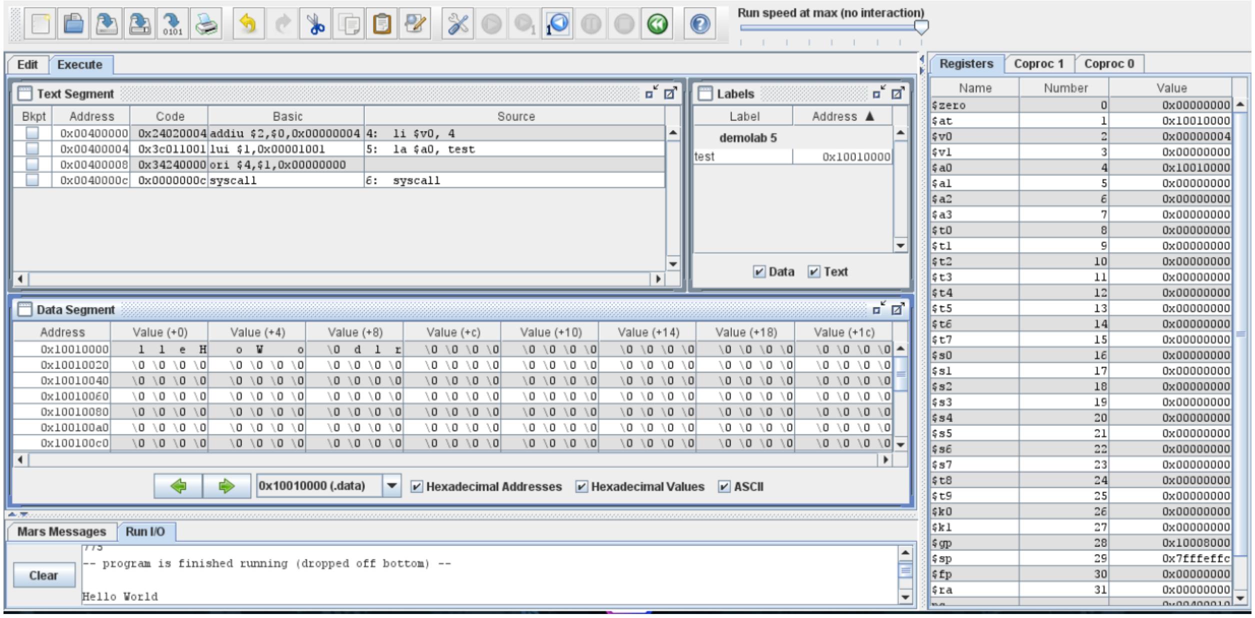
Phạm Duy Hưng – 20225850

Assignment 1:

```
.data
test: .asciiz "Hello World"
.text
 li $v0, 4
 la $a0, test
 syscall
```



Bkpt	Address	Code	Basic	Source
□	0x00400000	0x24020004	addiu $2,$0,0x00000004	4: li $v0, 4
□	0x00400004	0x3c011001	lui $1,0x00001001	5: la $a0, test
□	0x00400008	0x34240000	ori $4,$1,0x00000000	
□	0x0040000c	0x0000000c	syscall	6: syscall

