BÀI BÁO CÁO TUẦN 4

Họ và tên: Nguyễn Trọng Khánh Duy

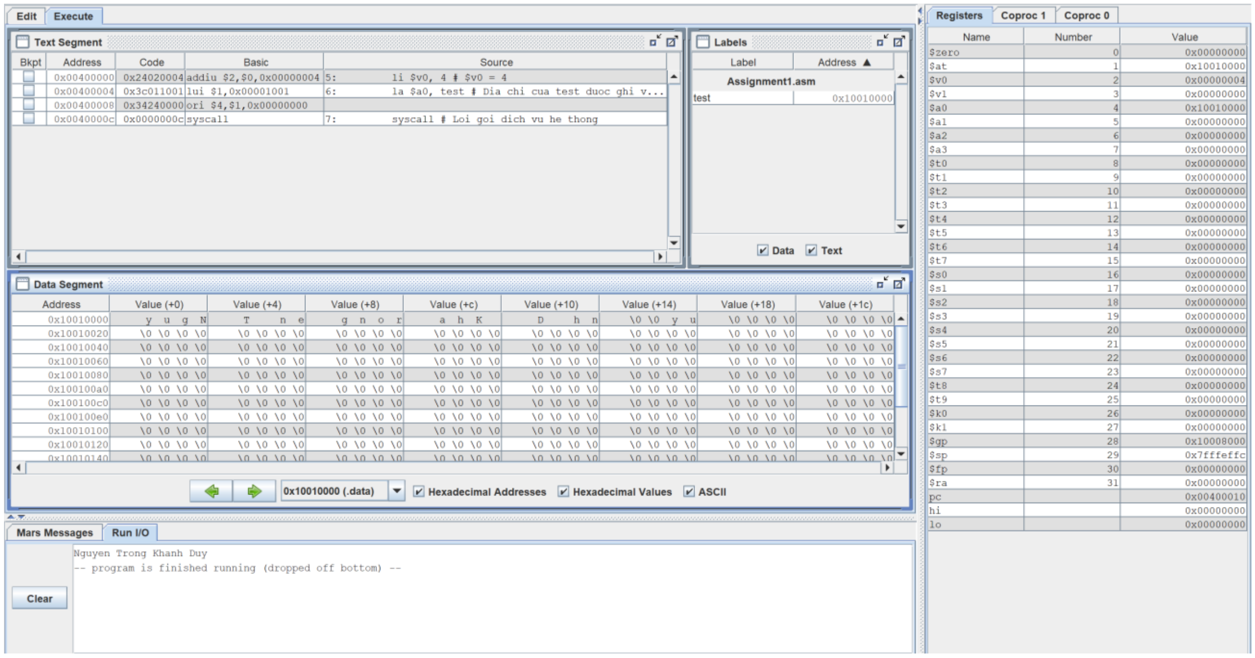
MSSV: 20210284

Assignment 1

<u>CODE:</u>

```
#Laboratory Exercise 5, Assignment 1
.data
        test: .asciiz "Nguyen Trong Khanh Duy"
.text
        li $v0, 4               # $v0 = 4
        la $a0, test            # Dia chi cua test duoc ghi vao $a0
        syscall                 # Loi goi dich vu he thong
```

<u>Kết quả:</u>



Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x24020004	addiu $2,$0,0x00000004	5: li $v0, 4 # $v0 = 4
	0x00400004	0x3c011001	lui $1,0x00001001	6: la $a0, test # Dia chi cua test duoc ghi v...
	0x00400008	0x34240000	ori $4,$1,0x00000000	
	0x0040000c	0x0000000c	syscall	7: syscall # Loi goi dich vu he thong

Labels

Label	Address ▲
Assignment1.asm	
test	0x10010000

✓ Data ✓ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	y u g N	T n e	g n o r	a h K	D h n	\0 \0 y u	\0 \0 \0 \0	\0 \0 \0 \0
0x10010020	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010040	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010060	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010080	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100a0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100c0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100e0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010100	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010120	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010140	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

