**Báo cáo thực hành KTMT tuần 3**

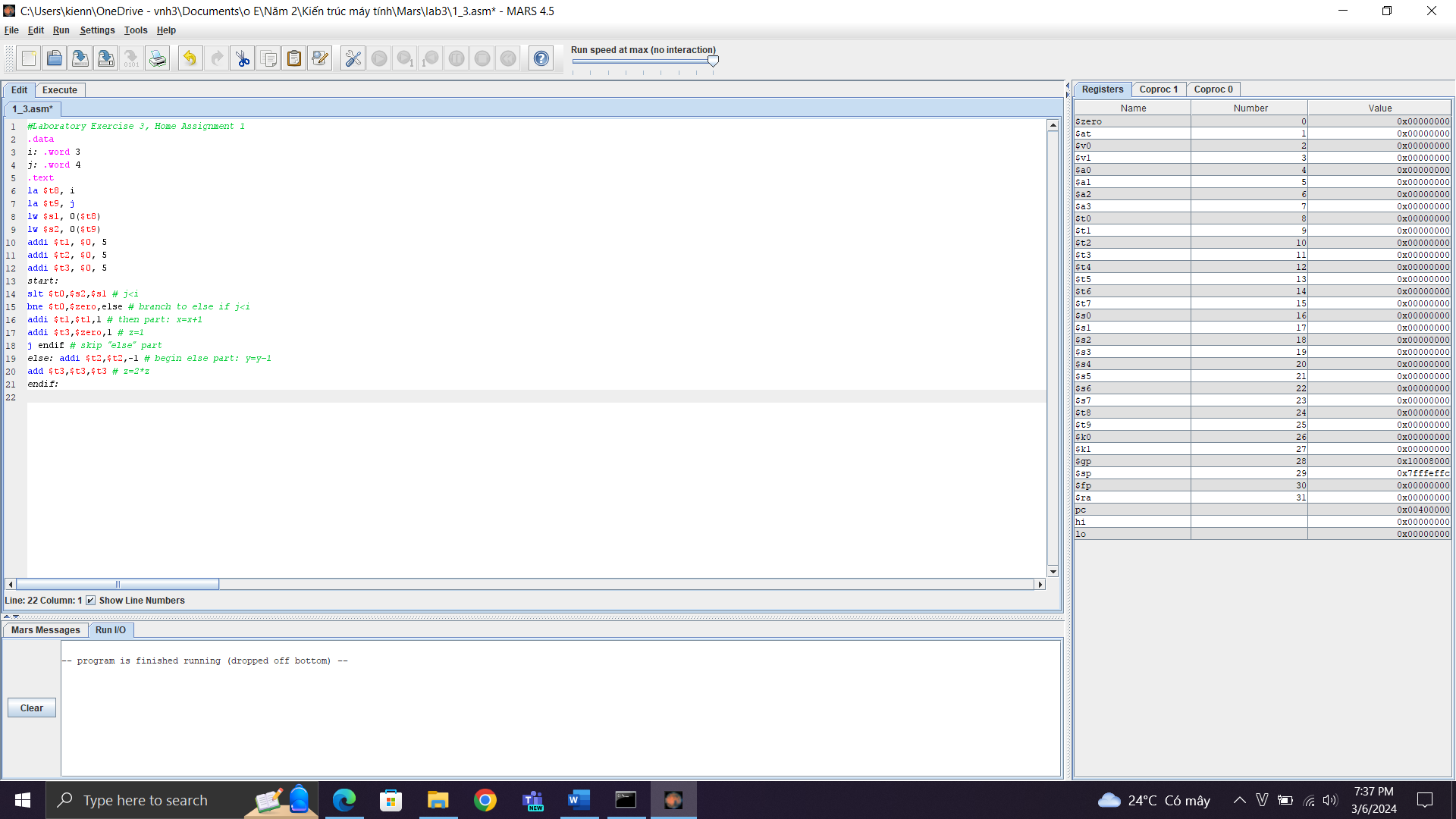
Họ và tên: Nguyễn Trung Kiên

MSSV: 20226110

**Assignment 1:**

Ta khởi tạo x=5,y=5,z=5 và các biến i,j theo từng trường hợp sau:

***Trường hợp 1: i=3, j=4 (i<j)***



**Giải thích:**

-$at thay đổi mỗi khi gọi lệnh la.

-$t8, $t9 lưu địa chỉ của 2 biến i, j

-$s1, $s2 được ghi giá trị của biến i, j

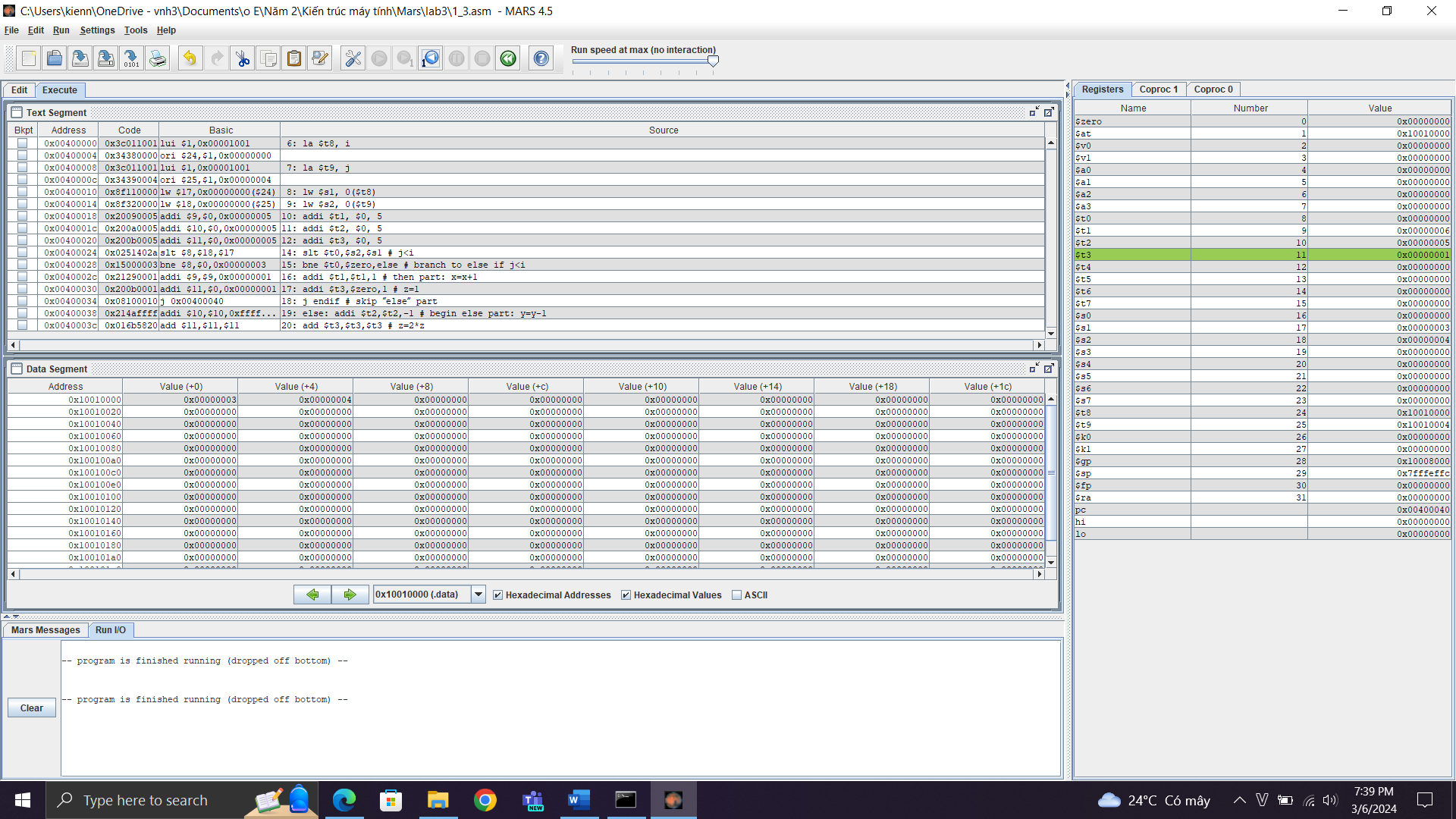
-$t1, $t2, $t3 được ghi giá trị của biến x, y, z

-$t0 lưu kết quả của slt (ở đây trả về giá trị 0 vì điều kiện j<i ko thỏa mãn)