Label	Address ▲
demolab 5	
test	0x10010000

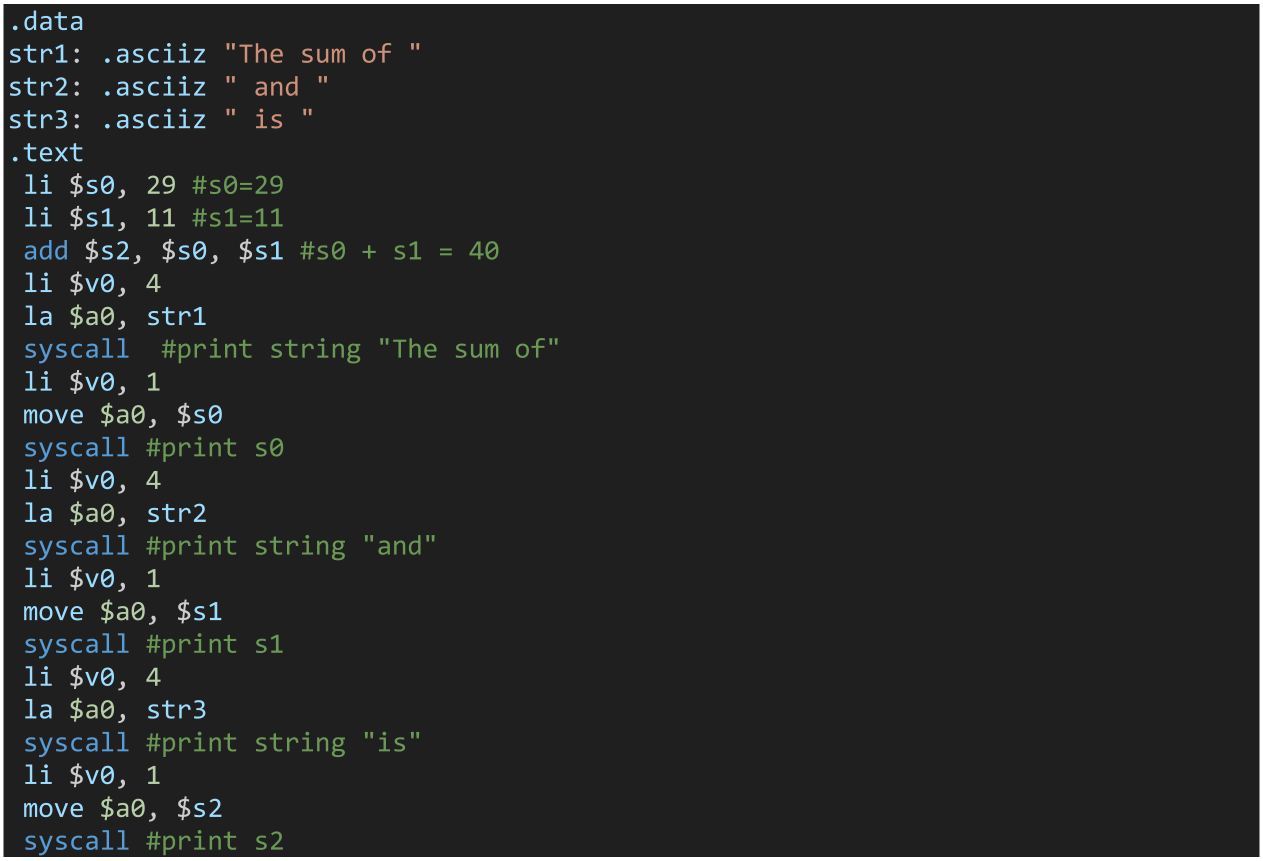
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	l l e H	o W o	\0 d l r	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010020	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
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

0x10010000 (.data)

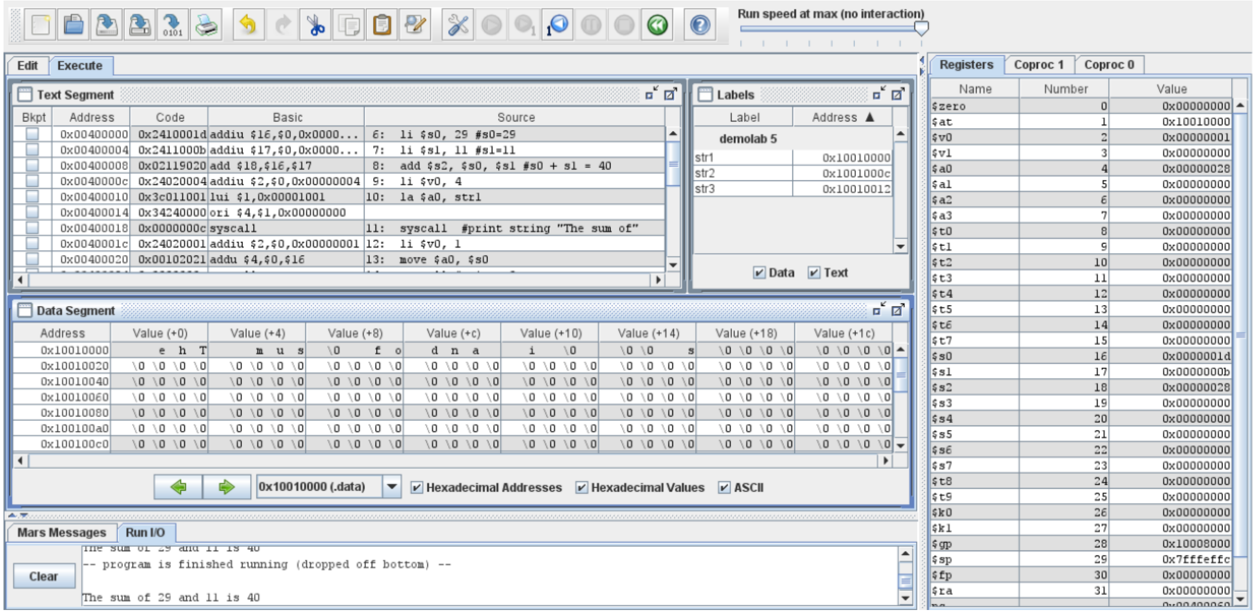
```
-- program is finished running (dropped off bottom) --

Hello World
```

