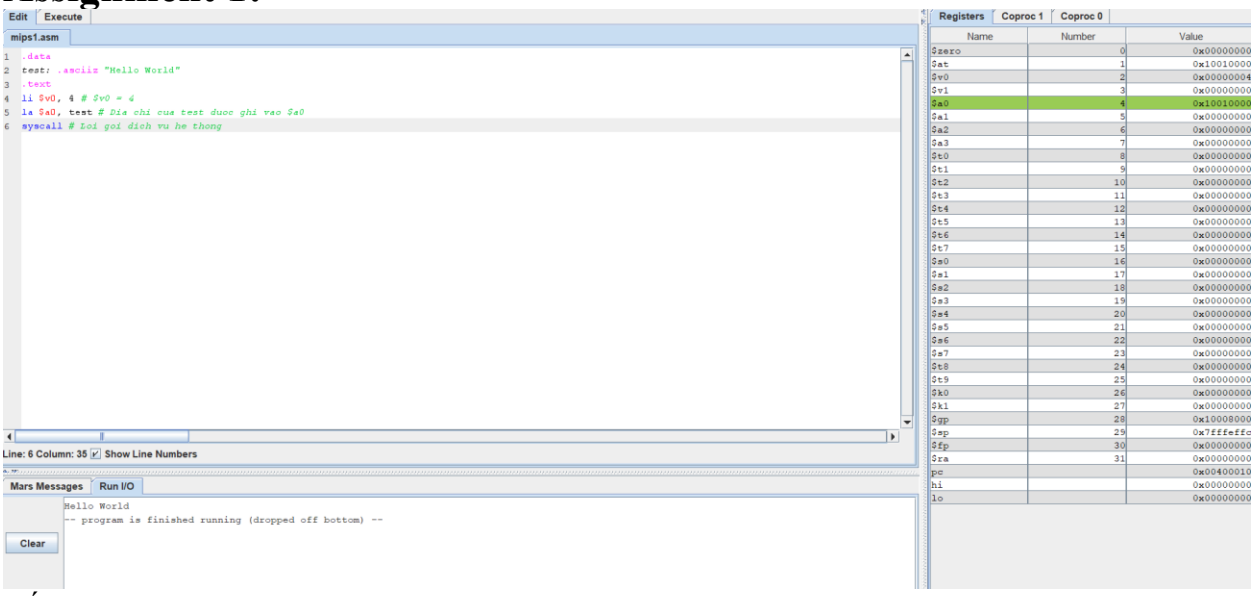
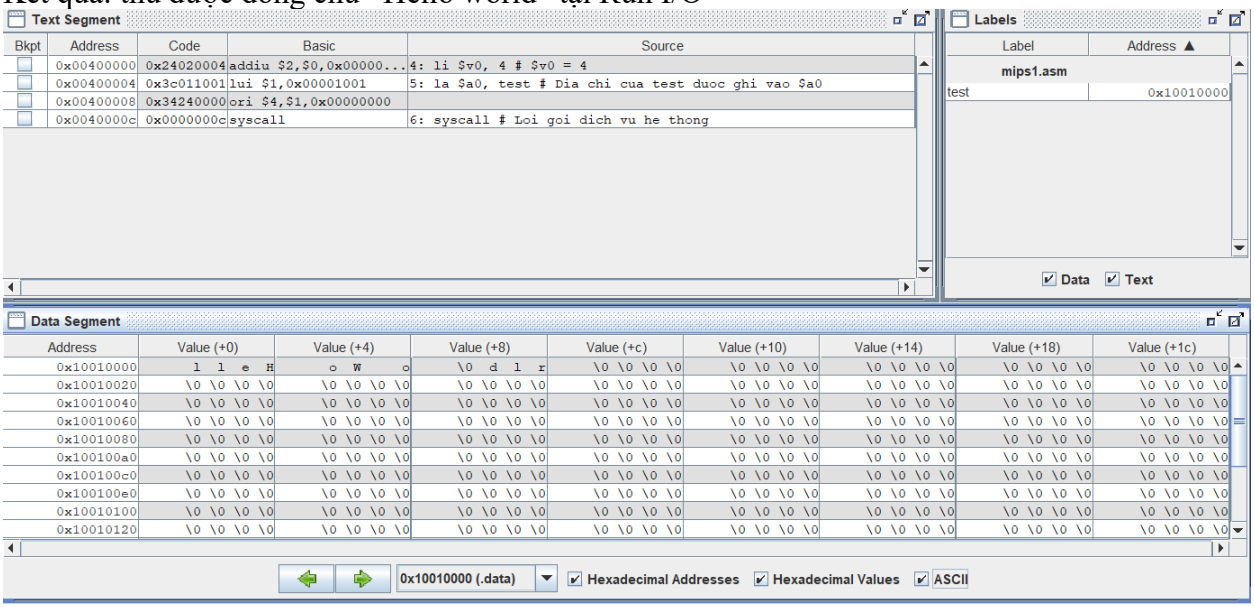
Họ và Tên: Hà Trung Chiến
MSSV: 20225794