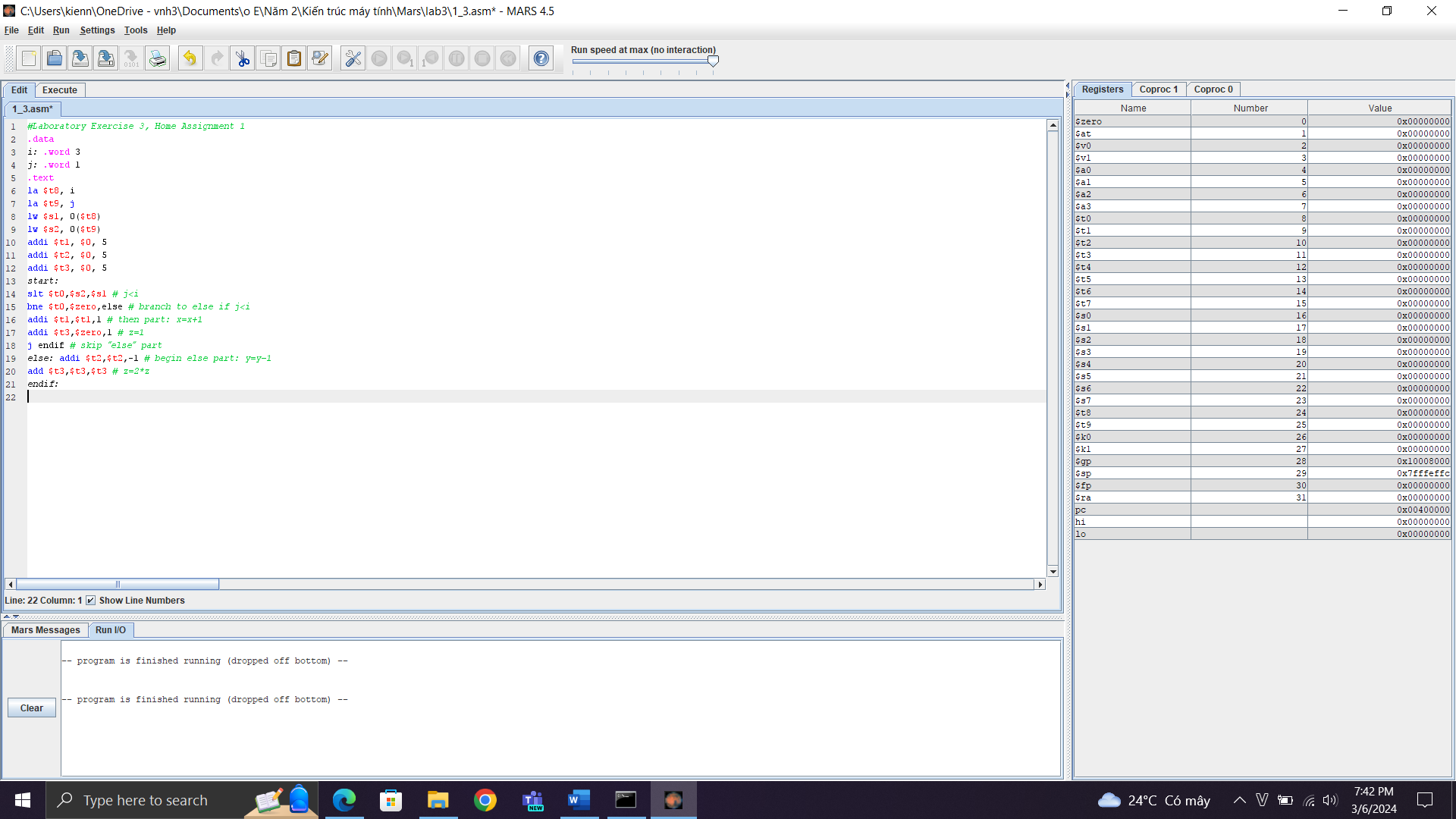
-$t1 tăng 1 đơn vị, $t3 được gán bằng 1

-pc thay đổi sau mỗi lệnh

=> Kết quả sau đúng với lý thuyết



***Trường hợp 2: i=3, j=1 (i>j)***



**Giải thích:**

-$at thay đổi mỗi khi gọi lệnh la

-$t8, $t9 lưu địa chỉ của 2 biến i, j

-$s1, $s2 được ghi giá trị của biến i, j

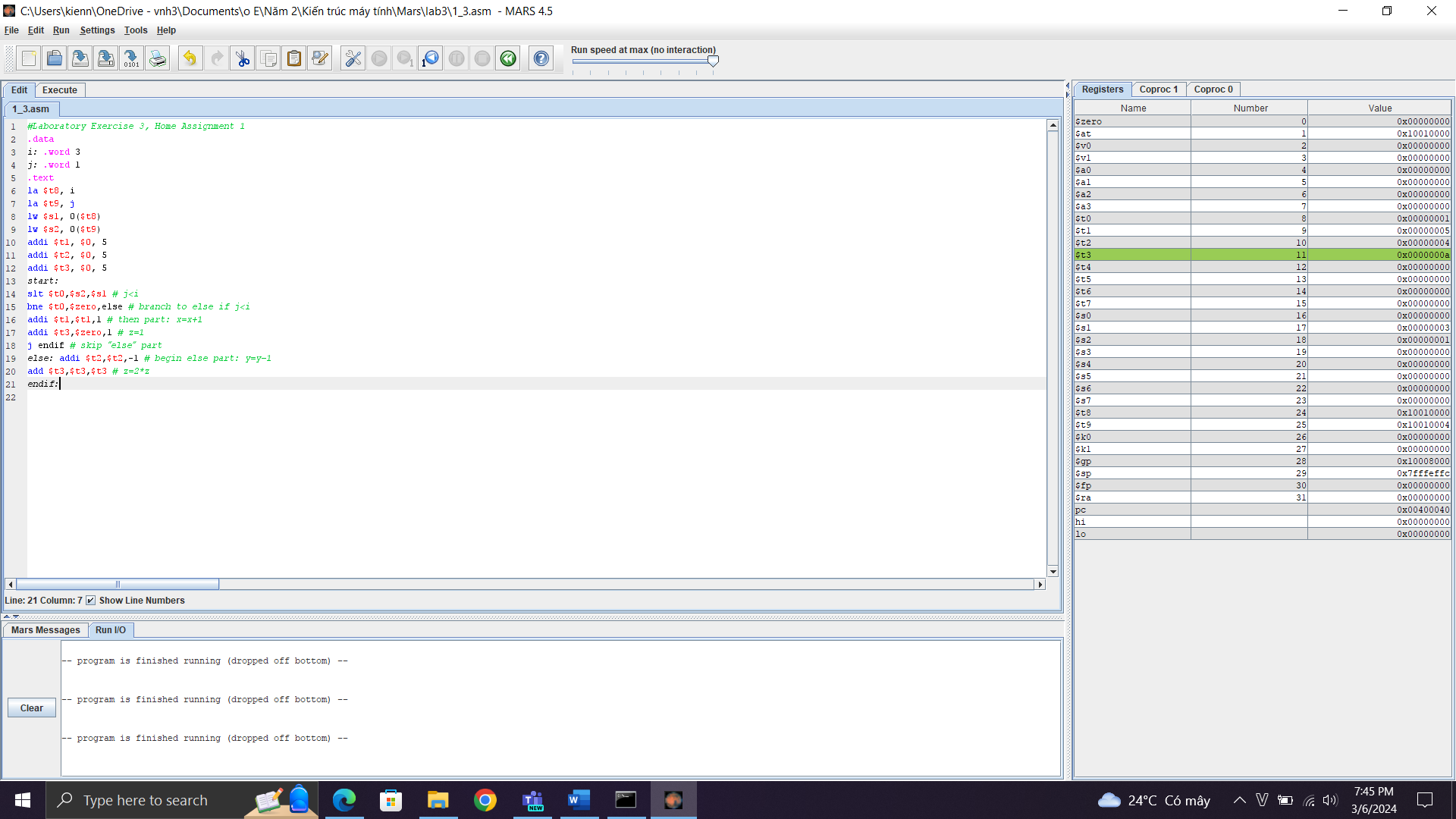
-$t1, $t2, $t3 được ghi giá trị của biến x, y, z

-$t0 lưu kết quả của slt (ở đây trả về giá trị 1 vì điều kiện j<i thỏa mãn)