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

Assignment 2:



```
.data
str1: .asciiz "The sum of "
str2: .asciiz " and "
str3: .asciiz " is "
.text
 li $s0, 29 #s0=29
 li $s1, 11 #s1=11
 add $s2, $s0, $s1 #s0 + s1 = 40
 li $v0, 4
 la $a0, str1
 syscall  #print string "The sum of"
 li $v0, 1
 move $a0, $s0
 syscall #print s0
 li $v0, 4
 la $a0, str2
 syscall #print string "and"
 li $v0, 1
 move $a0, $s1
 syscall #print s1
 li $v0, 4
 la $a0, str3
 syscall #print string "is"
 li $v0, 1
 move $a0, $s2
 syscall #print s2
```



Text Segment

Bkpt	Address	Code	Basic	Source
□	0x00400000	0x2410001d	addiu $16,$0,0x0000...	6: li $s0, 29 #s0=29
□	0x00400004	0x2411000b	addiu $17,$0,0x0000...	7: li $s1, 11 #s1=11
□	0x00400008	0x02119020	add $18,$16,$17	8: add $s2, $s0, $s1 #s0 + s1 = 40
□	0x0040000c	0x24020004	addiu $2,$0,0x00000004	9: li $v0, 4
□	0x00400010	0x3c011001	lui $1,0x00001001	10: la $a0, str1
□	0x00400014	0x34240000	ori $4,$1,0x00000000	
□	0x00400018	0x0000000c	syscall	11: syscall #print string "The sum of"
□	0x0040001c	0x24020001	addiu $2,$0,0x00000001	12: li $v0, 1
□	0x00400020	0x00102021	addu $4,$0,$16	13: move $a0, $s0

Labels

Label	Address ▲
demolab 5	
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	e h T	m u s	\0 f o	d n a	i \0	\0 \0 s	\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

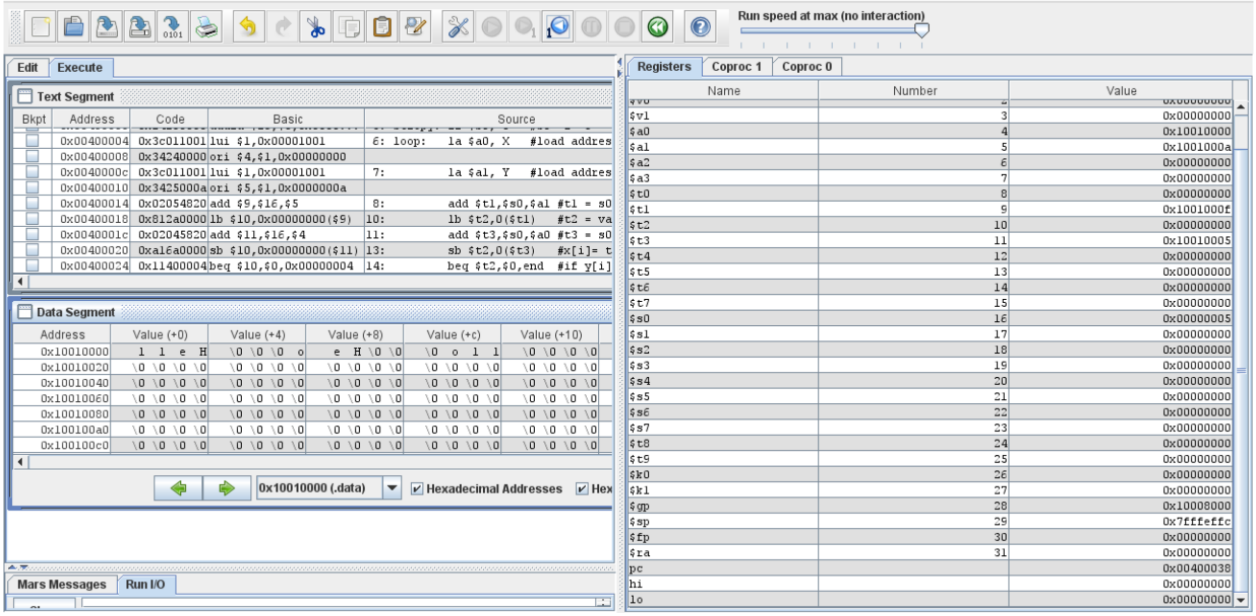
0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hexadecimal Values ☑ ASCII

```
The sum of 29 and 11 is 40
-- program is finished running (dropped off bottom) --

The sum of 29 and 11 is 40
```

