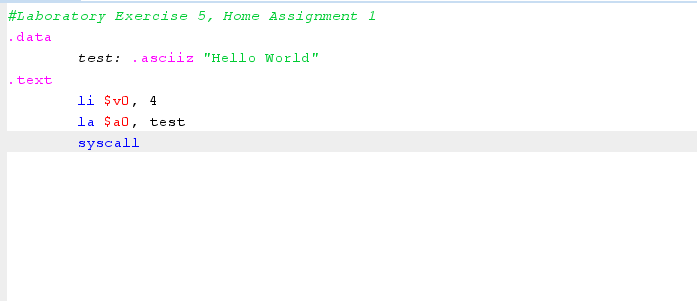
**BÁO CÁO THỰC HÀNH KTMT TUẦN 5**

**Họ tên: Nguyễn Đức Đại Dương**

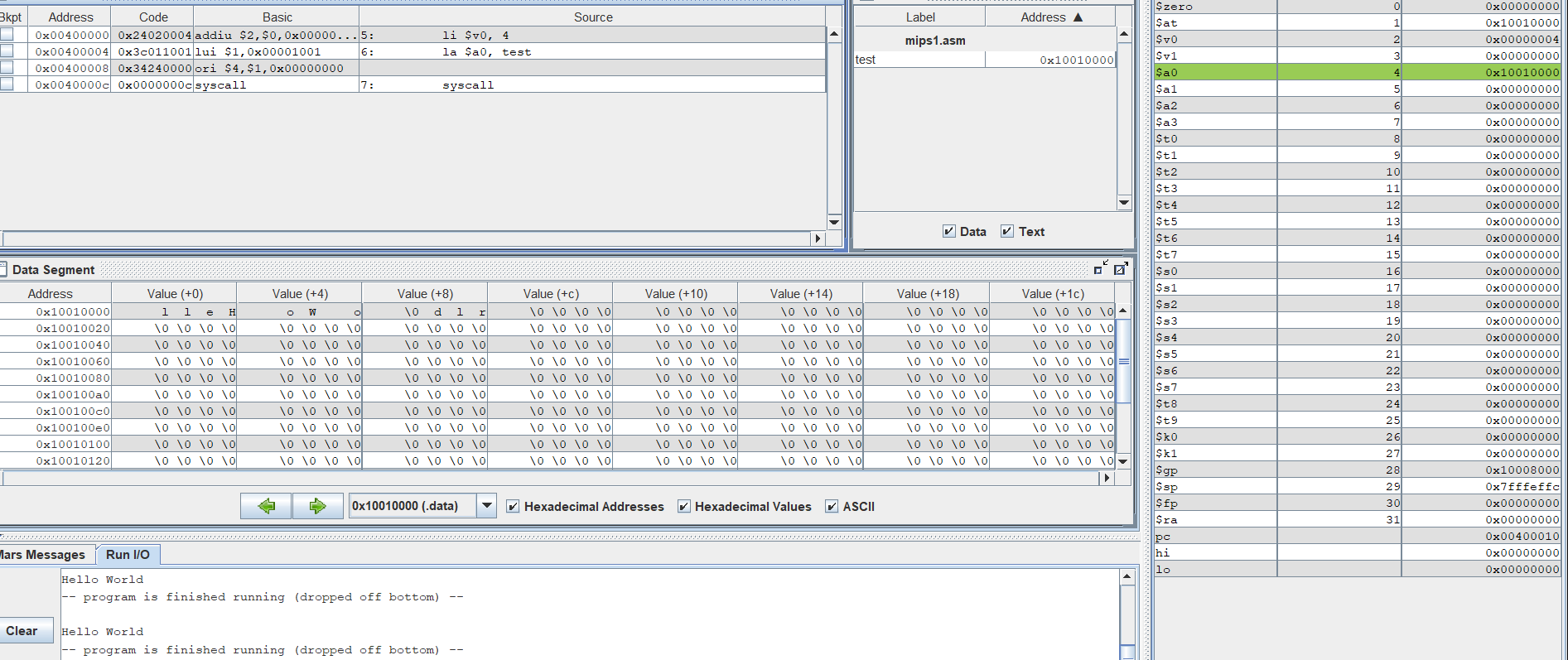
**MSSV: 20225616**

Assignment 1

Code:



Kết quả:



Nhận xét:

Kết quả đúng với lí thuyết

Assignment 2 :

Trường hợp tính tổng của 4 và 5 rồi in ra kết quả

Code:

.data

Message1: .asciiz "The sum of "

Message2: .asciiz " and "