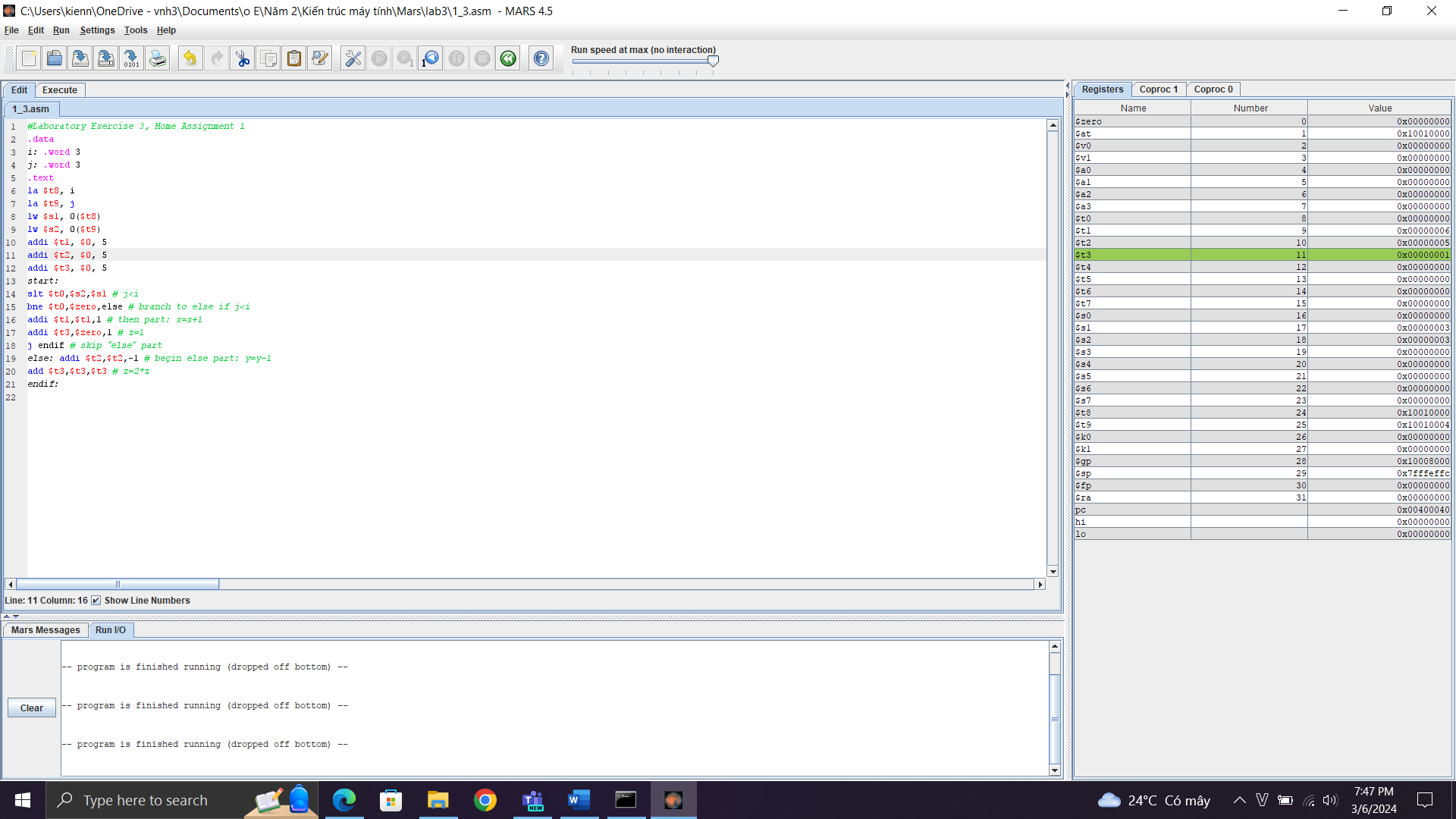
-$t1 giảm 1 đơn vị, $t3 được gán bằng giá trị gấp đôi của chính nó

-pc thay đổi sau mỗi lệnh

=> Kết quả đúng với lý thuyết



***Trường hợp 3: i=j=3 (i=j)***



**Giải thích:**

-$at thay đổi mỗi khi gọi lệnh la

-$t8, $t9 lưu địa chỉ của 2 biến i, j

-$s1, $s2 được ghi giá trị của biến i, j

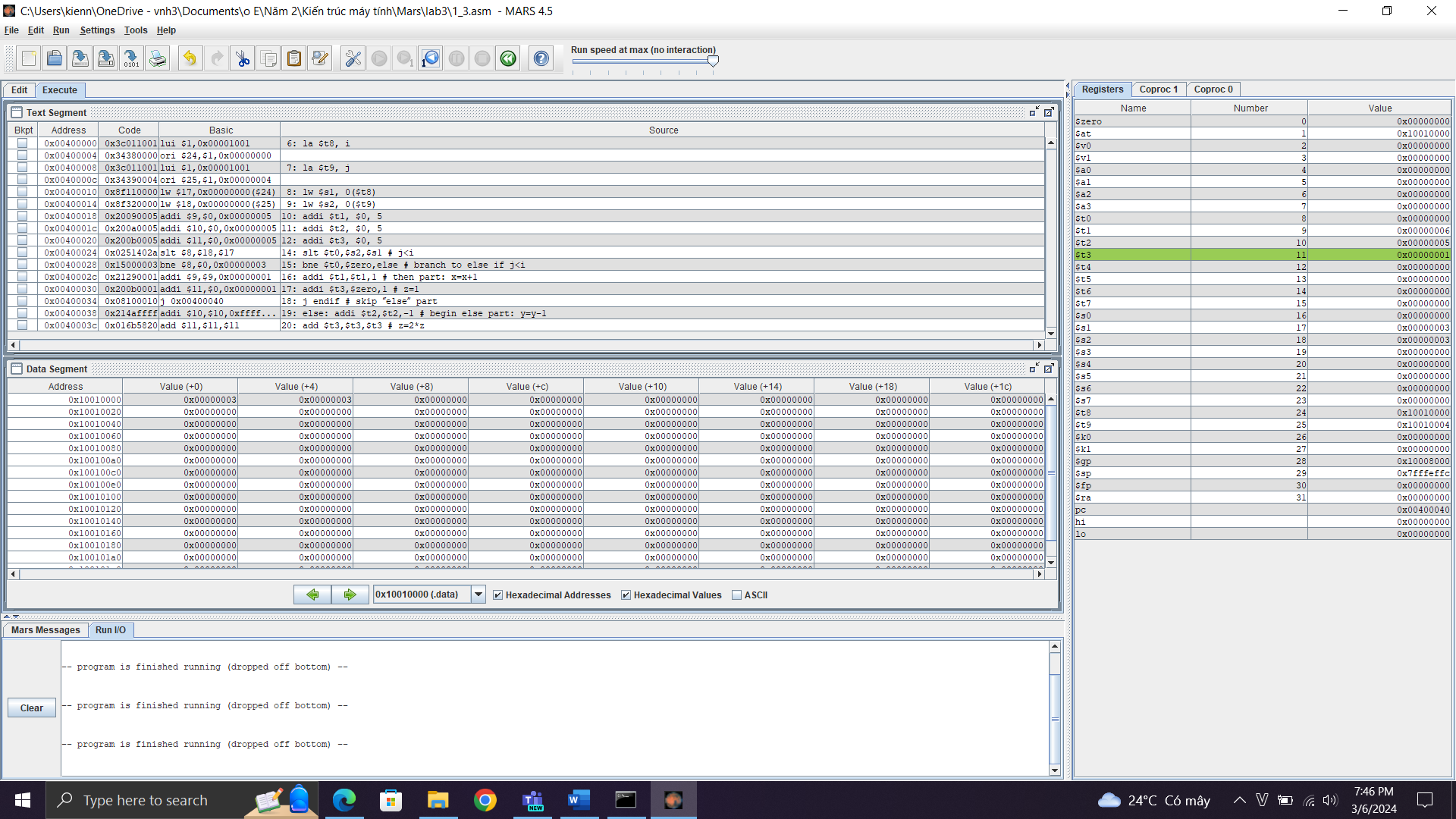
-$t1, $t2, $t3 được ghi giá trị của biến x, y, z

-$t0 lưu kết quả của slt (ở đây trả về giá trị 0 vì điều kiện j<i ko thỏa mãn)