LAB 05

Assignment 1:



Kết quả: thu được dòng chữ “Hello world” tại Run I/O



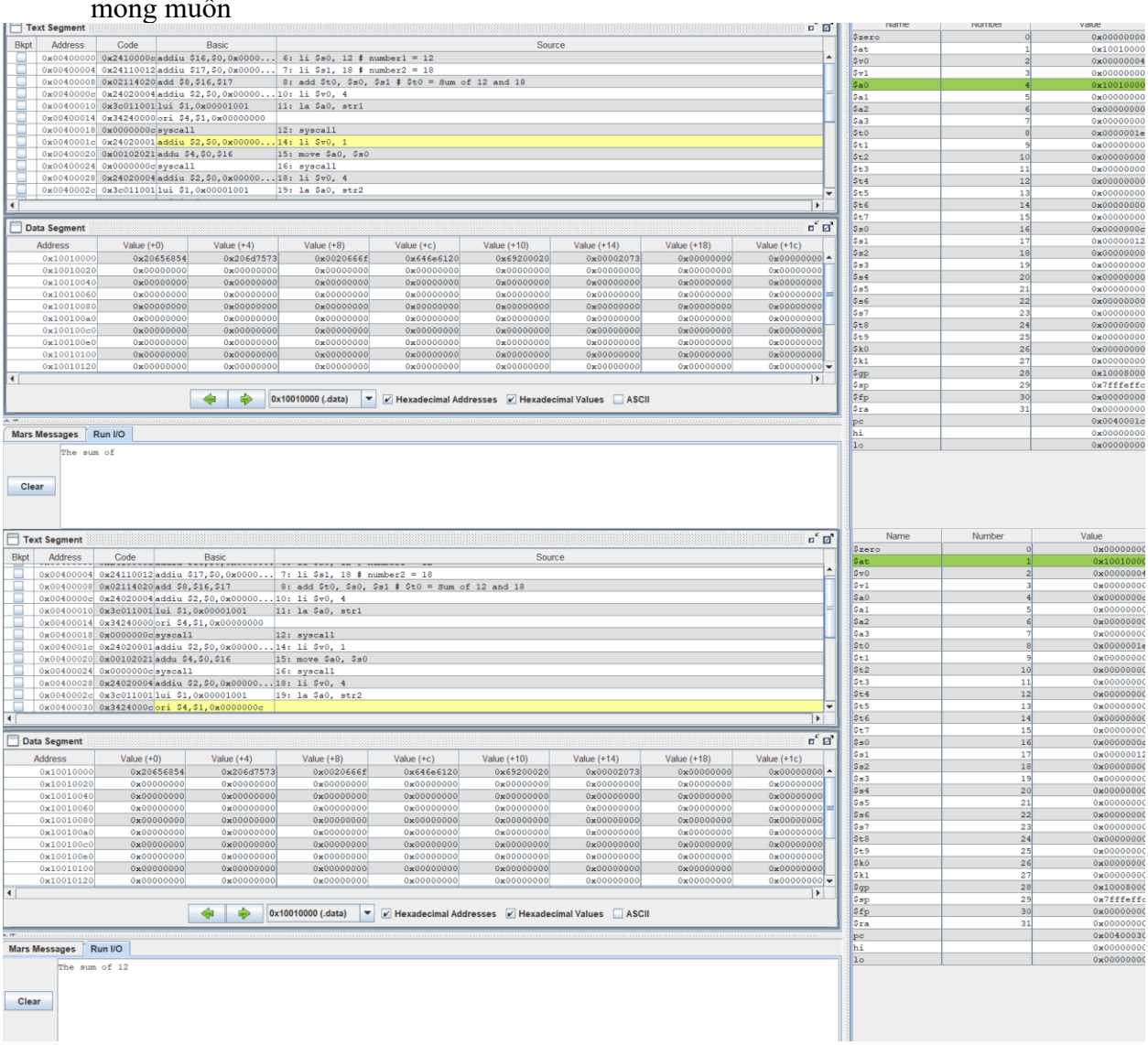
Giải thích:

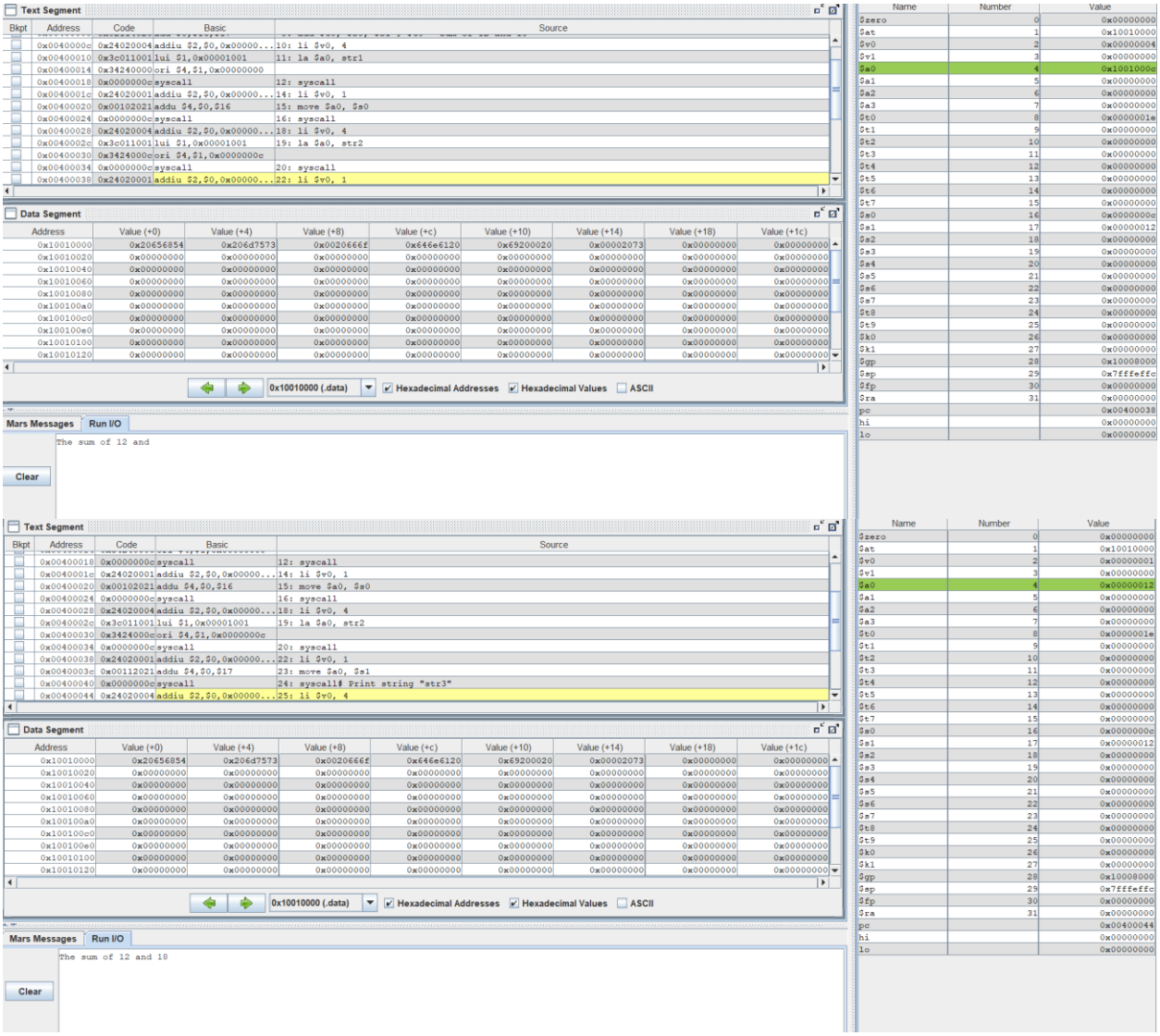
- Chọn giá trị 4 cho thanh ghi $v0 để thực hiện chức năng in ra xâu “Hello world”.
- Địa chỉ của test là 0x1001000 lưu các kí tự của xâu sau đó được lưu vào thanh ghi $a0 bằng câu lệnh “la $a0,test”.