0x10010000 (.data) ✓ Hexadecimal Addresses ✓ Hexadecimal Values ✓ ASCII

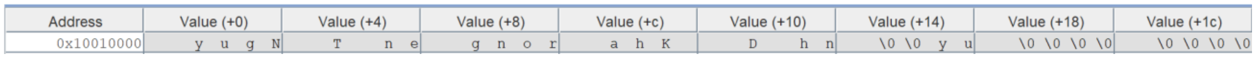
Run I/O

```
Nguyen Trong Khanh Duy
-- program is finished running (dropped off bottom) --
```

Registers

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000004
$v1	3	0x00000000
$a0	4	0x10010000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400010
hi		0x00000000
lo		0x00000000

Nhận xét: Chuỗi được lưu vào bộ nhớ với thứ tự ngược lại. Và mỗi value sẽ có 4 ký tự.



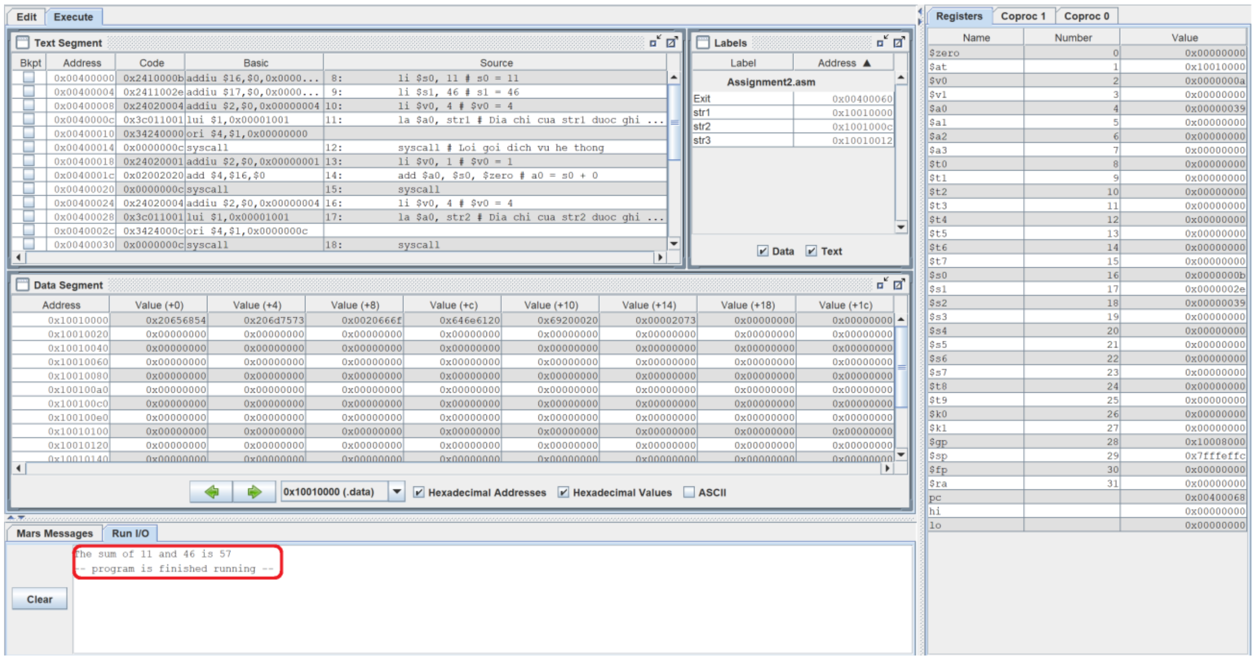
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	y u g N	T n e	g n o r	a h K	D h n	\0 \0 y u	\0 \0 \0 \0	\0 \0 \0 \0