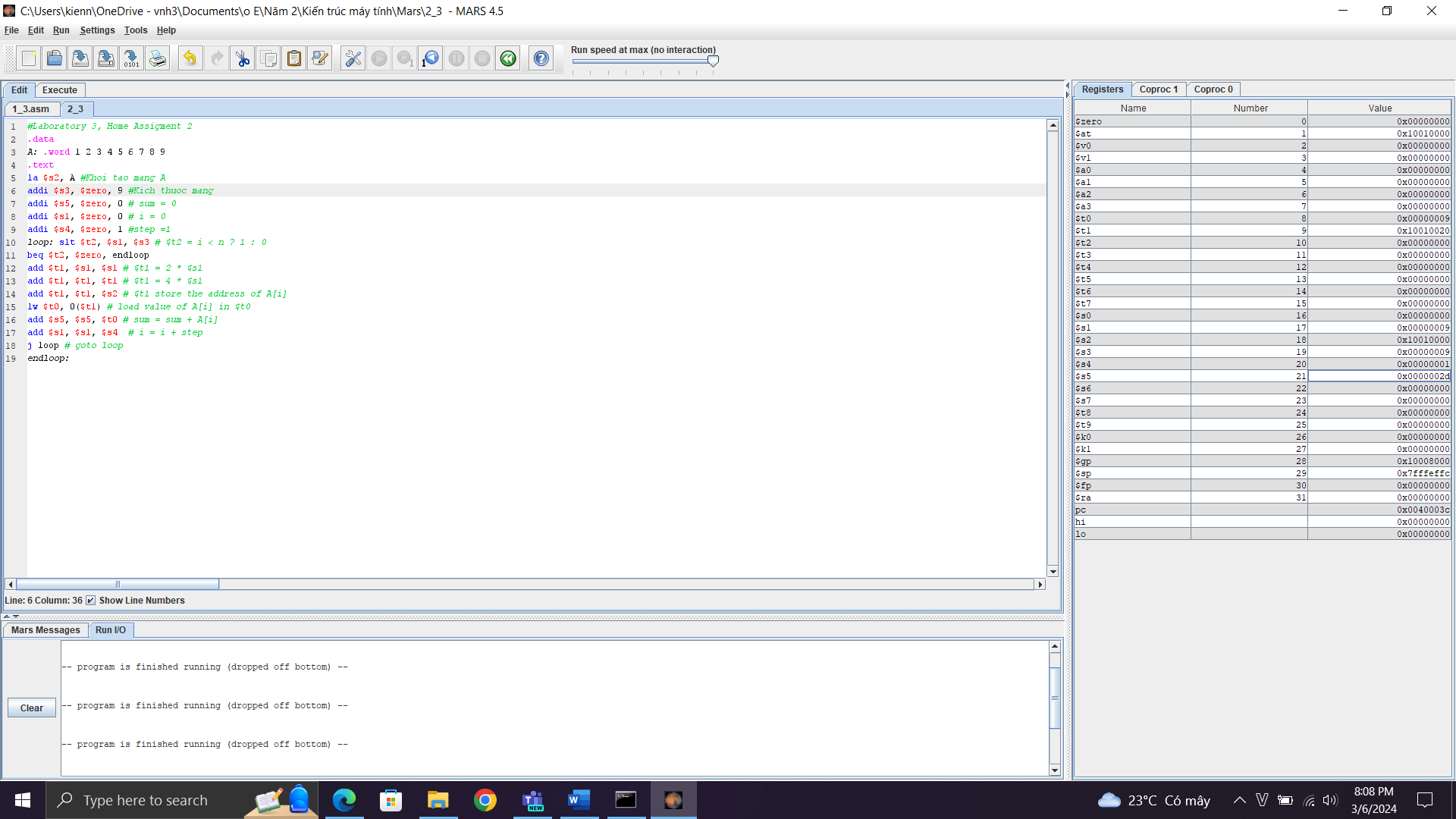
-$t1 tăng 1 đơn vị, $t3 được gán bằng 1

- Thanh ghi pc thay đổi sau mỗi lệnh.

=> Kết quả đúng với lý thuyết



**Assignment 2:**



-Thanh ghi $s1, $s2, $s3, $s4, $s5 có công dụng như trên

-Thanh ghi $t1 lưu địa chỉ của A[i]

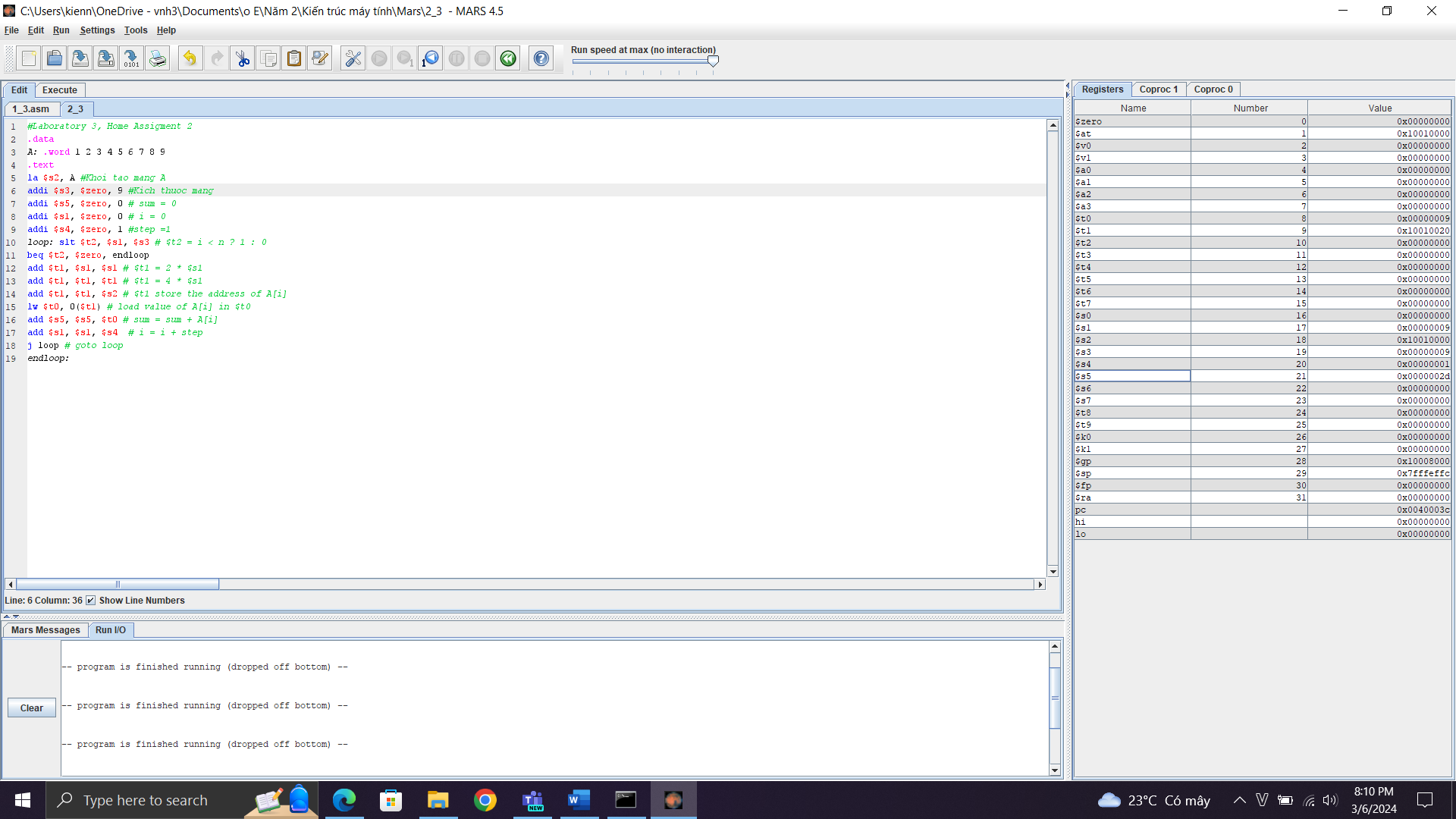
-Thanh ghi $t0 lưu giá trị của A[i]

-Thanh ghi $s5 tăng giá trị bằng A[i] sau mỗi vòng lặp

-Thanh ghi $s1(i) tăng giá trị 1 lượng bằng $s4 (bước nhảy)

