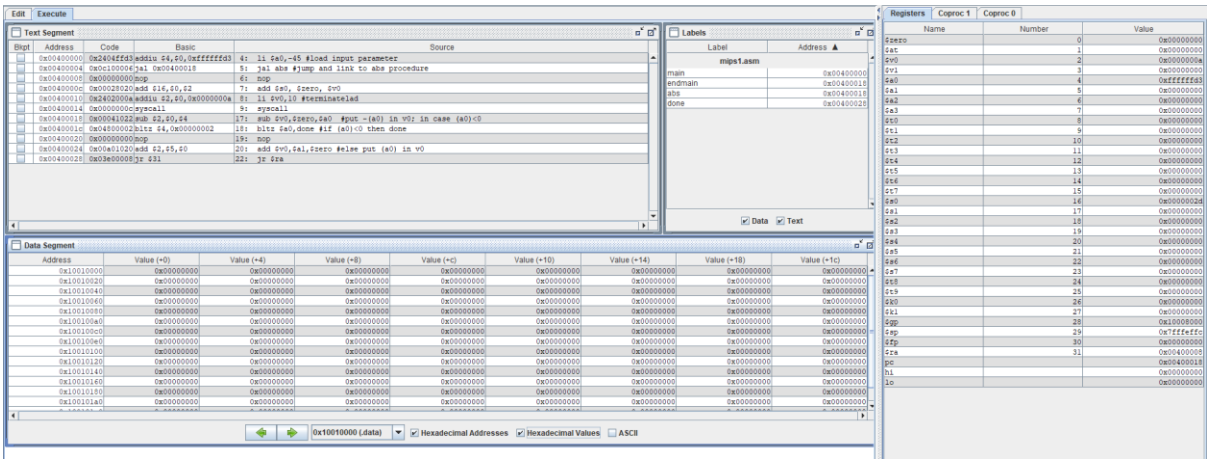
Họ tên	Thân Cát Ngọc Lan
MSSV	20225646

Assignment 1:

*Code

```
#Laboratory Exercise 7 Home Assignment 1
.text
main:
 li $a0,-45 #load input parameter
 jal abs #jump and link to abs procedure
 nop
 add $s0, $zero, $v0
 li $v0,10 #terminatelad
 syscall
endmain:
#---------------------------------------------------------------
# function abs
# param[in] $a1 the interger need to be gained the absolute
# return $v0 absolute value
#---------------------------------------------------------------
abs:
 sub $v0,$zero,$a0  #put -(a0) in v0; in case (a0)<0
 bltz $a0,done #if (a0)<0 then done
 nop
 add $v0,$a1,$zero #else put (a0) in v0
done:
 jr $ra
```

*Kết quả



***Giải thích:**

- Quan sát sự thay đổi của thanh ghi $ra và thanh ghi $pc:

+ Ngay trước khi thực hiện chương trình con (lệnh jal abs):

$ra	31	0x00000000
pc		0x00400004

+ Sau lệnh jal abs (bắt đầu chương trình con), thanh ghi $ra lưu giá trị là địa chỉ của lệnh ngay sau nó: nop (có địa chỉ 0x00400008). Thanh ghi pc hiện tại có giá trị 0x00400018
(là địa chỉ của lệnh sub $v0,$zero,$a0).