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000001
$v1	3	0x00000000
$a0	4	0x00000028
$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	0x0000001d
$s1	17	0x0000000b
$s2	18	0x00000028
$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

Assignment 3:

```
.data
X: .space 10      # destination string x, empty
Y: .asciiz "Hello"  # source string y
.text
strcpy: li $s0, 0   #s0 =i =0
loop:   la $a0, X   #load address of string X into a0
    la $a1, Y   #load address of string Y into a1
    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,$0,end  #if y[i]==0, exit
    nop
    addi $s0,$s0,1  #s0=s0 + 1 <-> i=i+1
    j loop          #next character
    nop
end:
```



Text Segment

Bkpt	Address	Code	Basic	Source
□	0x00400004	0x3c011001	lui $1,0x00001001	6: loop: la $a0, X #load addres
□	0x00400008	0x34240000	ori $4,$1,0x00000000	
□	0x0040000c	0x3c011001	lui $1,0x00001001	7: la $a1, Y #load addres
□	0x00400010	0x3425000a	ori $5,$1,0x0000000a	
□	0x00400014	0x02054820	add $9,$16,$5	8: add $t1,$s0,$a1 #t1 = s0
□	0x00400018	0x812a0000	lb $10,0x00000000($9)	10: lb $t2,0($t1) #t2 = va
□	0x0040001c	0x02045820	add $11,$16,$4	11: add $t3,$s0,$a0 #t3 = s0
□	0x00400020	0xa16a0000	sb $10,0x00000000($11)	13: sb $t2,0($t3) #x[i]= t
□	0x00400024	0x11400004	beq $10,$0,0x00000004	14: beq $t2,$0,end #if y[i]

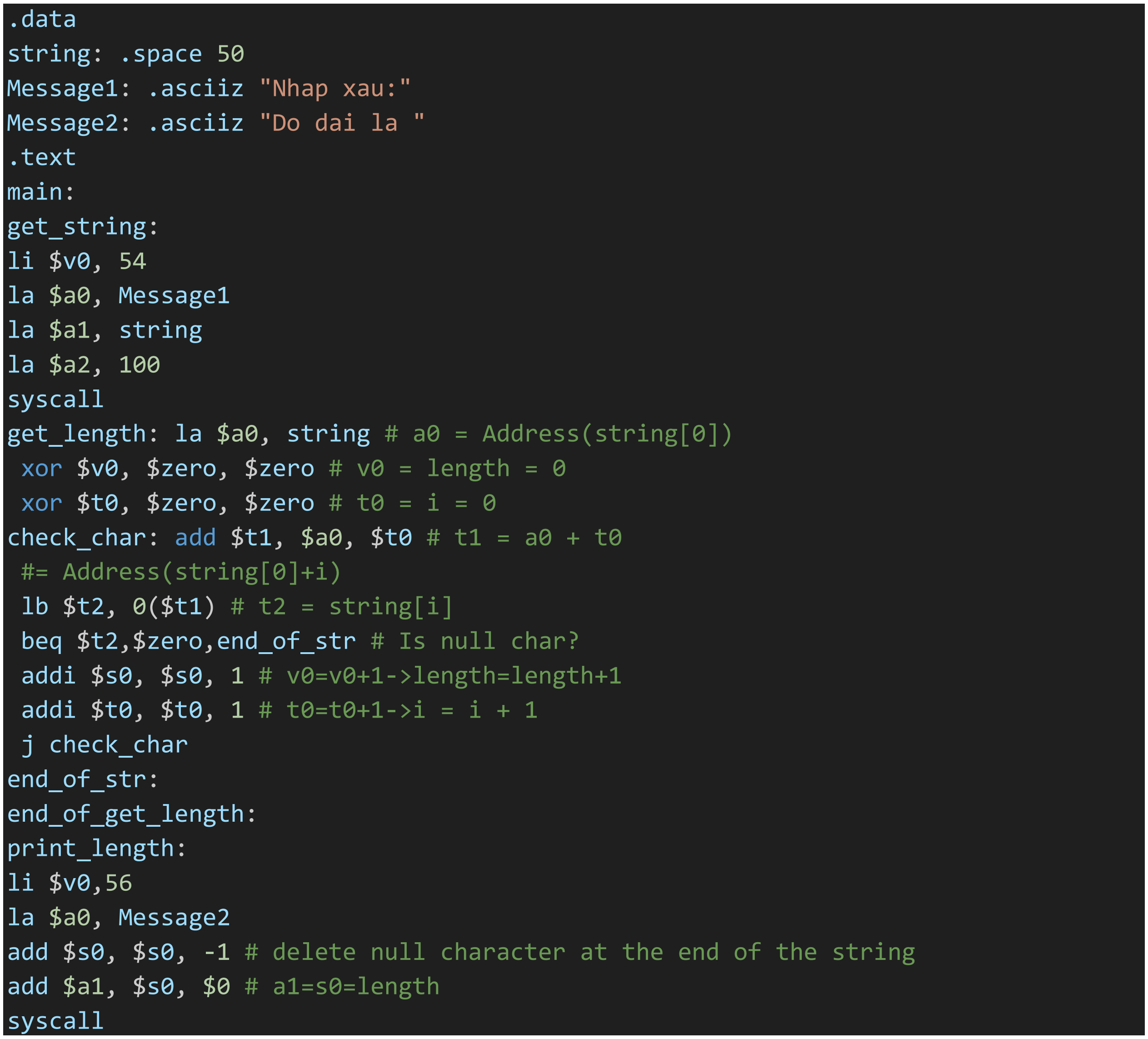
Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)
0x10010000	l l e H	\0 \0 \0 o	e H \0 \0	\0 o l l	\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
0x10010040	\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
0x10010080	\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
0x100100c0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hex