Assignment 2:

```
.data
str1: .asciiz "The sum of "
str2: .asciiz " and "
str3: .asciiz " is "
.text
li $s0, 12 # number1 = 12
li $s1, 18 # number2 = 18
add $t0, $s0, $s1 # $t0 = Sum of 12 and 18
# Print string "str1"
li $v0, 4
la $a0, str1
syscall
# Print $s0
li $v0, 1
move $a0, $s0
syscall
# Print string "str2"
li $v0, 4
la $a0, str2
syscall
# Print $s1
li $v0, 1
move $a0, $s1
syscall# Print string "str3"
li $v0, 4
la $a0, str3
syscall
```

→ Lần lượt thực hiện in từng xâu mà từng số nguyên lên màn hình để thu được kết quả như mong muốn





Text Segment

Bkpt	Address	Code	Basic	Source
	0x0040000c	0x24020004	addiu $2,$0,0x00000...	10: li $v0, 4
	0x00400010	0x3c011001	lui $1,0x00001001	11: la $a0, str1
	0x00400014	0x34240000	ori $4,$1,0x00000000	
	0x00400018	0x0000000c	syscall	12: syscall
	0x0040001c	0x24020001	addiu $2,$0,0x00000...	14: li $v0, 1
	0x00400020	0x00102021	addu $4,$0,$16	15: move $a0, $s0
	0x00400024	0x0000000c	syscall	16: syscall
	0x00400028	0x24020004	addiu $2,$0,0x00000...	18: li $v0, 4
	0x0040002c	0x3c011001	lui $1,0x00001001	19: la $a0, str2
	0x00400030	0x3424000c	ori $4,$1,0x0000000c	
	0x00400034	0x0000000c	syscall	20: syscall
	0x00400038	0x24020001	addiu $2,$0,0x00000...	22: li $v0, 1

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

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Mars Messages Run I/O

The sum of 12 and

Clear

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000004
$v1	3	0x00000000
$a0	4	0x1001000c
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x0000001e
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x0000000c
$s1	17	0x00000012
$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

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400018	0x0000000c	syscall	12: syscall
	0x0040001c	0x24020001	addiu $2,$0,0x00000...	14: li $v0, 1
	0x00400020	0x00102021	addu $4,$0,$16	15: move $a0, $s0
	0x00400024	0x0000000c	syscall	16: syscall
	0x00400028	0x24020004	addiu $2,$0,0x00000...	18: li $v0, 4
	0x0040002c	0x3c011001	lui $1,0x00001001	19: la $a0, str2
	0x00400030	0x3424000c	ori $4,$1,0x0000000c	
	0x00400034	0x0000000c	syscall	20: syscall
	0x00400038	0x24020001	addiu $2,$0,0x00000...	22: li $v0, 1
	0x0040003c	0x00112021	addu $4,$0,$17	23: move $a0, $s1
	0x00400040	0x0000000c	syscall	24: syscall# Print string "str3"
	0x00400044	0x24020004	addiu $2,$0,0x00000...	25: li $v0, 4