Assignment 2

CODE:

```
#Laboratory Exercise 5, Assignment 2
#Gia su s1 = 11, s2 = 46 => sum = 57
.data
        str1: .asciiz "The sum of "
        str2: .asciiz " and "
        str3: .asciiz " is "
.text
        li $s0, 11                  # s0 = 11
        li $s1, 46                  # s1 = 46
        li $v0, 4                   # $v0 = 4
        la $a0, str1                # Dia chi cua str1 duoc ghi vao $a0
        syscall
        li $v0, 1                   # $v0 = 1
        add $a0, $s0, $zero         # a0 = s0 + 0
        syscall
        li $v0, 4                   # $v0 = 4
        la $a0, str2                # Dia chi cua str2 duoc ghi vao $a0
        syscall
        li $v0, 1                    # $v0 = 1
        add $a0, $s1, $zero          # a0 = s1 + 0
        syscall
        li $v0, 4                    # $v0 = 4
        la $a0, str3                 # Dia chi cua str3 duoc ghi vao $a0
        syscall
        add $s2, $s1, $s0            # sum = s2 = s1 + s0
        li $v0, 1                    # $v0 = 1
        add $a0, $s2, $zero          # a0 = s2 + 0
        syscall
Exit:
        li $v0, 10                   # $v0 = 10
        syscall
```

<u>Kết quả:</u>



Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x2410000b	addiu $16,$0,0x0000...	8: li $s0, 11 # s0 = 11
☐	0x00400004	0x2411002e	addiu $17,$0,0x0000...	9: li $s1, 46 # s1 = 46
☐	0x00400008	0x24020004	addiu $2,$0,0x00000004	10: li $v0, 4 # $v0 = 4
☐	0x0040000c	0x3c011001	lui $1,0x00001001	11: la $a0, str1 # Dia chi cua str1 duoc ghi ...
☐	0x00400010	0x34240000	ori $4,$1,0x00000000	
☐	0x00400014	0x0000000c	syscall	12: syscall # Loi goi dich vu he thong
☐	0x00400018	0x24020001	addiu $2,$0,0x00000001	13: li $v0, 1 # $v0 = 1
☐	0x0040001c	0x02002020	add $4,$16,$0	14: add $a0, $s0, $zero # a0 = s0 + 0
☐	0x00400020	0x0000000c	syscall	15: syscall
☐	0x00400024	0x24020004	addiu $2,$0,0x00000004	16: li $v0, 4 # $v0 = 4
☐	0x00400028	0x3c011001	lui $1,0x00001001	17: la $a0, str2 # Dia chi cua str2 duoc ghi ...
☐	0x0040002c	0x3424000c	ori $4,$1,0x0000000c	
☐	0x00400030	0x0000000c	syscall	18: syscall

Labels

Label	Address ▲
Assignment2.asm	
Exit	0x00400060
str1	0x10010000
str2	0x1001000c
str3	0x10010012

☑ Data ☑ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x20656854	0x206d7573	0x0020666f	0x646e6120	0x69200020	0x00002073	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values ☐ ASCII