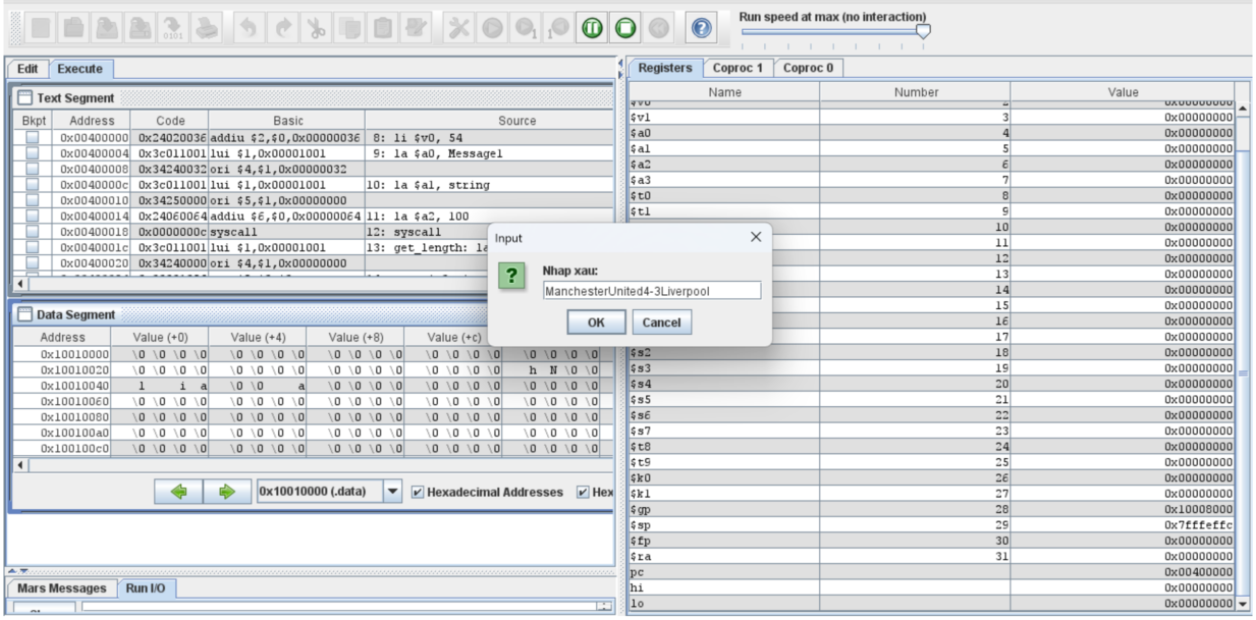
Name	Number	Value
$v1	3	0x00000000
$a0	4	0x10010000
$a1	5	0x1001000a
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x1001000f
$t2	10	0x00000000
$t3	11	0x10010005
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000005
$s1	17	0x00000000
$s2	18	0x00000000
$s3	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:



```
.data
string: .space 50
Message1: .asciiz "Nhap xau:"
Message2: .asciiz "Do dai 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])
 xor $v0, $zero, $zero # v0 = length = 0
 xor $t0, $zero, $zero # t0 = i = 0
check_char: add $t1, $a0, $t0 # t1 = a0 + t0
 #= Address(string[0]+i)
 lb $t2, 0($t1) # t2 = string[i]
 beq $t2,$zero,end_of_str # Is null char?
 addi $s0, $s0, 1 # v0=v0+1->length=length+1
 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
add $s0, $s0, -1 # delete null character at the end of the string
add $a1, $s0, $0 # a1=s0=length
syscall
```