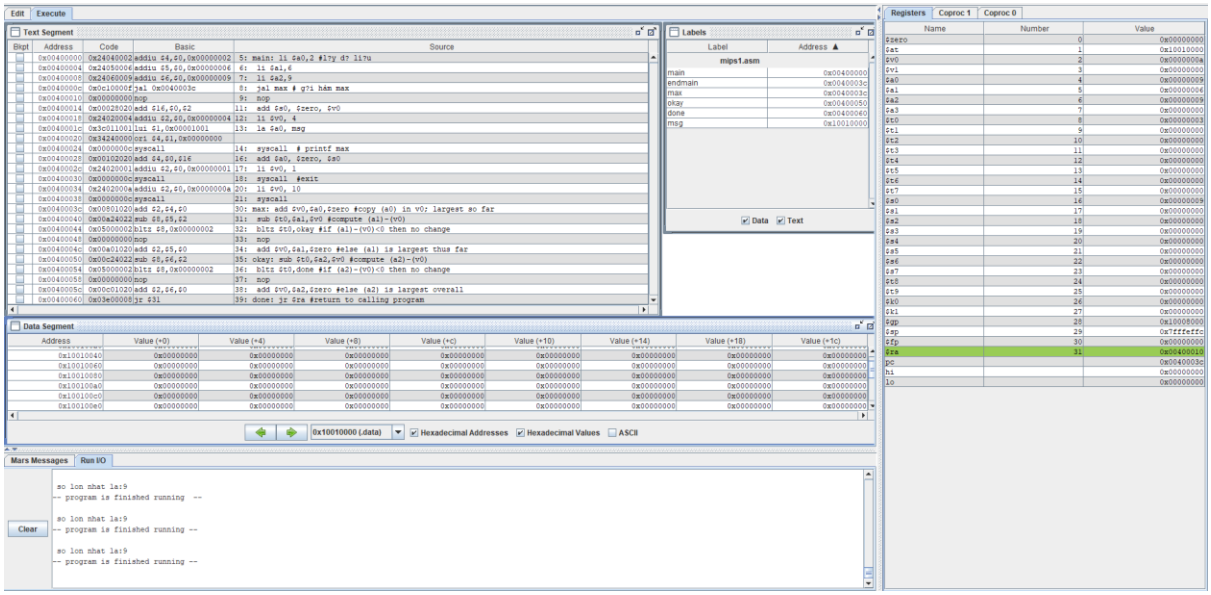
$ra	31	0x00400008
pc		0x00400018

Assignment 2:

***Code**

```
#Laboratory Exercise 7, Home Assignment 2
.data
msg: .asciiz " so lon nhat la:"
.text
main: li $a0,2 #load test input
 li $a1,6
 li $a2,9
 jal max #call max procedure
 nop

 add $s0, $zero, $v0
 li $v0, 4
 la $a0, msg
 syscall

 add $a0, $zero, $s0
 li $v0, 1
 syscall

 li $v0, 10
 syscall
endmain:
#---------------------------------------------------------------
#Procedure max: find the largest of three integers
#param[in] $a0 integers
#param[in] $a1 integers
#param[in] $a2 integers
#return $v0 the largest value
#---------------------------------------------------------------
max: add $v0,$a0,$zero #copy (a0) in v0; largest so far
 sub $t0,$a1,$v0 #compute (a1)-(v0)
 bltz $t0,okay #if (a1)-(v0)<0 then no change
 nop
 add $v0,$a1,$zero #else (a1) is largest thus far
okay: sub $t0,$a2,$v0 #compute (a2)-(v0)
 bltz $t0,done #if (a2)-(v0)<0 then no change
 nop
 add $v0,$a2,$zero #else (a2) is largest overall
done: jr $ra #return to calling program
```

***Kết quả:**



***Giải thích:**

- Quan sát sự thay đổi của thanh ghi $ra và thanh ghi $pc:

+ Ngay trước khi thực hiện chương trình con (lệnh jal max):

$ra	31	0x00000000
pc		0x0040000c

+ Sau lệnh jal max (bắt đầu chương trình con), thanh ghi $ra lưu giá trị là địa chỉ của lệnh ngay sau nó: nop (có địa chỉ 0x00400010). Thanh ghi pc hiện tại có giá trị 0x0040003c
(là địa chỉ của lệnh add $v0,$a0,$zero).

