Lab Report 07

## Assignment 1

**Comments:**

* Initially
  + $pc = 0040000, the beginning of main
  + $ra = 0000000
  + $sp = 7fffeffc, the beginning of stack

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

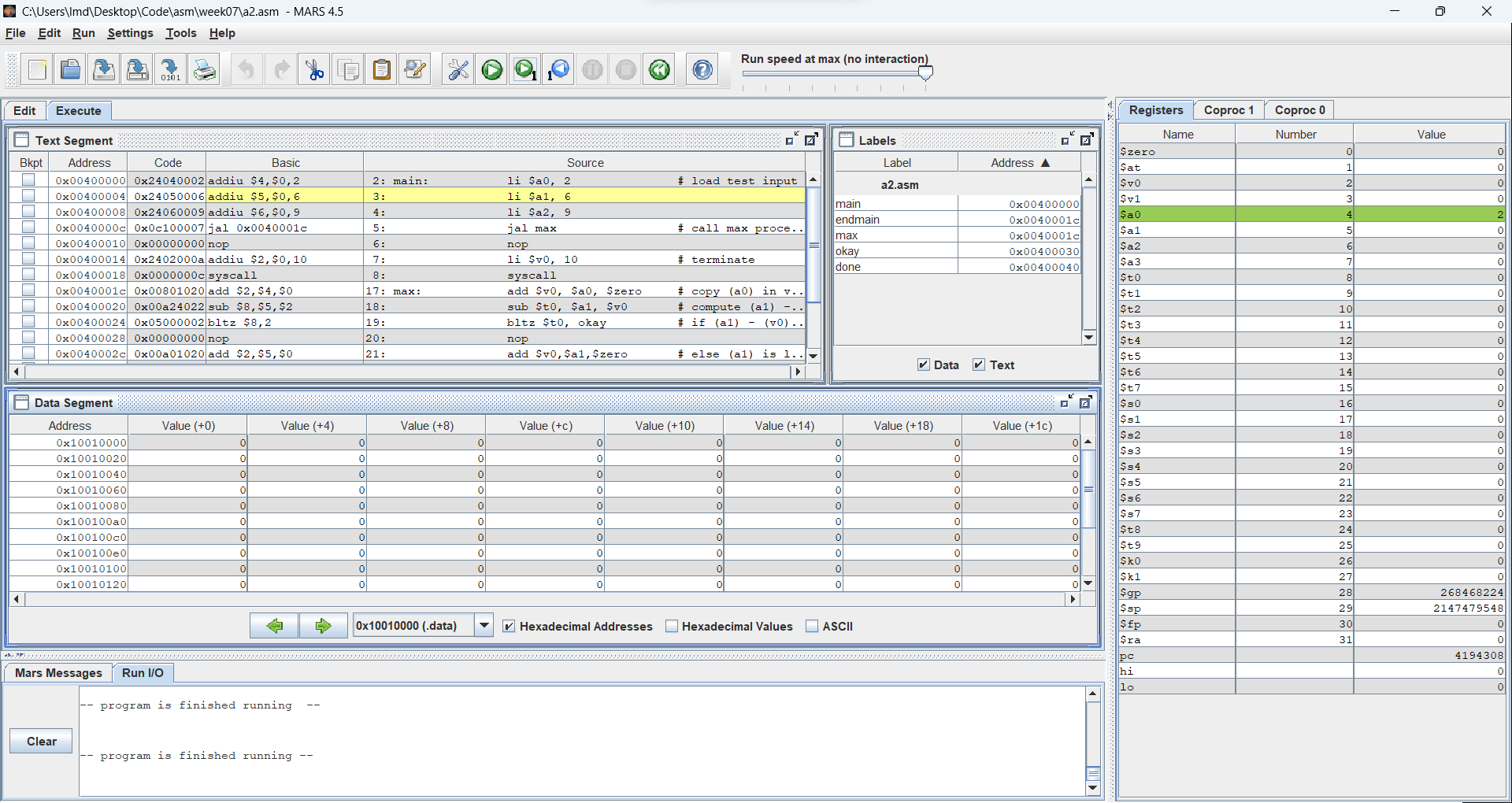
Description automatically generated with medium confidence