-Nhập xâu “ManchesterUnited4-3Liverpool”



Text Segment

Bkpt	Address	Code	Basic	Source
□	0x00400000	0x24020036	addiu $2,$0,0x00000036	8: li $v0, 54
□	0x00400004	0x3c011001	lui $1,0x00001001	9: la $a0, Message1
□	0x00400008	0x34240032	ori $4,$1,0x00000032	
□	0x0040000c	0x3c011001	lui $1,0x00001001	10: la $a1, string
□	0x00400010	0x34250000	ori $5,$1,0x00000000	
□	0x00400014	0x24060064	addiu $6,$0,0x00000064	11: la $a2, 100
□	0x00400018	0x0000000c	syscall	12: syscall
□	0x0040001c	0x3c011001	lui $1,0x00001001	13: get_length: la
□	0x00400020	0x34240000	ori $4,$1,0x00000000	

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	
0x10010000	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\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	h N \0 \0
0x10010040	l i a	\0 \0 a	\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
0x10010080	\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
0x100100c0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hex

Input

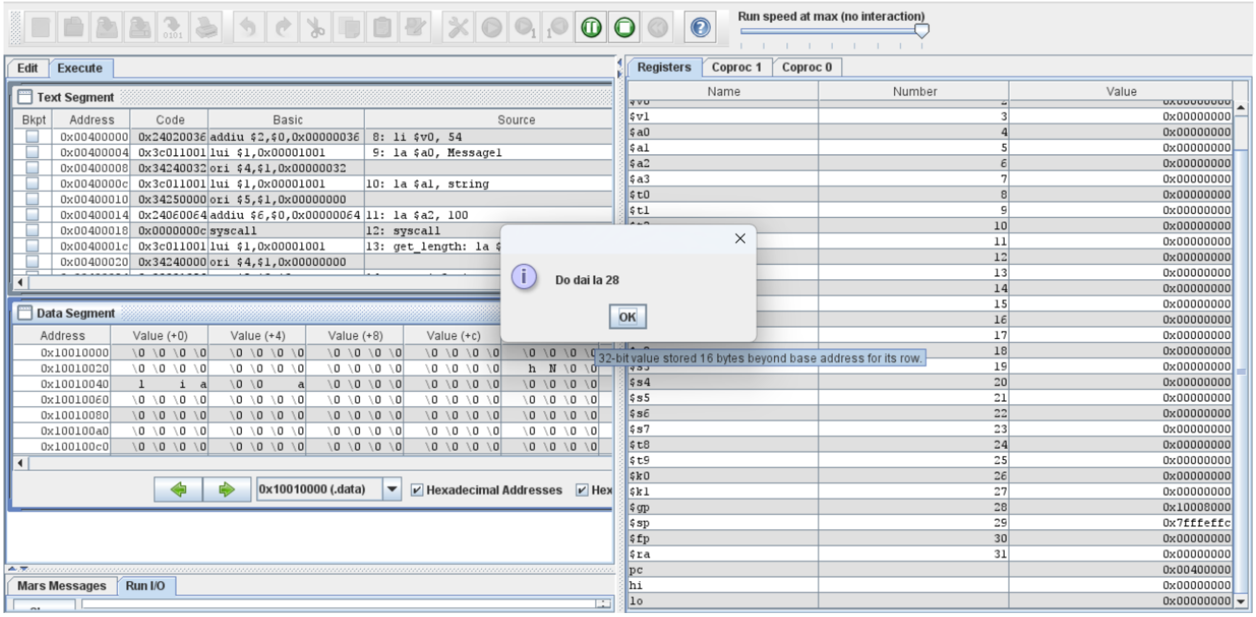
Nhap xau:

ManchesterUnited4-3Liverpool

OK Cancel

Name	Number	Value
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
	10	0x00000000
	11	0x00000000
	12	0x00000000
	13	0x00000000
	14	0x00000000
	15	0x00000000
	16	0x00000000
	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		0x00400000
hi		0x00000000
lo		0x00000000