Run I/O

```
The sum of 11 and 46 is 57
-- program is finished running --
```

Registers

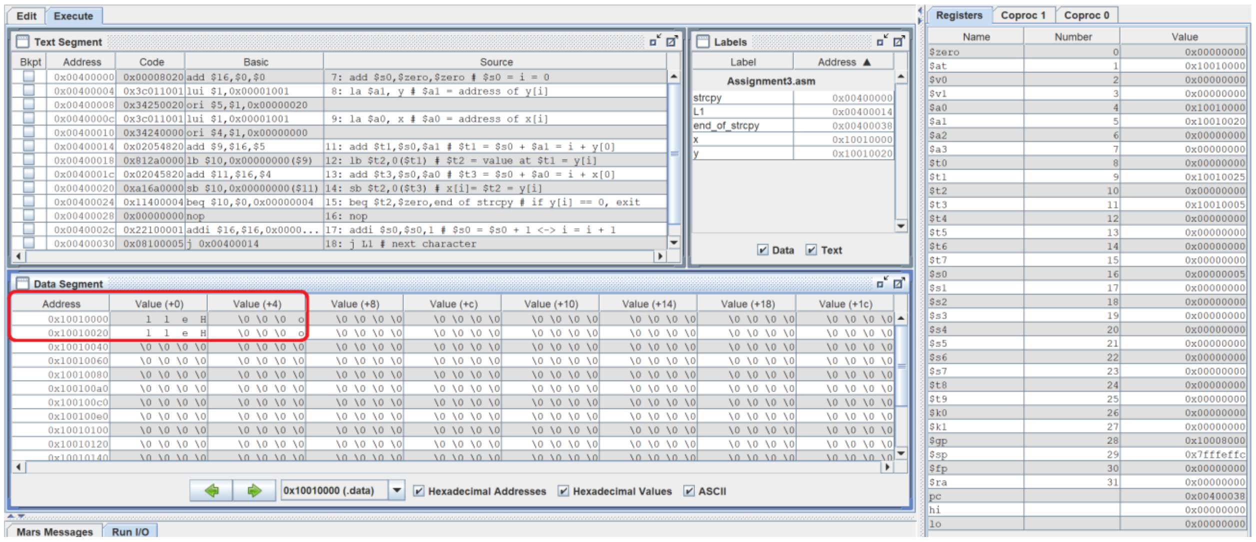
Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x0000000a
$v1	3	0x00000000
$a0	4	0x00000039
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x0000000b
$s1	17	0x0000002e
$s2	18	0x00000039
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400068
hi		0x00000000
lo		0x00000000

Assignment 3

<u>CODE:</u>

```
#Laboratory Exercise 5, Assignment 3
.data
        x: .space 32                    # destination string x, empty
        y: .asciiz "Hello"              # source string y
.text
        strcpy:
        add $s0, $zero, $zero           # $s0 = i = 0
        la $a1, y                       # $a1 = address of y[i]
        la $a0, x                       # $a0 = address of x[i]
L1:
        add $t1, $s0, $a1                  # $t1 = $s0 + $a1 = i + y[0]

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

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

        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:
```

<u>Kết quả:</u>



Text Segment

Bkpt	Address	Code	Basic	Source
☐	0x00400000	0x00008020	add $16,$0,$0	7: add $s0,$zero,$zero # $s0 = i = 0
☐	0x00400004	0x3c011001	lui $1,0x00001001	8: la $a1, y # $a1 = address of y[i]
☐	0x00400008	0x34250020	ori $5,$1,0x00000020	
☐	0x0040000c	0x3c011001	lui $1,0x00001001	9: la $a0, x # $a0 = address of x[i]
☐	0x00400010	0x34240000	ori $4,$1,0x00000000	
☐	0x00400014	0x02054820	add $9,$16,$5	11: add $t1,$s0,$a1 # $t1 = $s0 + $a1 = i + y[0]
☐	0x00400018	0x812a0000	lb $10,0x00000000($9)	12: lb $t2,0($t1) # $t2 = value at $t1 = y[i]
☐	0x0040001c	0x02045820	add $11,$16,$4	13: add $t3,$s0,$a0 # $t3 = $s0 + $a0 = i + x[0]
☐	0x00400020	0xa16a0000	sb $10,0x00000000($11)	14: sb $t2,0($t3) # x[i]= $t2 = y[i]
☐	0x00400024	0x11400004	beq $10,$0,0x00000004	15: beq $t2,$zero,end of strcpy # if y[i] == 0, exit
☐	0x00400028	0x00000000	nop	16: nop
☐	0x0040002c	0x22100001	addi $16,$16,0x0000...	17: addi $s0,$s0,1 # $s0 = $s0 + 1 <-> i = i + 1
☐	0x00400030	0x08100005	j 0x00400014	18: j L1 # next character

Labels

Label	Address ▲
Assignment3.asm	
strcpy	0x00400000
L1	0x00400014
end_of_strcpy	0x00400038
x	0x10010000
y	0x10010020