## Assignment 2

**Results:**

**Graphical user interface, application

Description automatically generated**

****

**Graphical user interface, application, table

Description automatically generated**

**Graphical user interface, application, table

Description automatically generated**

**Graphical user interface, application, table

Description automatically generated**

**Graphical user interface, application, table, Excel

Description automatically generated**

**Graphical user interface, application, table

Description automatically generated**

**Graphical user interface, application, table

Description automatically generated**

**Graphical user interface, application, table

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

## 

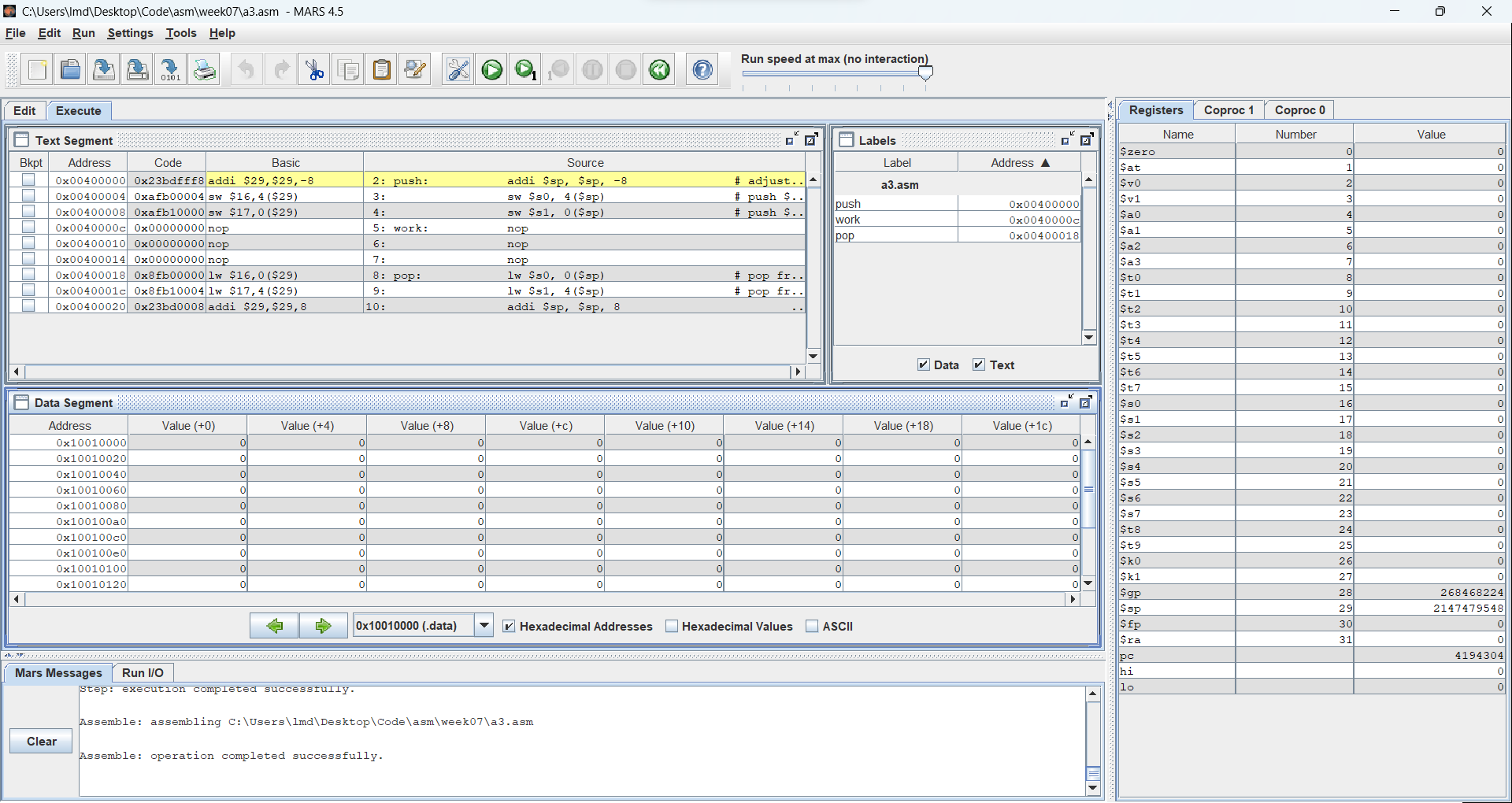
## **Graphical user interface, application Description automatically generated**