$ra	31	0x00400010
pc		0x0040003c

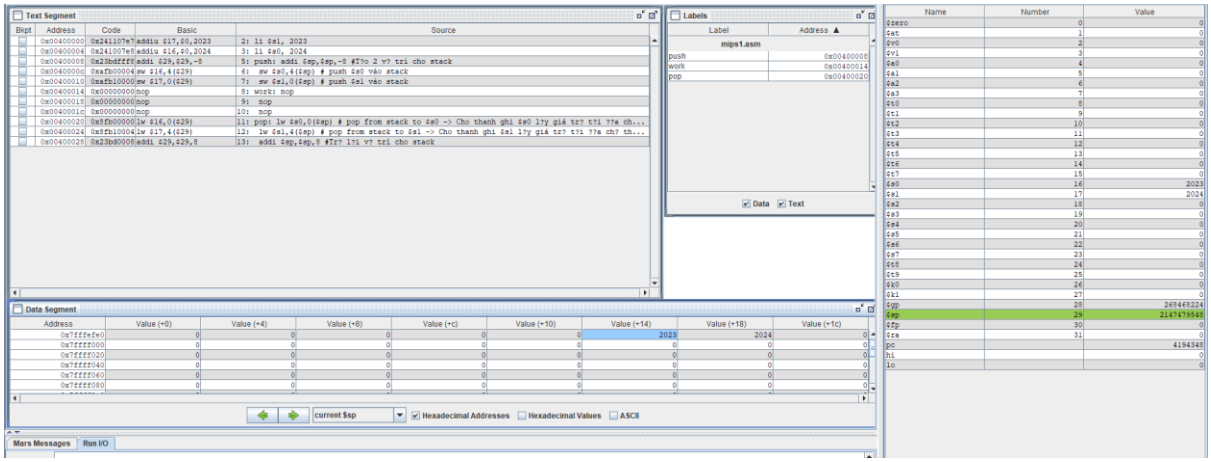
Assignment 3:

***Code**

```
#Laboratory Exercise 7, Home Assignment 3
li $s1, 2023
li $s0, 2024
.text
push: addi $sp,$sp,-8 #Tao 2 vi tri cho stack
 sw $s0,4($sp) # push $s0 vào stack
 sw $s1,0($sp) # push $s1 vào stack
work: nop
 nop
 nop
pop: lw $s0,0($sp) # pop from stack to $s0 -> Cho thanh ghi $s0 lấy giá trị tại địa chỉ thanh ghi $s1
 lw $s1,4($sp) # pop from stack to $s1 -> Cho thanh ghi $s1 lấy giá trị tại địa chỉ thanh ghi $s0
 addi $sp,$sp,8 #Trả lại vị trí cho stack

```

***Kết quả**

Khởi tạo $s0 = 2024, $s1 = 2023



⇨ Sau khi chạy chương trình: $s0 = 2023, $s1 = 2024. Kết quả chạy đúng.

***Giải thích:**

- Sau khi gặp lệnh addi $sp,$sp,-8. Địa chỉ thanh ghi $sp giảm 8. Có thể hiểu stack dành ra 2 vị trí để lưu các giá trị. Khi nạp giá trị vào lúc này địa chỉ thanh ghi $sp không thay đổi.
- Giải thích chi tiết trong code.

Assignment 4:

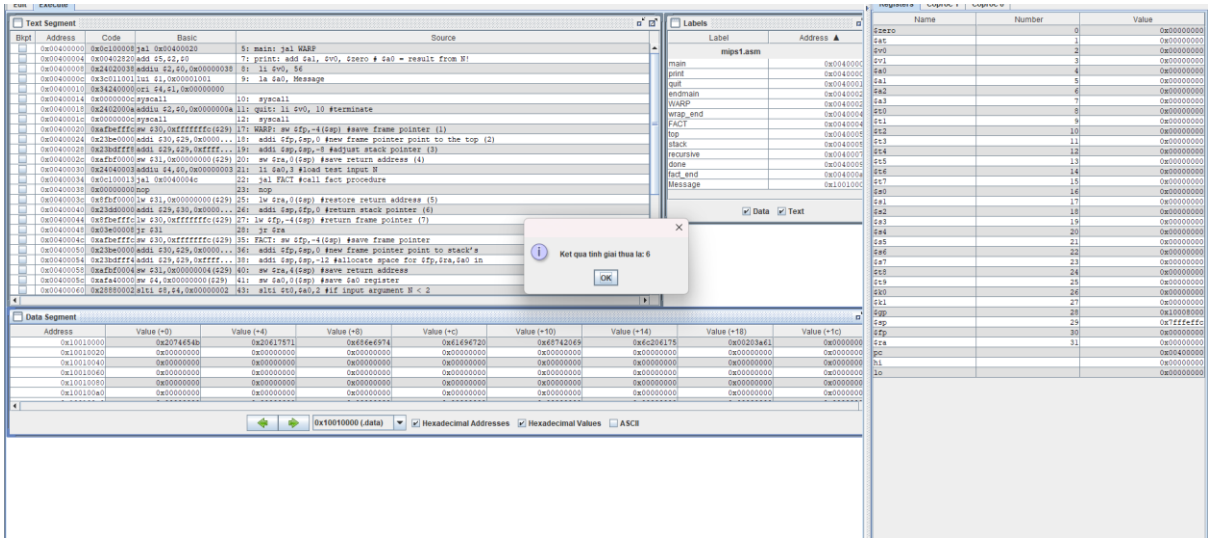
***Code:**

```
#Laboratory Exercise 7, Home Assignment 4
.data
Message: .asciiz "Ket qua tinh giai thua la: "
.text
main: jal WARP

print: add $a1, $v0, $zero # $a0 = result from N!
 li $v0, 56
 la $a0, Message
 syscall
quit: li $v0, 10 #terminate
 syscall
endmain:
#---------------------------------------------------------------
#Procedure WARP: assign value and call FACT
#---------------------------------------------------------------
WARP: sw $fp,-4($sp) #save frame pointer (1)
 addi $fp,$sp,0 #new frame pointer point to the top (2)
 addi $sp,$sp,-8 #adjust stack pointer (3)
 sw $ra,0($sp) #save return address (4)
 li $a0,3 #load test input N
 jal FACT #call fact procedure
 nop

 lw $ra,0($sp) #restore return address (5)
 addi $sp,$fp,0 #return stack pointer (6)
lw $fp,-4($sp) #return frame pointer (7)
 jr $ra
wrap_end:
#---------------------------------------------------------------
#Procedure FACT: compute N!
#param[in] $a0 integer N
#return $v0 the largest value
#---------------------------------------------------------------
```

```
#-----------------------------------------------------------------------
FACT: sw $fp,-4($sp) #save frame pointer
 addi $fp,$sp,0 #new frame pointer point to stack's
top:
 addi $sp,$sp,-12 #allocate space for $fp,$ra,$a0 in
stack:
 sw $ra,4($sp) #save return address
 sw $a0,0($sp) #save $a0 register

 slti $t0,$a0,2 #if input argument N < 2
 beq $t0,$zero,recursive#if it is false ((a0 = N) >=2)
 nop
 li $v0,1 #return the result N!=1
 j done
 nop
recursive:
 addi $a0,$a0,-1 #adjust input argument
 jal FACT #recursive call
 nop
 lw $v1,0($sp) #load a0
 mult $v1,$v0 #compute the result
 mflo $v0
done: lw $ra,4($sp) #restore return address
 lw $a0,0($sp) #restore a0
 addi $sp,$fp,0 #restore stack pointer
 lw $fp,-4($sp) #restore frame pointer
 jr $ra #jump to calling
fact end:
```