Message3: .asciiz " is "

.text

li $s0, 4 # $s0 = 4

li $s1, 5 # $s1 = 5

add $s2,$s0,$s1 #$s2 = $s0+$s1

#print The sum of

li $v0, 4 #print string

la $a0, Message1

syscall

#print $s0

li $v0, 1 #print integer

move $a0, $s0

syscall

#print "and"

li $v0, 4 #print string

la $a0, Message2

syscall

#print $s1

li $v0, 1 #print integer

move $a0, $s1

syscall

#print "is"

li $v0, 4 #print string

la $a0, Message3

syscall

#print $s2

li $v0, 1 #print integer

move $a0, $s2

syscall

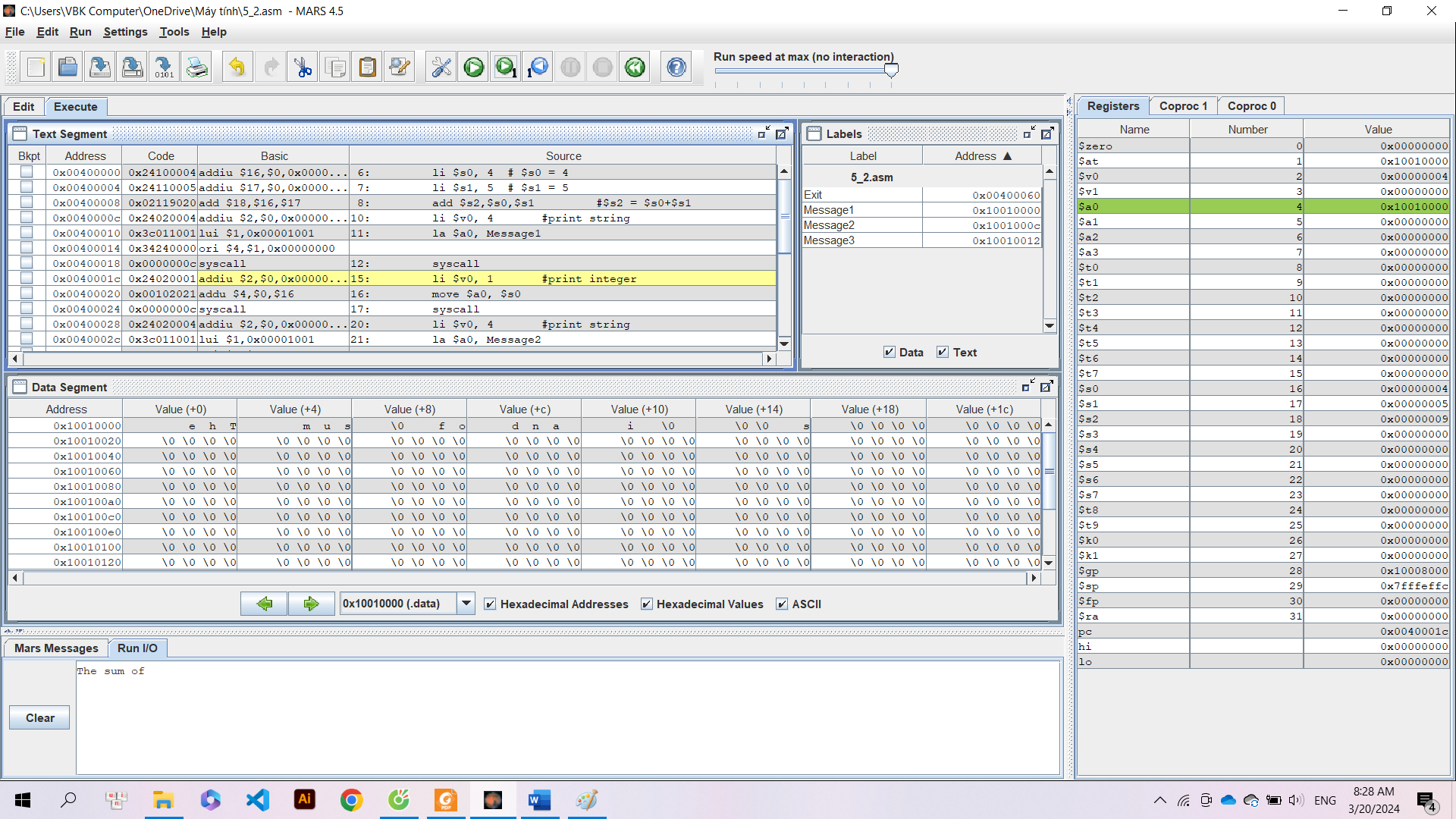
Exit:

li $v0, 10

syscall

Kết quả:

In xâu “The sum of”



In số 4

A screenshot of a computer

Description automatically generated

In xâu “and”

A screenshot of a computer

Description automatically generated

In số 5

A screenshot of a computer

Description automatically generated

In xâu “is”

A screenshot of a computer

Description automatically generated

In kết quả 9

A screenshot of a computer

Description automatically generated

=>Kết quả đúng với lí thuyết

Assignment 3

Code:

.data

x: .space 32 # destination string x, empty

y: .asciiz "Hello word" # source string y

.text

strcpy:

add $s0,$zero,$zero # $s0 = i = 0

la $a1, y #Load address of y to $a1

la $a0, x #Load address of x to $a0

L1:

add $t1,$s0,$a1 # $t1 = $s0 + $a1 = i + y[0]

# = address of y[i]

lb $t2,0($t1) # $t2 = value at $t1 = y[i]

add $t3,$s0,$a0 # $t3 = $s0 + $a0 = i + x[0]

# = address of x[i]

sb $t2,0($t3) # x[i]= $t2 = y[i]

beq $t2,$zero,end\_of\_strcpy # if y[i] == 0, exit

nop

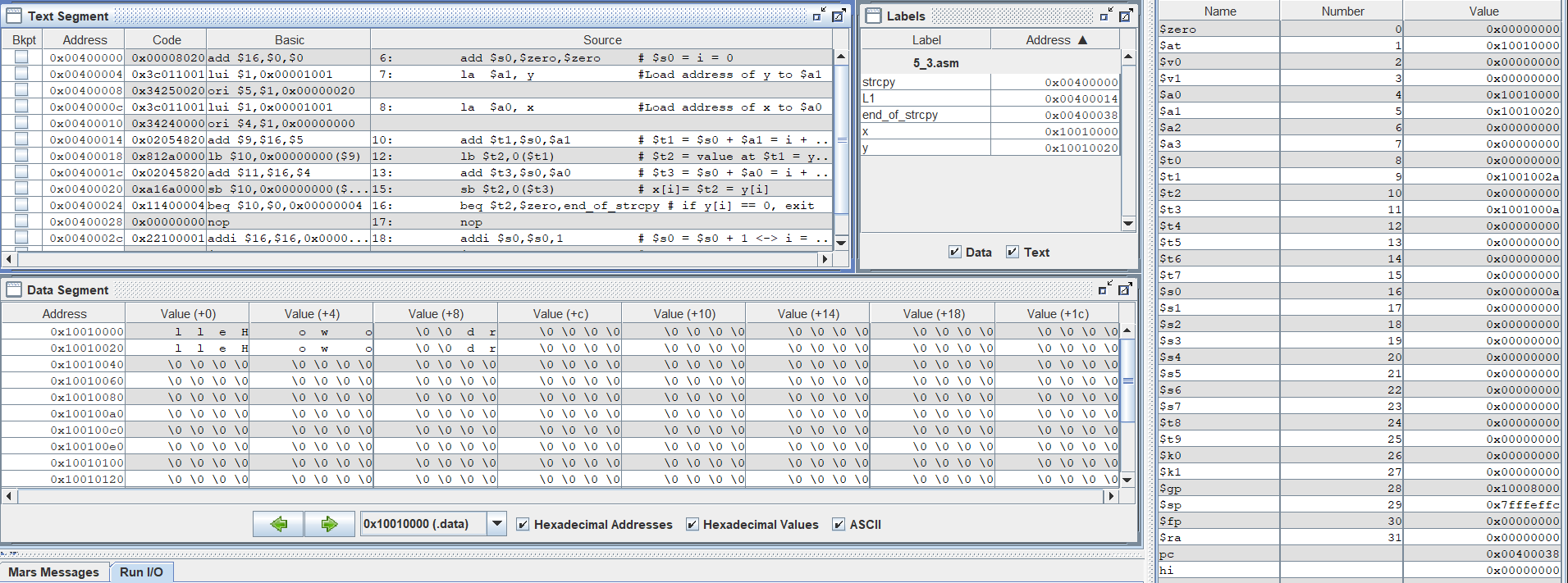
addi $s0,$s0,1 # $s0 = $s0 + 1 <-> i = i + 1

j L1 # next character

nop

end\_of\_strcpy:

Kết quả:



=>Kết quả đúng với lí thuyết

Assignment 4

Code:

.data

string: .space 50

Message1: .asciiz "Nhap xau: "