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

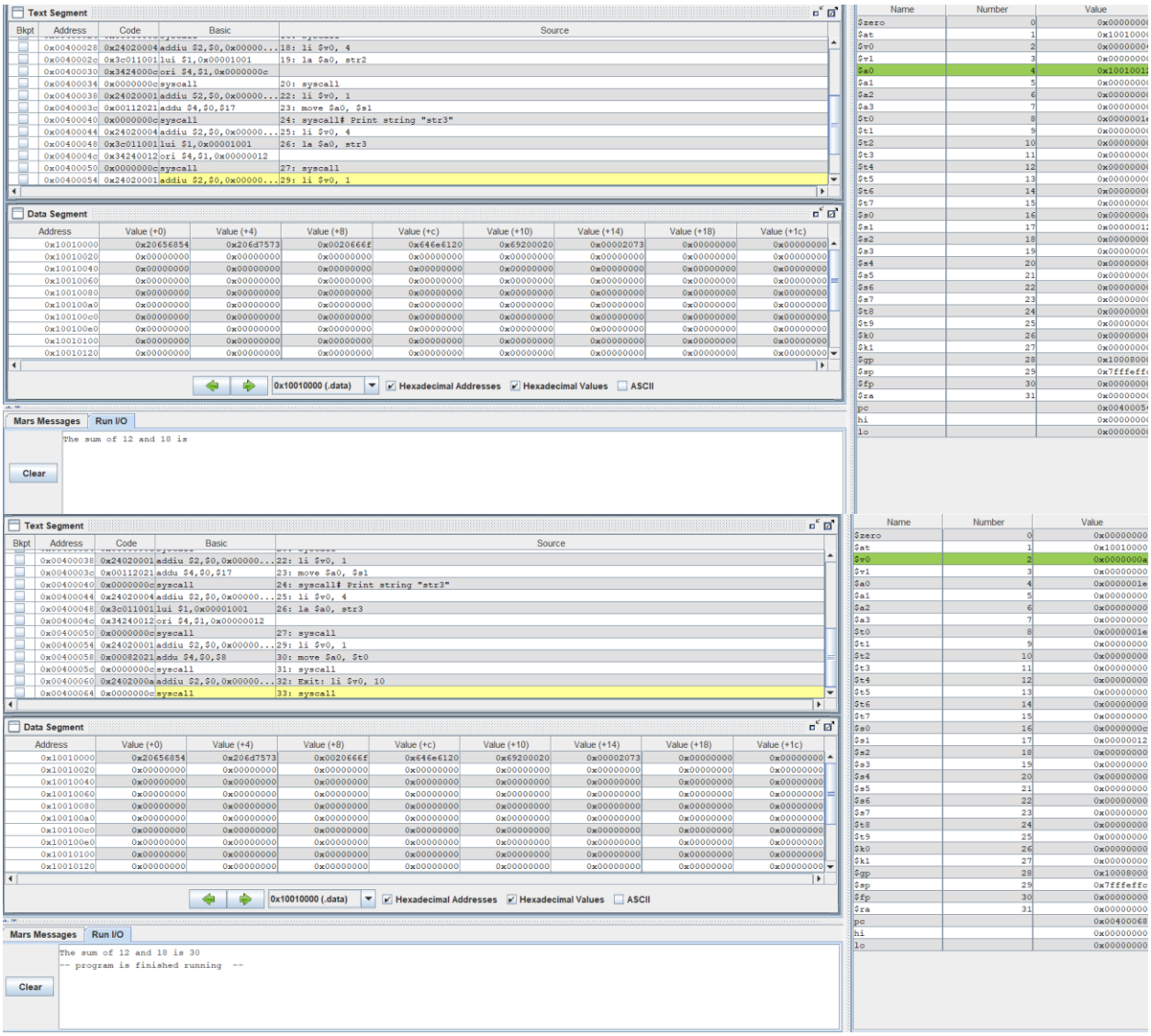
0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Mars Messages Run I/O

The sum of 12 and 18

Clear

Name	Number	Value
$zero	0	0x00000000
$at	1	0x10010000
$v0	2	0x00000001
$v1	3	0x00000000
$a0	4	0x00000012
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x0000001e
$t1	9	0x00000000
$t2	10	0x00000000
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x0000000c
$s1	17	0x00000012
$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		0x00400044
hi		0x00000000
lo		0x00000000



Assignment 3:

```
1  .data
2  x:
3          .space 100 # destination string x, empty
4  y:
5          .asciiz "Hello world" # source string y
6  .text
7  strcpy:
8          add $s0,$zero,$zero # $s0 = i = 0
9          la $a1, y # Load address of y to $a1
10         la $a0, x # Load address of x to $a0
11 Char:
12         add $t1,$s0,$a1 # $t1 = $s0 + $a1 = i + y[0]
13         # = address of y[i]
14         lb $t2,0($t1) # $t2 = value at $t1 = y[i]
15         add $t3,$s0,$a0 # $t3 = $s0 + $a0 = i + x[0]
16         # = address of x[i]
17         sb $t2,0($t3) # x[i]= $t2 = y[i]
18         beq $t2,$zero,end_of_strcpy # if y[i] == 0, exit
19         nop
20         addi $s0,$s0,1 # $s0 = $s0 + 1 <-> i = i + 1
21         j Char # next character
22         nop
23 end_of_strcpy:
24
```