Graphical user interface, application

Description automatically generated

## Assignment 3

**Results:**

****

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application

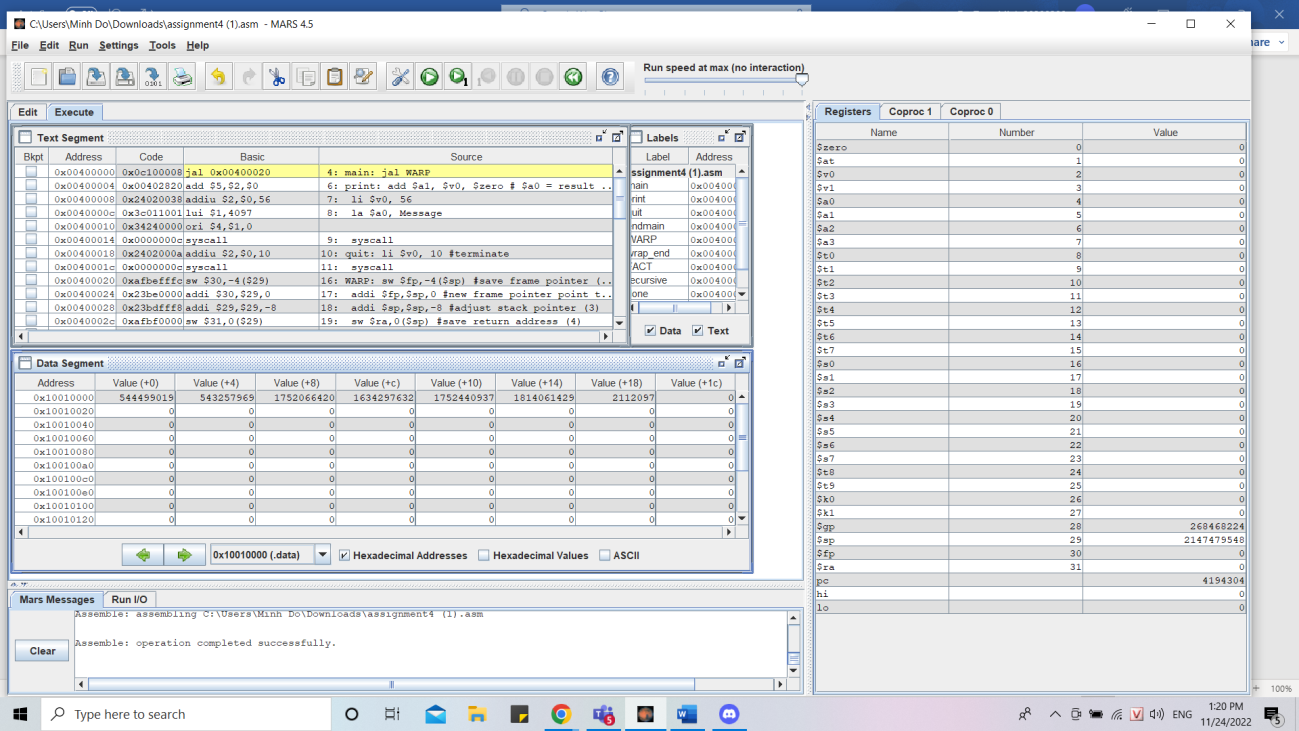
Description automatically generated**

## Assignment 4

**Comments:**

* N = 3 , get 3!