Message2: .asciiz "Do dai xau la: "

.text

main:

get\_string:

li $v0, 54

la $a0, Message1

la $a1, string

la $a2, 100

syscall

get\_length:

la $a0,string # $a0 = address(string[0])

add $t0,$zero,$zero # $t0 = i = 0

check\_char:

add $t1,$a0,$t0 # $t1 = $a0 + $t0

# = address(string[i])

lb $t2, 0($t1) # $t2 = string[i]

beq $t2, $zero, end\_of\_str # is null char?

addi $t0, $t0, 1 # $t0 = $t0 + 1 -> i = i + 1

j check\_char

end\_of\_str:

end\_of\_get\_length:

print\_length:

addi $t0, $t0, -1

li $v0, 56

la $a0, Message2

move $a1, $t0

syscall

Chạy chương trình: Nhập xâu “Duong ne”

A screenshot of a computer

Description automatically generated

Kết quả:

A screenshot of a computer

Description automatically generated

=>Xâu “Duong ne” có 8 kí tự nên kết quả đúng với lí thuyết

Assignment 5

Code:

.data

get\_char: .space 20

message1: .asciiz "Nhap ky tu thu "

message2: .asciiz ": "

message3: .asciiz "\n"

message4: .asciiz "Chuoi ky tu vua nhap bi dao nguoc la: "

.text

li $s0, 20 # N = 20

li $s1, 0 # i = 0

la $s2, get\_char # Load address of get\_char[0]

li $s3, 10 # Char \n in ASCII

read\_char:

beq $s1, $s0, end\_read\_char # i = N branch to exit

# Show message "Nhap ky tu thu i: "

li $v0, 4

la $a0, message1

syscall

addi $t1, $s1, 1

li $v0, 1

move $a0, $t1

syscall

li $v0, 4

la $a0, message2

syscall

li $v0, 12 # Read character

syscall

move $t0, $v0

beq $t0, $s3, end\_read\_char # Press "Enter" branch to exit

li $v0, 4

la $a0, message3

syscall

add $s5, $s2, $s1 #$s5 = Address of get\_char[i] = get\_char[0] + i

sb $t0, 0($s5) #Store character to get\_char[i]

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

j read\_char

end\_read\_char:

li $v0, 4

la $a0, message4

syscall

print\_string:

li $v0, 11

lb $a0, 0($s5)

syscall

beq $s5, $s2, exit

addi $s5, $s5, -1

j print\_string

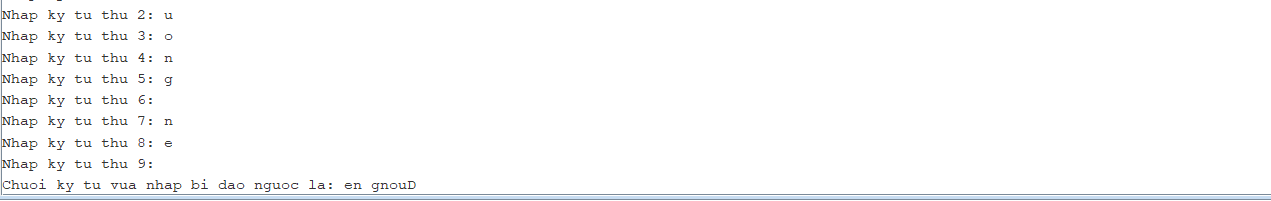
exit:

li $v0, 10

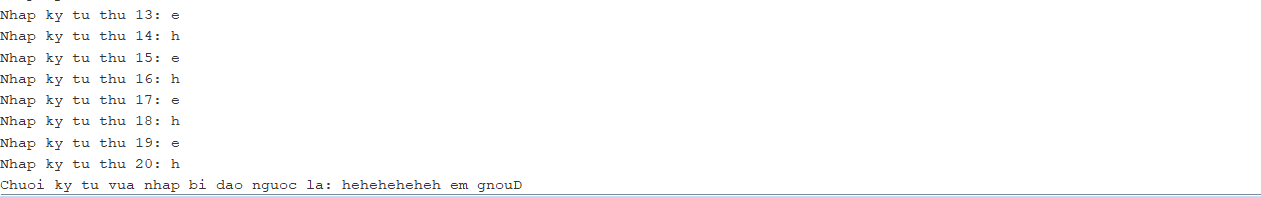
syscall

Kết quả:

Nhập xâu “Duong ne” có 8 ký tự:



Nhập xâu “Duong ne hehehehehehehe” có 23 ký tự:



=>Kết quả đúng với lí thuyết