-Trả về độ dài ký tự xâu



Text Segment

Bkpt	Address	Code	Basic	Source
□	0x00400000	0x24020036	addiu $2,$0,0x00000036	8: li $v0, 54
□	0x00400004	0x3c011001	lui $1,0x00001001	9: la $a0, Message1
□	0x00400008	0x34240032	ori $4,$1,0x00000032	
□	0x0040000c	0x3c011001	lui $1,0x00001001	10: la $a1, string
□	0x00400010	0x34250000	ori $5,$1,0x00000000	
□	0x00400014	0x24060064	addiu $6,$0,0x00000064	11: la $a2, 100
□	0x00400018	0x0000000c	syscall	12: syscall
□	0x0040001c	0x3c011001	lui $1,0x00001001	13: get_length: la $
□	0x00400020	0x34240000	ori $4,$1,0x00000000	

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	
0x10010000	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\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	h N \0 \0
0x10010040	l i a	\0 \0 a	\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
0x10010080	\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
0x100100c0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

0x10010000 (.data) ☑ Hexadecimal Addresses ☑ Hex

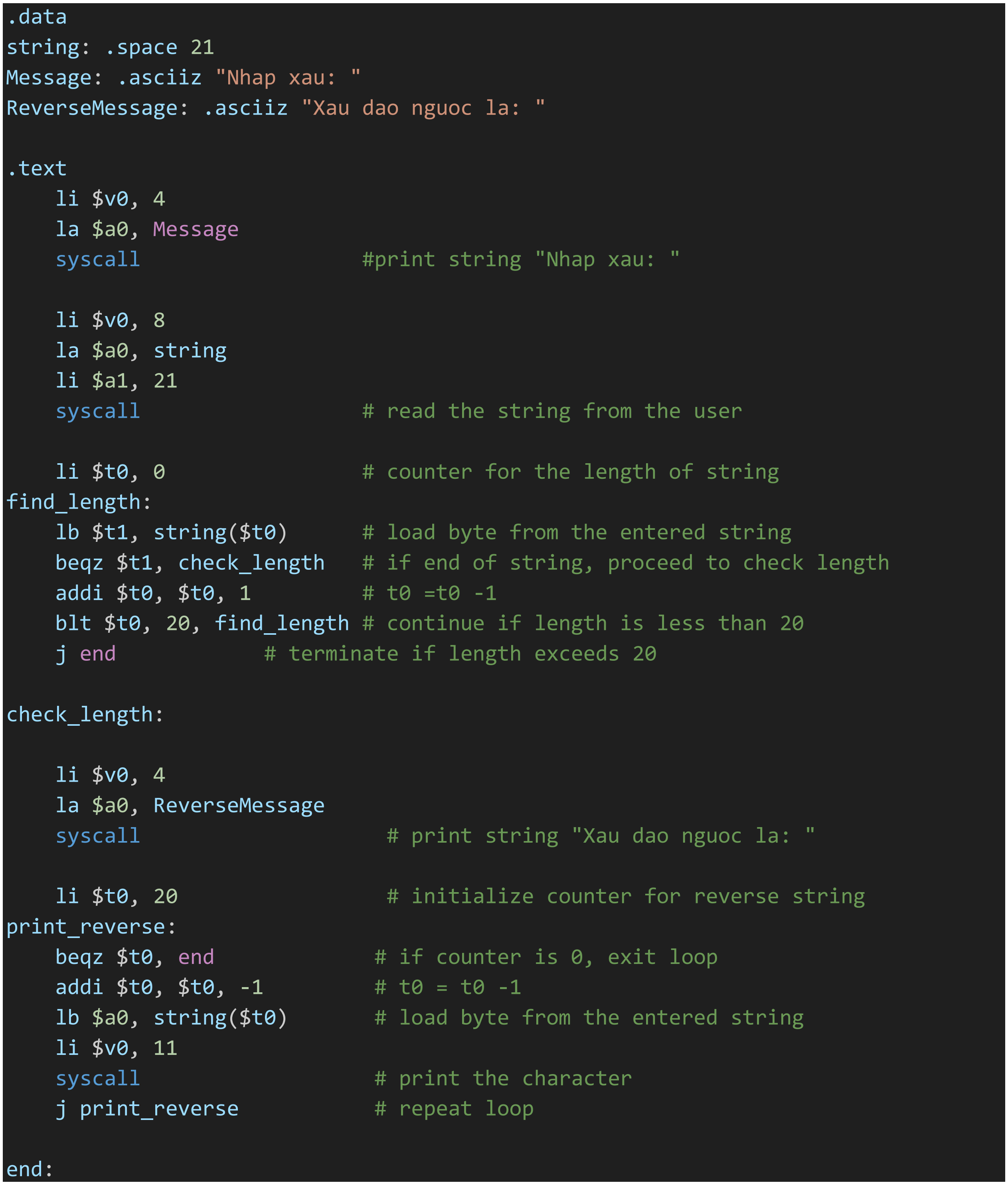
Do dai la 28

OK

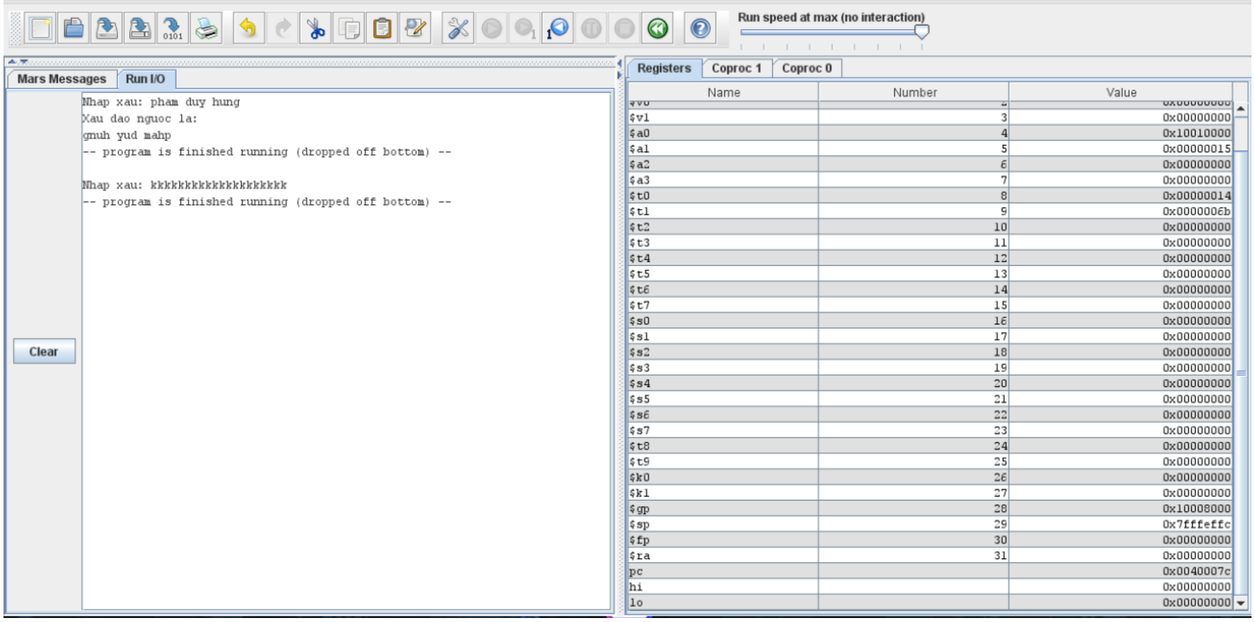
32-bit value stored 16 bytes beyond base address for its row.

Name	Number	Value
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x00000000
	10	0x00000000
	11	0x00000000
	12	0x00000000
	13	0x00000000
	14	0x00000000
	15	0x00000000
	16	0x00000000
	17	0x00000000
	18	0x00000000
	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		0x00400000
hi		0x00000000
lo		0x00000000

Assignment 5:



```
.data
string: .space 21
Message: .asciiz "Nhap xau: "
ReverseMessage: .asciiz "Xau dao nguoc la: "