Assemble & Run:



* Results:

Graphical user interface, application, table, Excel

Description automatically generated

- Giải thích từng bước :

- Chương trình đầu tiên nhảy vào procedure, tạo stack lưu vào thanh ghi sp

chứa các giá trị

- Thêm giá trị của n=3 vào thanh ghi a0, tạo bộ nhớ cho thanh ghi địa chỉ

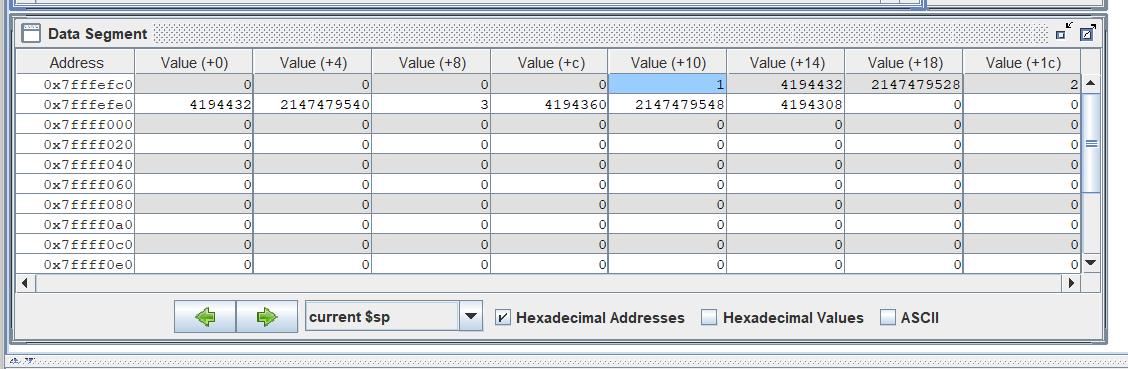
- Lưu các giá trị 2 vào thanh ghi a0, lúc này giá trị ba được lưu trong địa chỉ sau, bên phải là địa chỉ trả về

Table

Description automatically generated with medium confidence

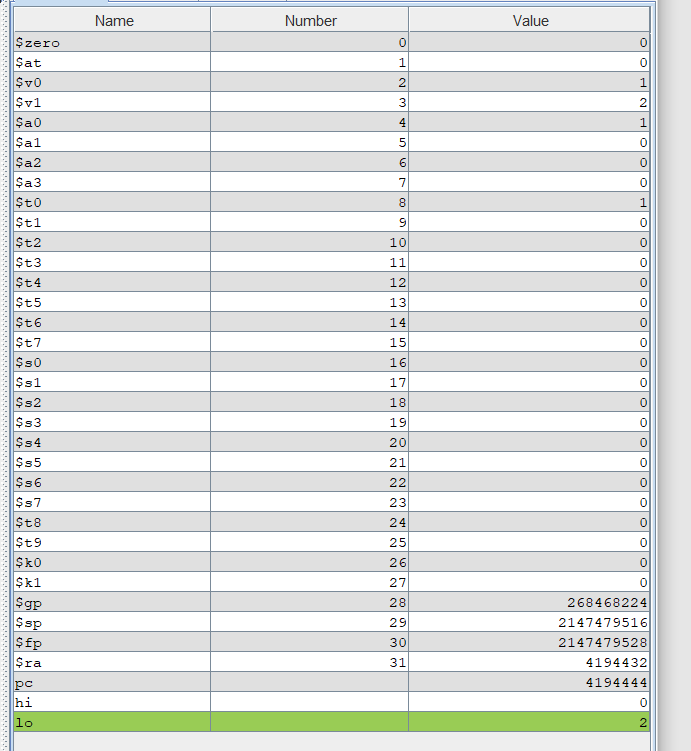
---

Tương tự ta có :