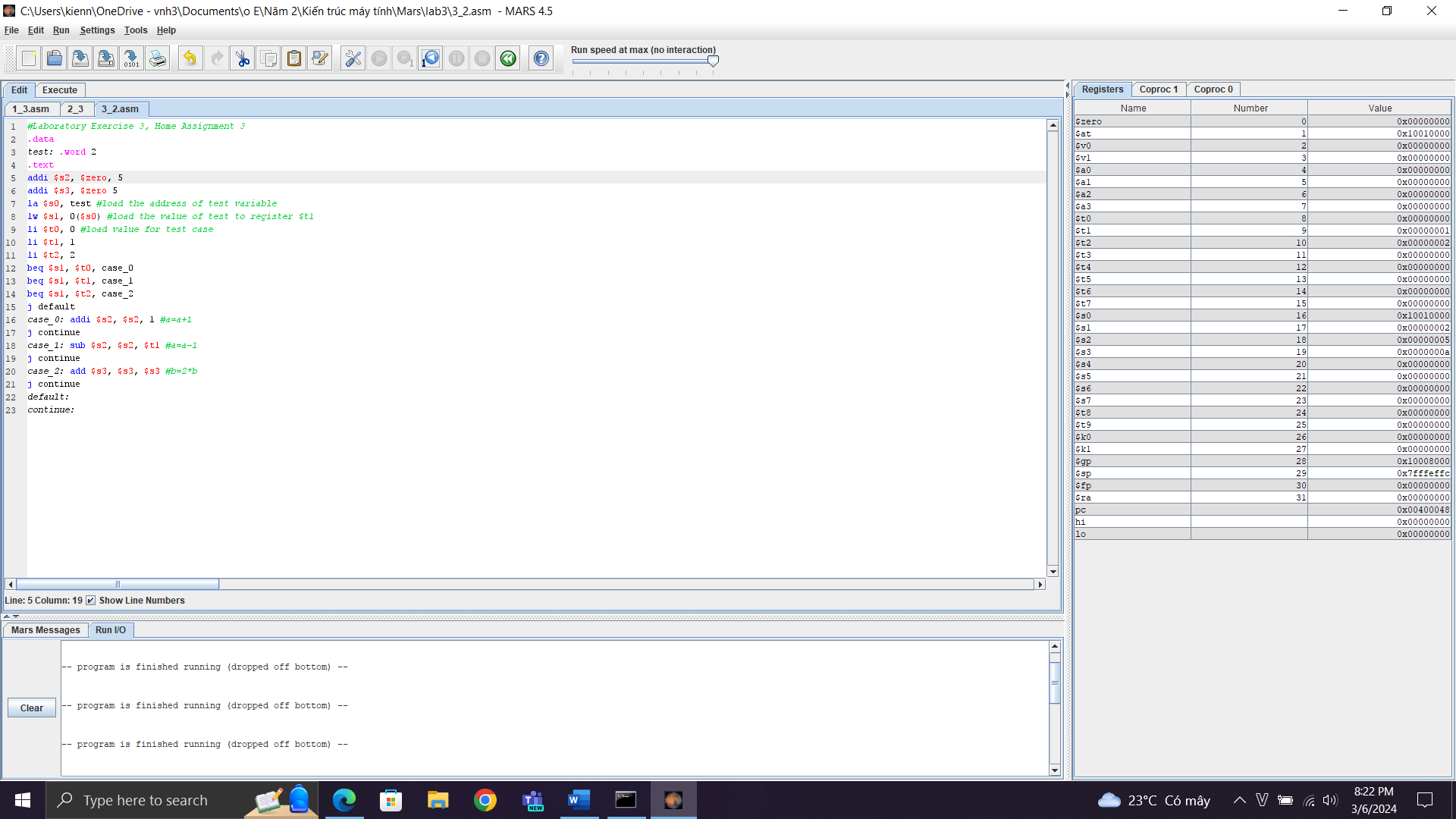
-Thanh ghi pc thay đổi sau mỗi lệnh

=> Kết quả của thanh ghi $s5 là 0x0000002d, tức là sum = 45, đúng với lý thuyết



**Assignment 3:**

Ta khởi tạo a=10 và b=10



-$s2, $s3 lưu giá trị khởi tạo của a, b

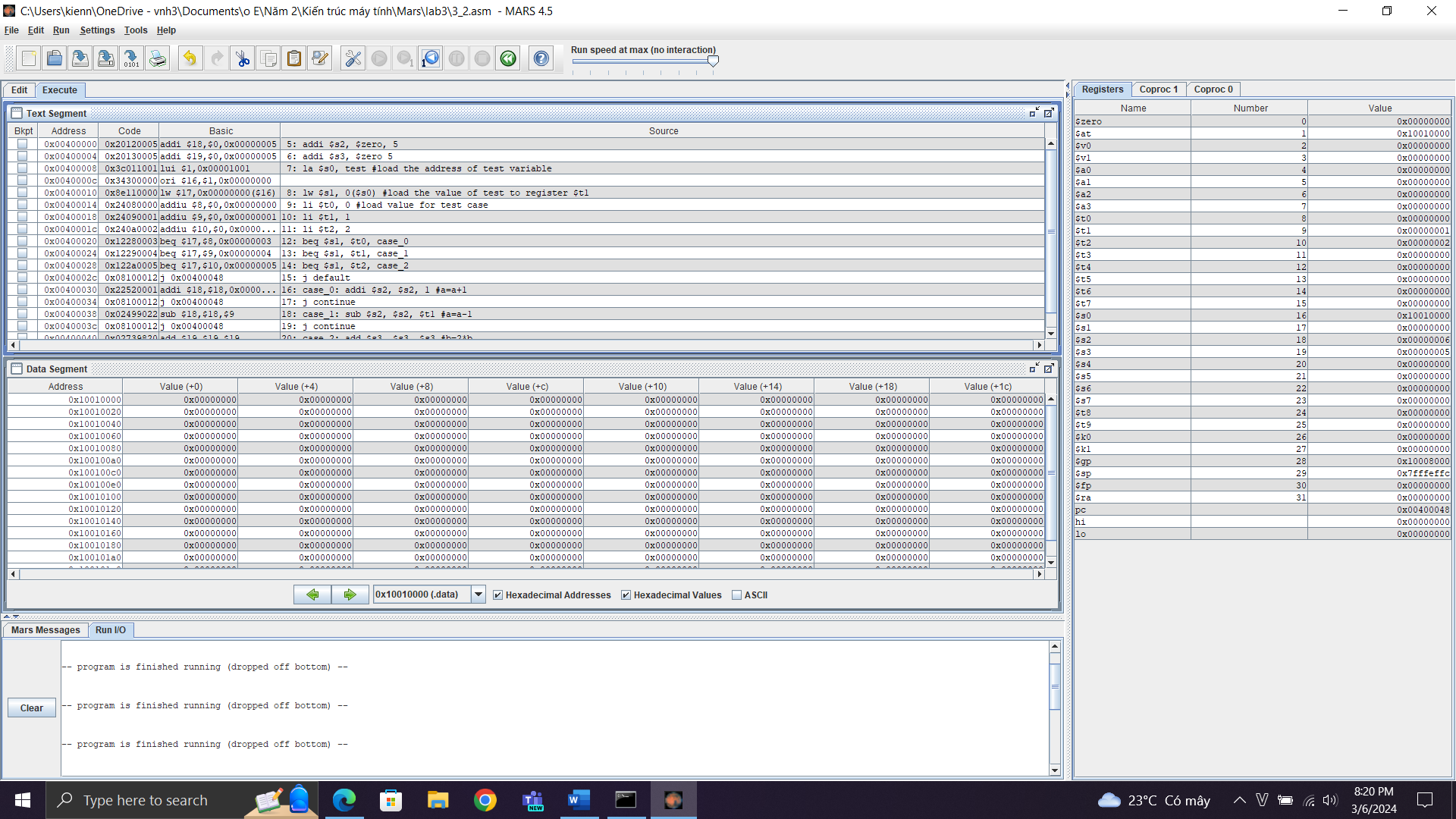
-$s0 lưu địa chỉ của biến test

-$s1 lưu giá trị của biến test

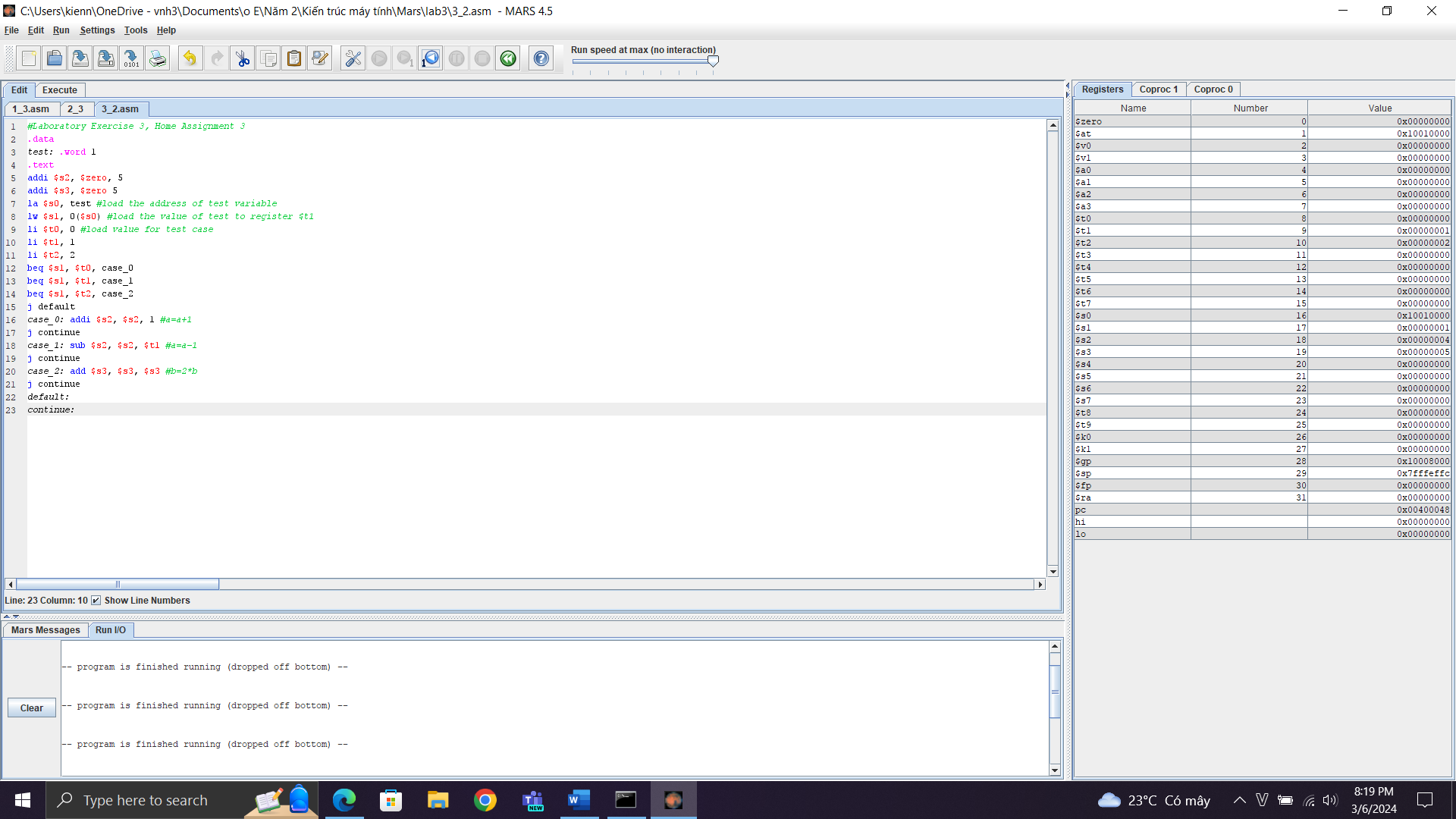
-$t0, $t1, $t2 lưu các case để biến test có thể thực hiện so sánh

-Nếu:

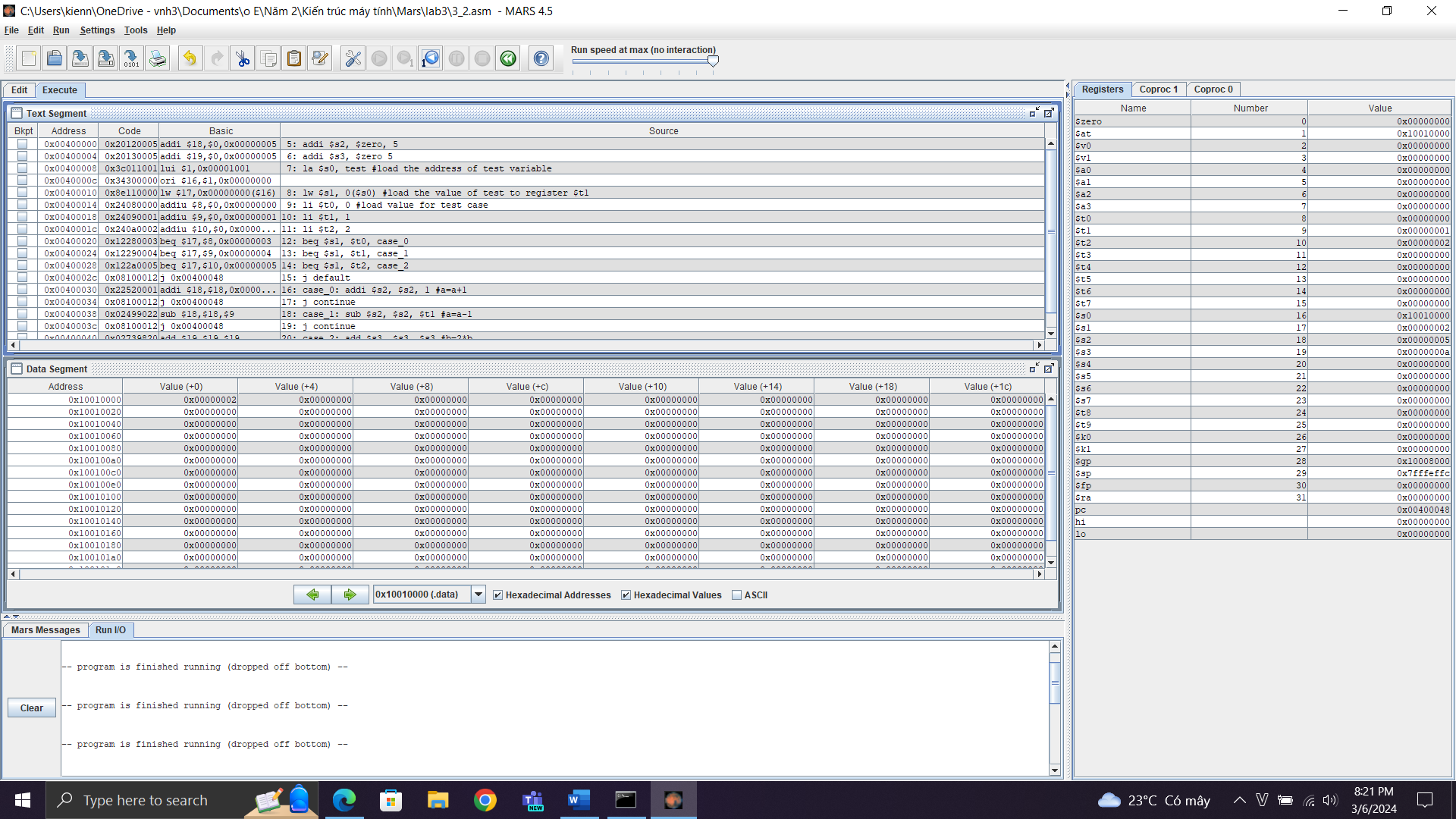
+Test = 0: a=6



+Test = 1: a=4



+Test = 2: b=10



* Kết quả chạy đúng với chương trình

**Assignment 4:**

1. ***i < j***

.data

i: .word 2

j: .word 3

.text

la $t8, i

la $t9, j

lw $s1, 0($t8) # i=2

lw $s2, 0($t9) # j=3

addi $t1, $0, 2 # x=2

addi $t2, $0, 2 # y=2

addi $t3, $0, 2 # z=2

start:

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

beq $t0,$zero,else # branch to else if i >= j

#Thay lệnh bne thành beq

# kiểm tra xem kết quả của so sánh j($s2) < i($s1), nếu sai gán $t0=0

# nếu $t0 khác 0 rẽ nhảy đến else

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***

.data

i: .word 2

j: .word 3

.text

la $t8, i

la $t9, j

lw $s1, 0($t8) # i=2

lw $s2, 0($t9) # j=3

addi $t1, $0, 2 # x=2

addi $t2, $0, 2 # y=2

addi $t3, $0, 2 # z=2

start:

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

bne $t0,$zero,else # branch to else if i<j

# kiểm tra xem kết quả của so sánh j($s2) < i($s1), nếu sai gán $t0=0

# nếu $t0 khác 0 rẽ nhảy đến else

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***

.data

i: .word 2

j: .word 3

.text

la $t8, i

la $t9, j

lw $s1, 0($t8) # i=2

lw $s2, 0($t9) # j=3

addi $t1, $0, 2 # x=2

addi $t2, $0, 2 # y=2

addi $t3, $0, 2 # z=2

add $t7, $s1, $s2 # tổng của i và j

start:

slt $t0,$0,$t7 # 0 < i+j

bne $t0,$zero,else # branch to else if 0 >= i+j

# kiểm tra xem kết quả của so sánh j($s2) < i($s1), nếu sai gán $t0=0

# nếu $t0 khác 0 rẽ nhảy đến else

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***

.data

i: .word 2

j: .word 3

.text

la $t8, i

la $t9, j

lw $s1, 0($t8) # i=2

lw $s2, 0($t9) # j=3

addi $t1, $0, 2 # x=2

addi $t2, $0, 2 # y=2

addi $t3, $0, 2 # z=2

addi $t4, $0, 5 # m = 5

addi $t5, $0, 6 # n = 6

add $t6, $t4, $t5 # m+n = 5+6

add $t7, $s1, $s2 # tổng của i và j

start:

slt $t0, $t6, $t7 # m+n< i+j thì đặt là 1

beq $t0, $zero, else # branch to else if m+n >= i+j

# kiểm tra xem kết quả của so sánh j($s2) < i($s1), nếu sai gán $t0=0

# nếu $t0 khác 0 rẽ nhảy đến else

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:

**Assignment 5:**

1. ***i <= n***

.data

A: .word 1 2 3 4 5 6 7 8 9 10 11

.text

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

la $s2, A # load address of A into $s2

addi $s3, $zero, 10 # n = 10

addi $s4, $zero, 1 # step = 1

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

loop:

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

bne $t2, $zero, endloop # if i > n then 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 -100 9 10

.text

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

la $s2, A # load address of A into $s2

addi $s3, $zero, 10 # n = 10

addi $s4, $zero, 1 # step = 1

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

loop:

bltz $s5, endloop # if sum < 0 then 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. ***A[i] != 0***

.data

A: .word 1 2 3 4 5 6 7 0 9 10

.text

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

la $s2, A # load address of A into $s2

addi $s3, $zero, 10 # n = 10

addi $s4, $zero, 1 # step = 1

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

loop:

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, $0, endloop # if a[i]=0 endloop

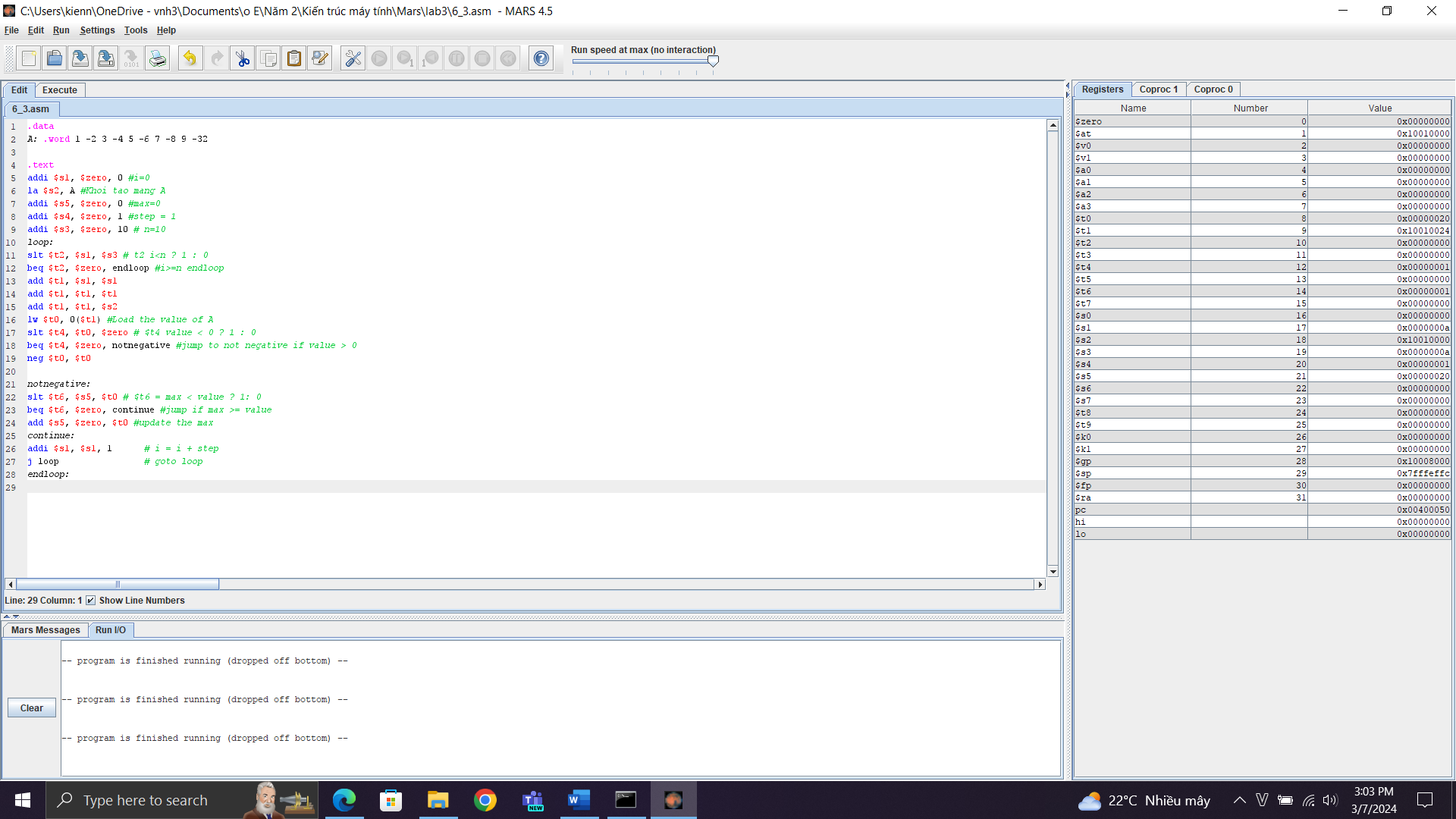
add $s5, $s5, $t0 # sum = sum + A[i]