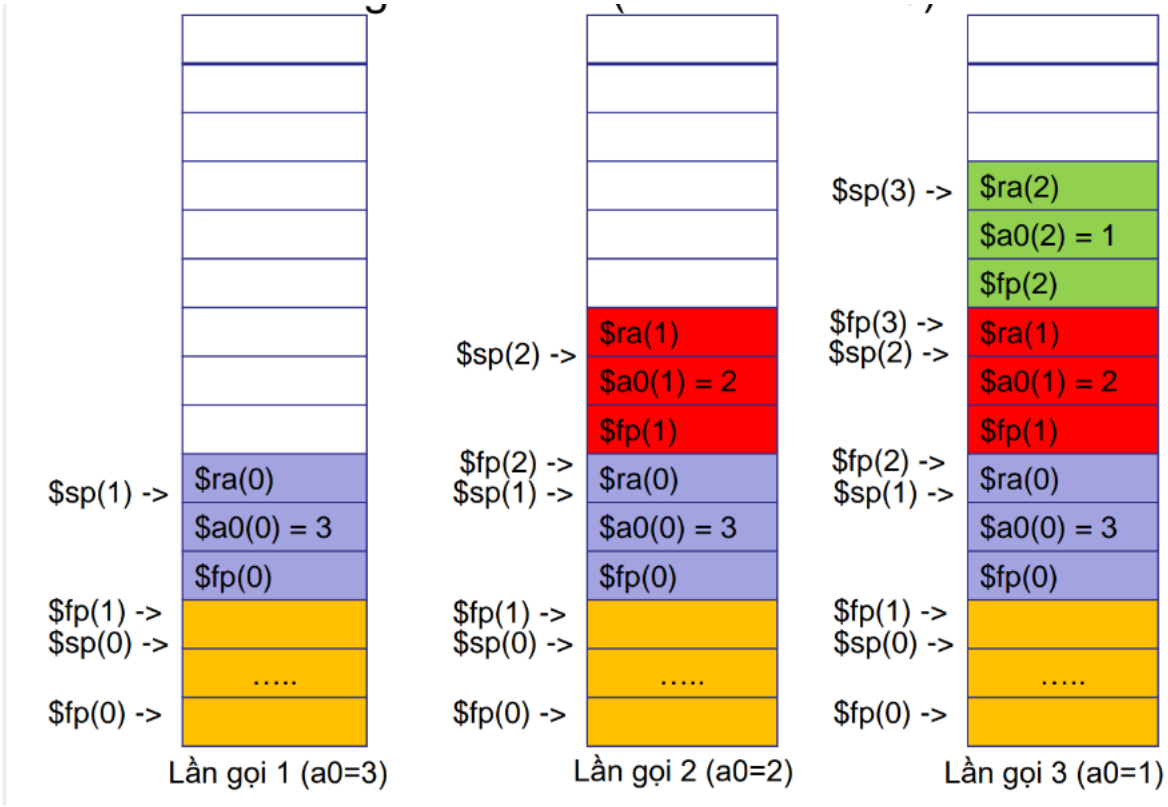
***Kết quả:**

Khởi tạo n=3;



***Giải thích:**

Vẽ ngăn xếp đệ quy:



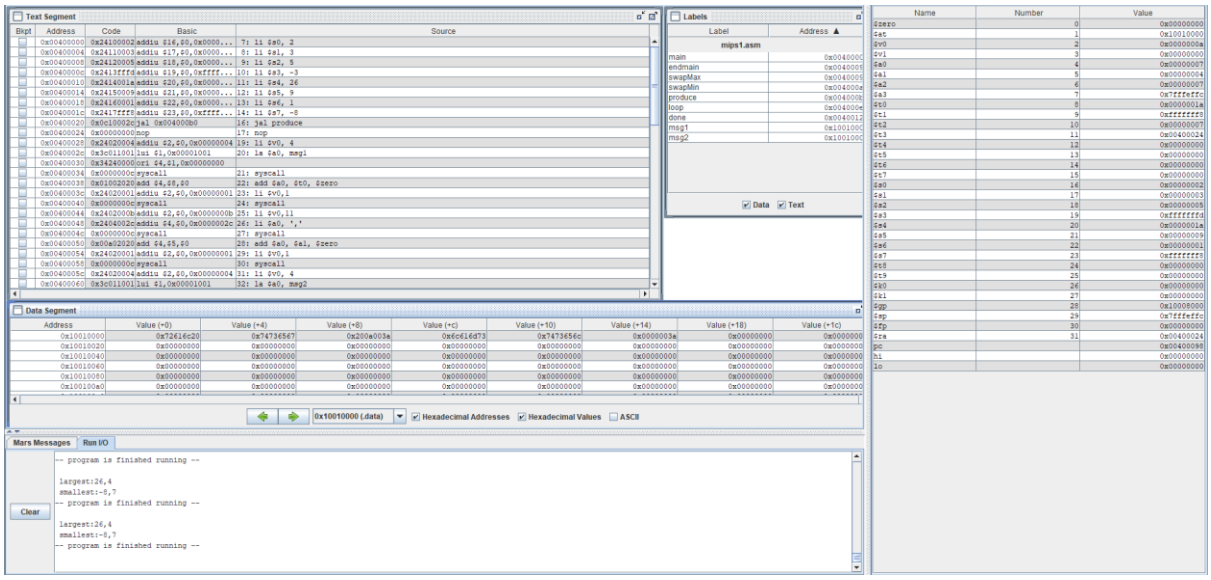
Assignment 5:

*Code:

```
.data
msg1: .asciiz " largest:"
msg2: .asciiz "\n smallest:"
.text
main:
# Khởi tạo giá trị từ thanh ghi $s0 đến thanh ghi $s7
li $s0, 2
li $s1, 3
li $s2, 5
li $s3, -3
li $s4, 26
li $s5, 9
li $s6, 1
li $s7, -8

jal produce
nop

li $v0, 4
la $a0, msg1
syscall
add $a0, $t0, $zero
li $v0,1
syscall
li $v0,11
li $a0, ','
syscall
add $a0, $a1, $zero
li $v0,1
syscall
li $v0, 4
la $a0, msg2
syscall
add $a0, $t1, $zero
li $v0,1
syscall
li $v0,11
li $a0, ','
syscall
add $a0, $a2, $zero
li $v0,1
syscall
li $v0, 10
syscall
endmain:
```

```
endmain:

swapMax: add $t0,$t3,$zero
add $a1,$t2,$zero
jr $ra

swapMin: add $t1,$t3,$zero
add $a2,$t2,$zero
jr $ra

produce: #Tìm giá trị nhỏ nhất và lớn nhất
add $a3,$sp,$zero # Lưu địa chỉ bắt đầu của thanh ghi $sp
addi $sp, $sp, -32 # integerS in stack
sw $s1, 0($sp)
sw $s2, 4($sp)
sw $s3, 8($sp)
sw $s4, 12($sp)
sw $s5, 16($sp)
sw $s6, 20($sp)
sw $s7, 24($sp)
sw $ra, 28($sp)
add $t0,$s0,$zero # Max = $s0
add $t1,$s0,$zero # Min = $s0
li $a1, 0 # Index of Max
li $a2, 0 # Index of Min
li $t2, 0 # i = 0
loop:
addi $sp, $sp, 4
lw $t3, -4($sp)
sub $t6, $sp, $a3
beq $t6,$zero, done # Nếu $sp = $fp nhảy tới 'done'
nop
addi $t2,$t2,1 # i++
sub $t6,$t0,$t3
bltzal $t6, swapMax # If $t3 > Max branch to the swapMax
nop
sub $t6,$t3,$t1
bltzal $t6, swapMin # If $t3 < Min branch to the swapMin
nop
j loop
done:
lw $ra, -4($sp)
jr $ra # Trở lại
```

*Kết quả:



⇨ **Kết quả chạy đúng.**