- 3 ô cạnh nhau lần lượt lưu giá trị thanh ghi a0 , địa chỉ trả về, địa chỉ fp sau mỗi vòng lặp

- Sau đó lưu lần lượt 2 giá trị cạnh nhau vào thanh ghi v0 ,v1, thực hiện phép nhân lưu vào thanh ghi lo



---

Tiến hành lưu lại giá trị thanh ghi lo vào thanh ghi v0, cập nhập giá trị tiếp

theo cần nhân vào thanh ghi v1, rồi tiến hành như trên đến khi giá trị v1

bằng n (với ở đây là bằng 3).Lúc này thì in ra giá trị.

---

Graphical user interface, table

Description automatically generated

Kết thúc chương trình

## Assignment 5

**Code:**

.data

Greatest: .asciiz "Greatest: "

Smallest: .asciiz "Smallest: "

Location: .asciiz ", Register: "

.text

mainInit: li $s0, 8

li $s1, 7

li $s2, 6

li $s3, 5

li $s4, 4

li $s5, 3

li $s6, 2

li $s7, 9

push: addi $sp, $sp, -32 # adjust the stack pointer

sw $s0, 28($sp) # push $s0 to stack

sw $s1, 24($sp) # push $s1 to stack

sw $s2, 20($sp) # push $s2 to stack

sw $s3, 16($sp) # push $s3 to stack

sw $s4, 12($sp) # push $s4 to stack

sw $s5, 08($sp) # push $s5 to stack

sw $s6, 04($sp) # push $s6 to stack

sw $s7, 00($sp) # push $s7 to stack

loopInit: li $s0, -100000 # s0 stores the greatest value

li $s1, -1 # s1 stores the location of the greatest value

li $s2, +100000 # s2 stores the smallest value

li $s3, -1 # s3 stores the location of the smallest value

li $s4, 7 # current index of the stack's top

loop: beq $sp, 0x7fffeffc, endLoop # while stack isn't empty

lw $t0, 00($sp) # get the top of the stack

blt $s0, $t0, update1

afterU1: bgt $s2, $t0, update2

afterU2 addi $sp, $sp, +4 # pop the top of the stack

addi $s4, $s4, -1 # update the top's index

j loop

update1: add $s0, $zero, $t0 # update greatest

add $s1, $zero, $s4

j afterU1

update2: add $s2, $zero, $t0 # update smallest

add $s3, $zero, $s4

j afterU2

endLoop:

printGreatest: li $v0, 4

la $a0, Greatest

syscall

li $v0, 1

add $a0, $zero, $s0

syscall

li $v0, 4

la $a0, Location

syscall

li $v0, 1

add $a0, $zero, $s1

syscall

li $v0, 11

li $a0, '\n'

syscall

printSmallest: li $v0, 4

la $a0, Smallest

syscall

li $v0, 1

add $a0, $zero, $s2

syscall

li $v0, 4

la $a0, Location

syscall

li $v0, 1

add $a0, $zero, $s3

syscall

li $v0, 11

li $a0, '\n'

syscall

**Comments:**

* The idea is to create a stack to store the values of 8 registers and using pop() to traverse through them quickly.
* By using the stack to loop, we don’t have to write 8 separate IF statements to update the largest and the greatest register.
* By pushing registers’ values to stack, we can reuse registers to store the final results.

**Results:**

* The sample list in the code was [8, 7, 6, 5, 4, 3, 2, 9].
* The greatest value was 9, in $s7.
* The smallest value was 2, in $s6.

Text

Description automatically generated