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

j loop # goto loop

endloop:

**Assignment 6:**



.data

A: .word 1 -2 3 -4 5 -6 7 -8 9 -32

.text

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

la $s2, A #Khoi tao mang A

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

addi $s4, $zero, 1 #step = 1

addi $s3, $zero, 10 # n=10

loop:

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

beq $t2, $zero, endloop #i>=n endloop

add $t1, $s1, $s1

add $t1, $t1, $t1

add $t1, $t1, $s2

lw $t0, 0($t1) #Load the value of A

slt $t4, $t0, $zero # $t4 value < 0 ? 1 : 0

beq $t4, $zero, notnegative #jump to not negative if value > 0

neg $t0, $t0

notnegative:

slt $t6, $s5, $t0 # $t6 = max < value ? 1: 0

beq $t6, $zero, continue #jump if max >= value

add $s5, $zero, $t0 #update the max

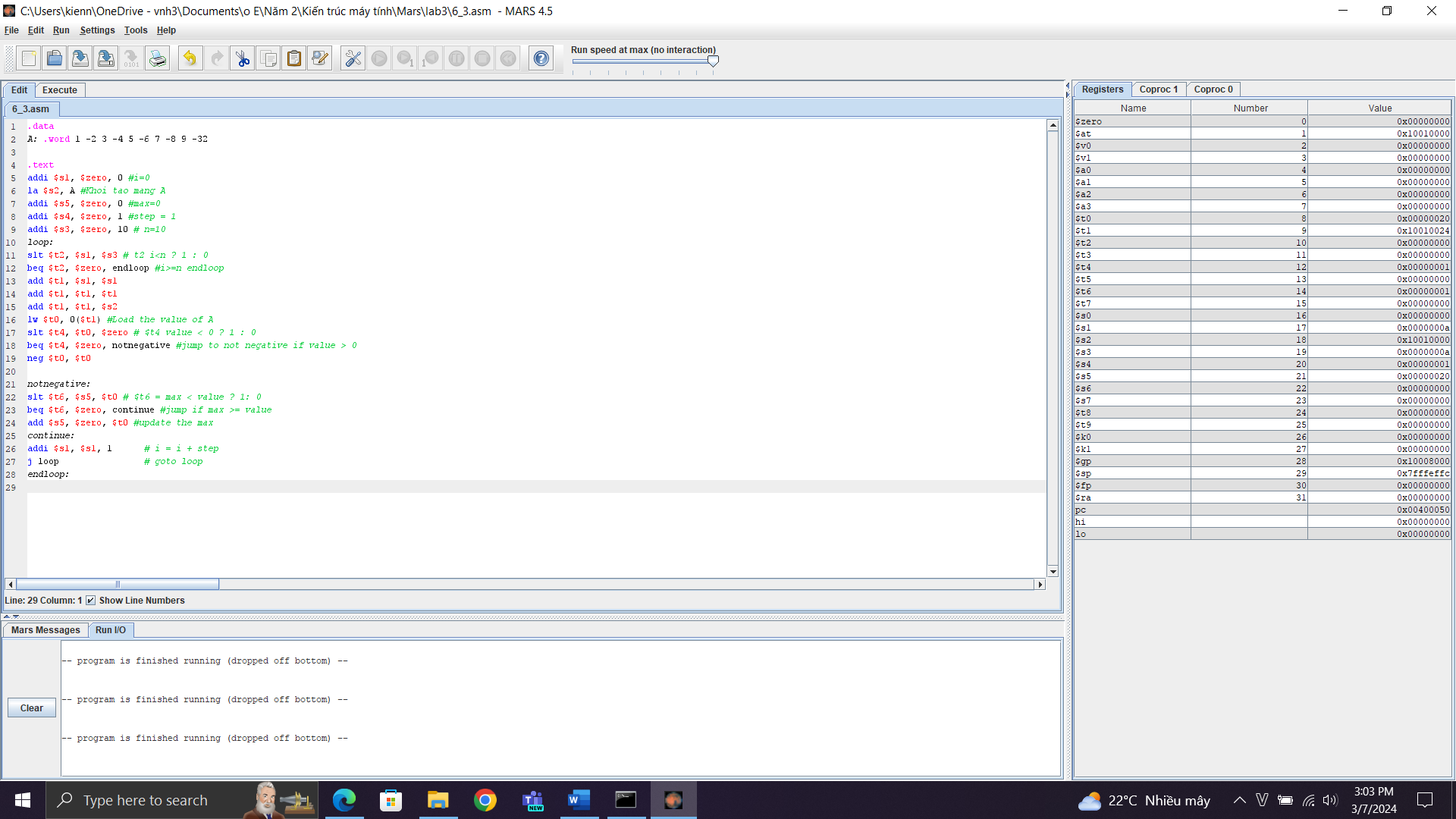
continue:

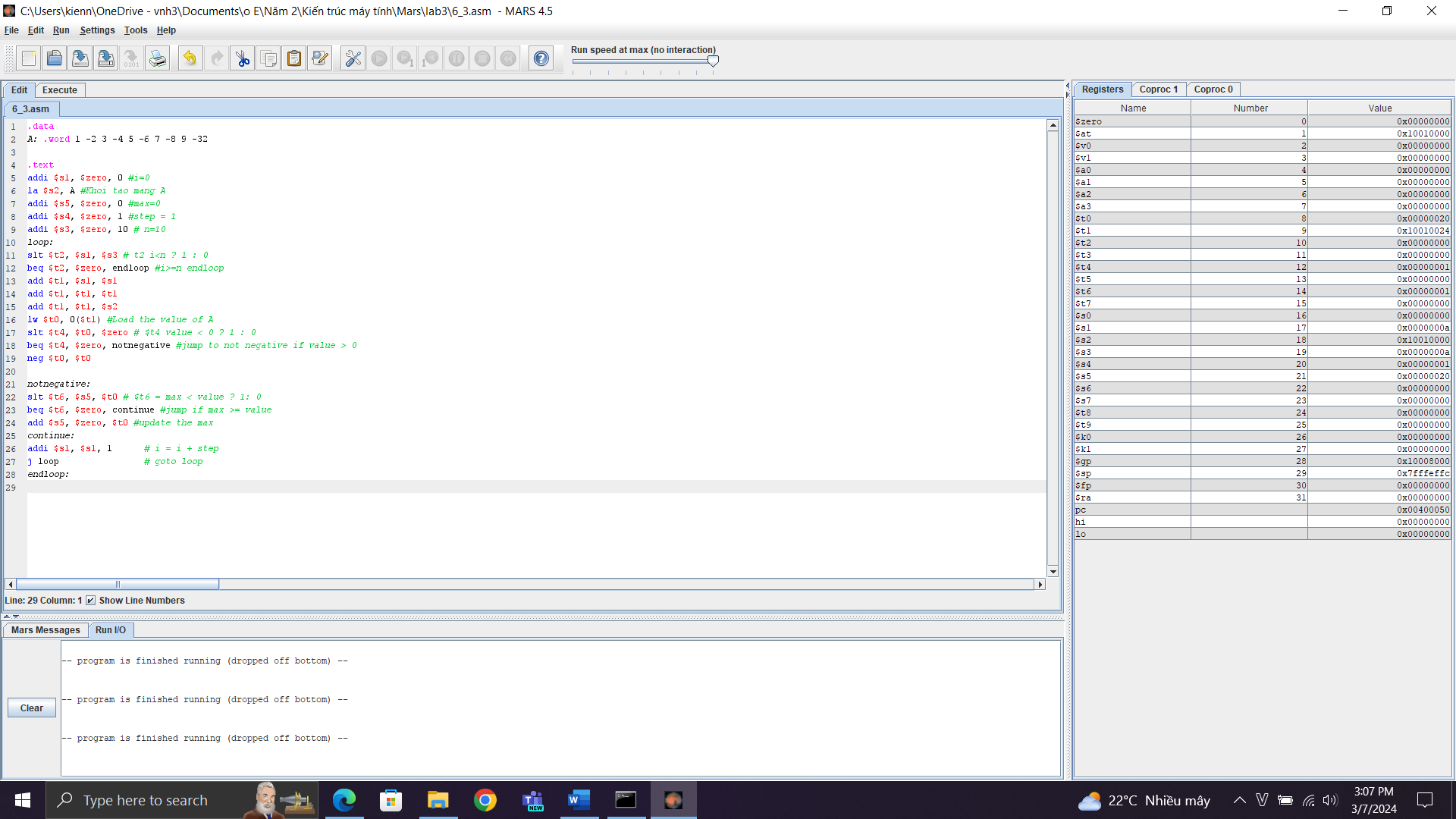
addi $s1, $s1, 1 # i = i + step

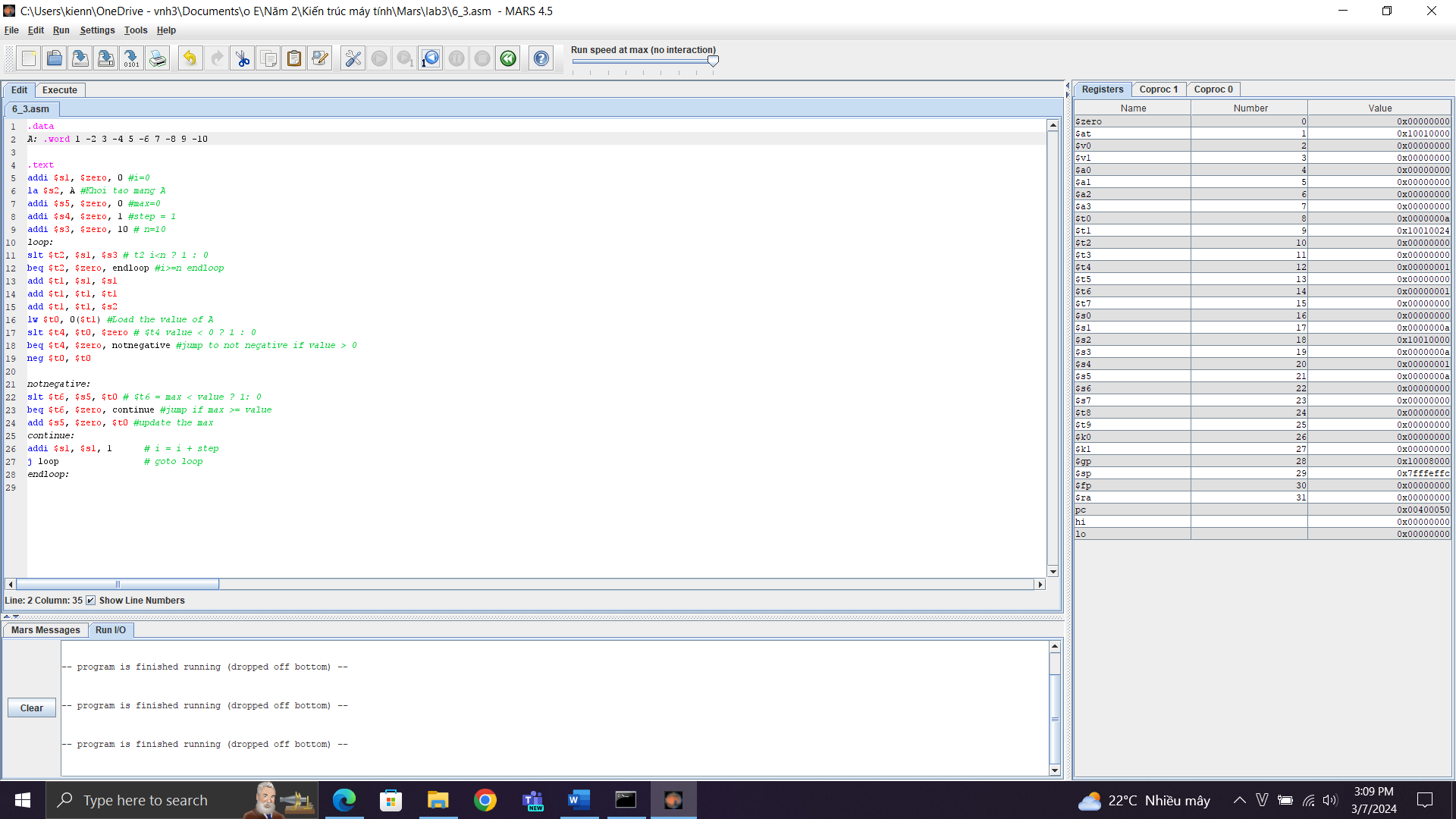
j loop # goto loop

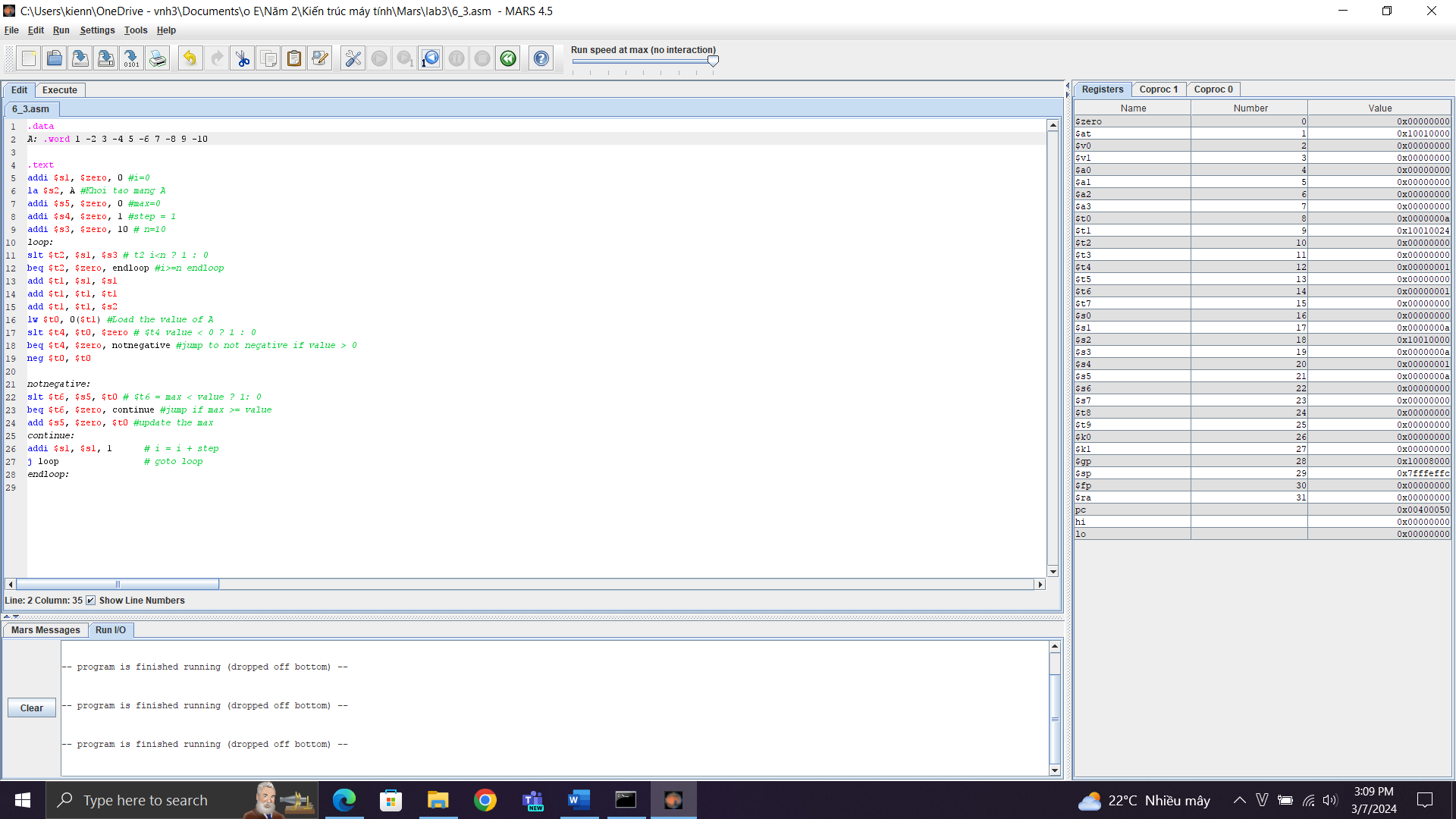
endloop:

* Ta lựa chọn các giá trị có giá trị tuyệt đối max cho mảng A[i] để kiểm tra chương trình.

+ Với max = 32. 

+ Giá trị thanh ghi $s5: 

+ Với max = 10. 

+ Giá trị thanh ghi $s5: 

* Kiểm tra thêm, có thể nhận xét là chương trình chạy đúng.