.text
    li $v0, 4
    la $a0, Message
    syscall                  #print string "Nhap xau: "

    li $v0, 8
    la $a0, string
    li $a1, 21
    syscall                  # read the string from the user

    li $t0, 0                # counter for the length of string
find_length:
    lb $t1, string($t0)      # load byte from the entered string
    beqz $t1, check_length   # if end of string, proceed to check length
    addi $t0, $t0, 1         # t0 =t0 -1
    blt $t0, 20, find_length # continue if length is less than 20
    j end            # terminate if length exceeds 20

check_length:

    li $v0, 4
    la $a0, ReverseMessage
    syscall                    # print string "Xau dao nguoc la: "

    li $t0, 20                 # initialize counter for reverse string
print_reverse:
    beqz $t0, end             # if counter is 0, exit loop
    addi $t0, $t0, -1         # t0 = t0 -1
    lb $a0, string($t0)       # load byte from the entered string
    li $v0, 11
    syscall                   # print the character
    j print_reverse           # repeat loop

end:
```



```
Nhap xau: pham duy hung
Xau dao nguoc la:
gnuh yud mahp
-- program is finished running (dropped off bottom) --

Nhap xau: kkkkkkkkkkkkkkkkkkkk
-- program is finished running (dropped off bottom) --
```

Name	Number	Value
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x10010000
$a1	5	0x00000015
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000014
$t1	9	0x0000006b
$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		0x0040007c
hi		0x00000000
lo		0x00000000