✓ Data ✓ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	l l e H	\0 \0 \0 o	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010020	l l e H	\0 \0 \0 o	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010040	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010060	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010080	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100a0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100c0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100e0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010100	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010120	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010140	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

0x10010000 (.data) ✓ Hexadecimal Addresses ✓ Hexadecimal Values ✓ ASCII

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x10010000
$a1	5	0x10010020
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x10010025
$t2	10	0x00000000
$t3	11	0x10010005
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000005
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400038
hi		0x00000000
lo		0x00000000

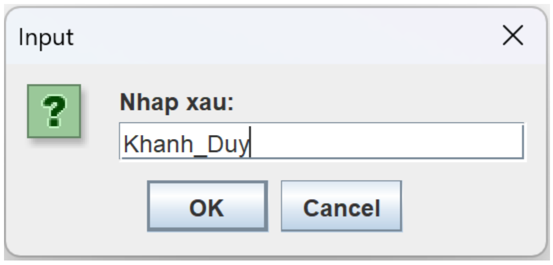
Assignment 4

CODE:

```
#Laboratory Exercise 5, Assignment 4
.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, 50
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:
li $v0, 56
la $a0, Message2
addi $a1, $t0, -1                        #Do ký tự cuối là null nên ta phải - 1
syscall
```

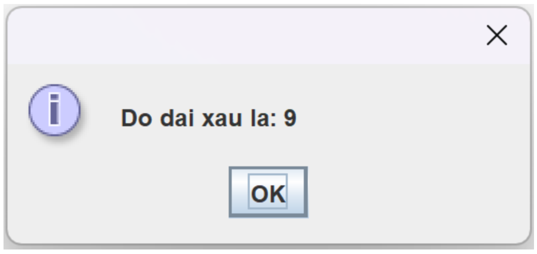
Kết quả:



Input

Nhap xau:

Khanh_Duy



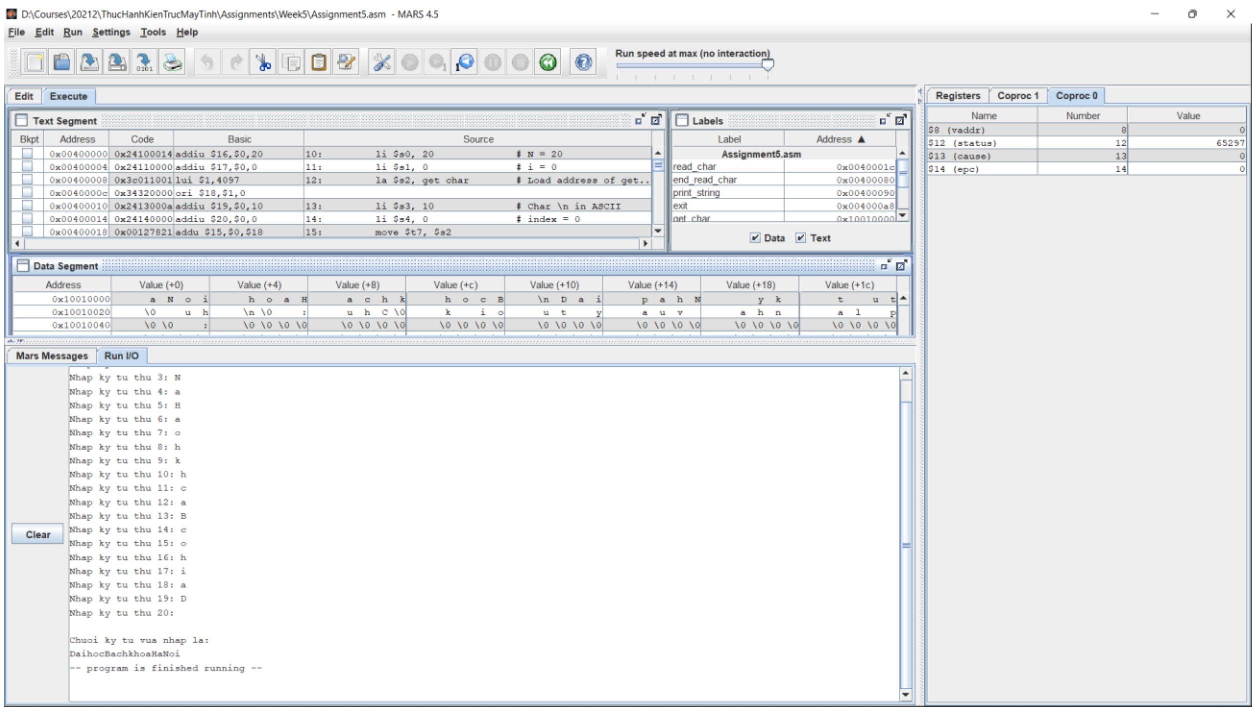
Do dai xau la: 9

Assignment 5

CODE:

```
#Laboratory Exercise 5, Assignment 5
.data
        get_char: .space 20
        message1: .asciiz "Nhap ky tu thu "
        message2: .asciiz ": "
        message3: .asciiz "\n"
        message4: .asciiz "Chuoi ky tu vua nhap 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                    # Show message4
        la $a0, message4
        syscall
print_string:
        li $v0, 11                  # Show ky tu tai dia chi trong $s5
        lb $a0, 0($s5)
        syscall

        beq $s5, $s2, exit # $s5 = address cua ky tu cuoi cung
        addi $s5, $s5, -1   # Tien dan den ky tu dau tien
        j print_string
exit:
        li $v0, 10
        syscall
```

<u>Kết quả:</u>


D:\Courses\20212\ThucHanhKienTrucMayTinh\Assignments\Week5\Assignment5.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit
Execute
Text Segment
Labels
Data Segment
Mars Messages
Run I/O
Clear
Registers
Coproc 1
Coproc 0

Bkpt	Address	Code	Basic	Source		
	0x00400000	0x24100014	addiu $16,$0,20	10:	li $s0, 20	# N = 20
	0x00400004	0x24110000	addiu $17,$0,0	11:	li $s1, 0	# i = 0
	0x00400008	0x3c011001	lui $1,4097	12:	la $s2, get char	# Load address of get..
	0x0040000c	0x34320000	ori $18,$1,0			
	0x00400010	0x2413000a	addiu $19,$0,10	13:	li $s3, 10	# Char \n in ASCII
	0x00400014	0x24140000	addiu $20,$0,0	14:	li $s4, 0	# index = 0
	0x00400018	0x00127821	addu $15,$0,$18	15:	move $t7, $s2	

Label	Address ▲
Assignment5.asm	
read_char	0x0040001c
end_read_char	0x00400080
print_string	0x00400090
exit	0x004000a8
get_char	0x10010000

☑ Data ☑ Text

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	a N o i	h o a H	a c h k	h o c B	\n D a i	p a h N	y k	t u t
0x10010020	\0 u h	\n \0 :	u h C \0	k i o	u t y	a u v	a h n	a l p
0x10010040	\0 \0 :	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

```
Nhap ky tu thu 3: N
Nhap ky tu thu 4: a
Nhap ky tu thu 5: H
Nhap ky tu thu 6: a
Nhap ky tu thu 7: o
Nhap ky tu thu 8: h
Nhap ky tu thu 9: k
Nhap ky tu thu 10: h
Nhap ky tu thu 11: c
Nhap ky tu thu 12: a
Nhap ky tu thu 13: B
Nhap ky tu thu 14: c
Nhap ky tu thu 15: o
Nhap ky tu thu 16: h
Nhap ky tu thu 17: i
Nhap ky tu thu 18: a
Nhap ky tu thu 19: D
Nhap ky tu thu 20:

Chuoi ky tu vua nhap la:
DaihocBachkhoaHaNoi
-- program is finished running --
```

Name	Number	Value
$8 (vaddr)	8	0
$12 (status)	12	65297
$13 (cause)	13	0
$14 (epc)	14	0