Text Segment

Bkpt	Address	Code	Basic		Source
	0x00400000	0x00008020	add $16,$0,$0	8:	add $s0,$zero,$zero # $s0 = i = 0
	0x00400004	0x3c011001	lui $1,0x00001001	9:	la $a1, y # Load address of y to $a1
	0x00400008	0x34250064	ori $5,$1,0x00000064		
	0x0040000c	0x3c011001	lui $1,0x00001001	10:	la $a0, x # Load address of x to $a0
	0x00400010	0x34240000	ori $4,$1,0x00000000		
	0x00400014	0x02054820	add $9,$16,$5	12:	add $t1,$s0,$a1 # $t1 = $s0 + $a1 = i + y[0]
	0x00400018	0x812a0000	lb $10,0x00000000($9)	14:	lb $t2,0($t1) # $t2 = value at $t1 = y[i]
	0x0040001c	0x02045820	add $11,$16,$4	15:	add $t3,$s0,$a0 # $t3 = $s0 + $a0 = i + x[0]
	0x00400020	0xa16a0000	sb $10,0x00000000($...	17:	sb $t2,0($t3) # x[i]= $t2 = y[i]
	0x00400024	0x11400004	beq $10,$0,0x00000004	18:	beq $t2,$zero,end_of_strcpy # if y[i] == 0, exit
	0x00400028	0x00000000	nop	19:	nop
	0x0040002c	0x22100001	addi $16,$16,0x0000...	20:	addi $s0,$s0,1 # $s0 = $s0 + 1 <-> i = i + 1

Data Segment

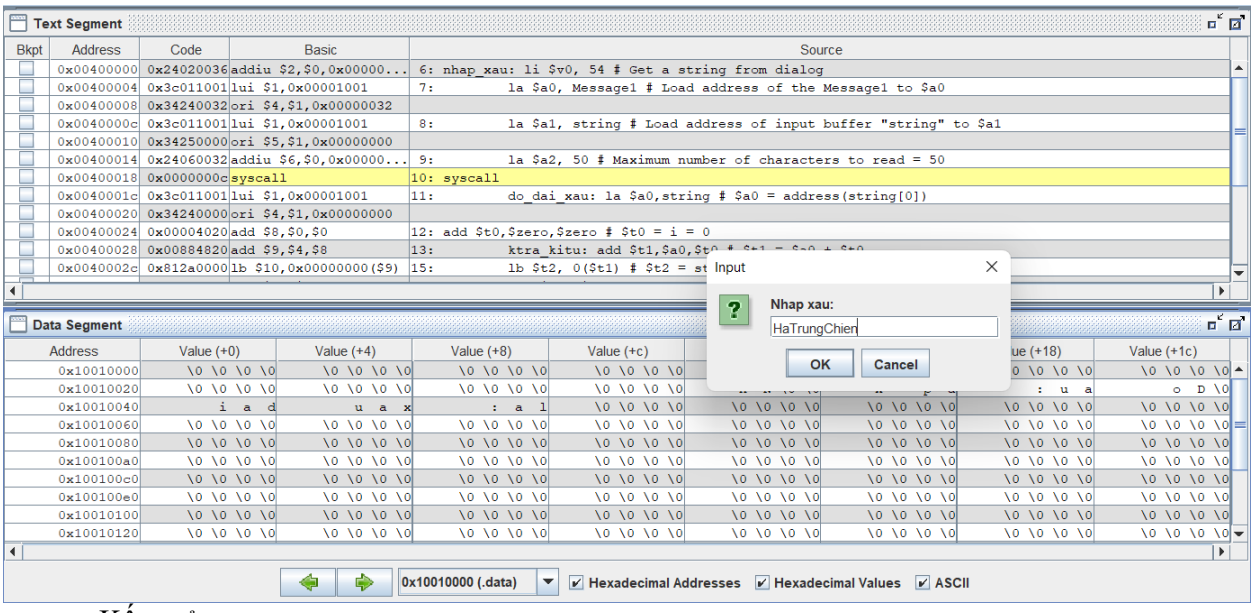
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	l l e H	o w o	\0 d l r	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010020	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0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

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

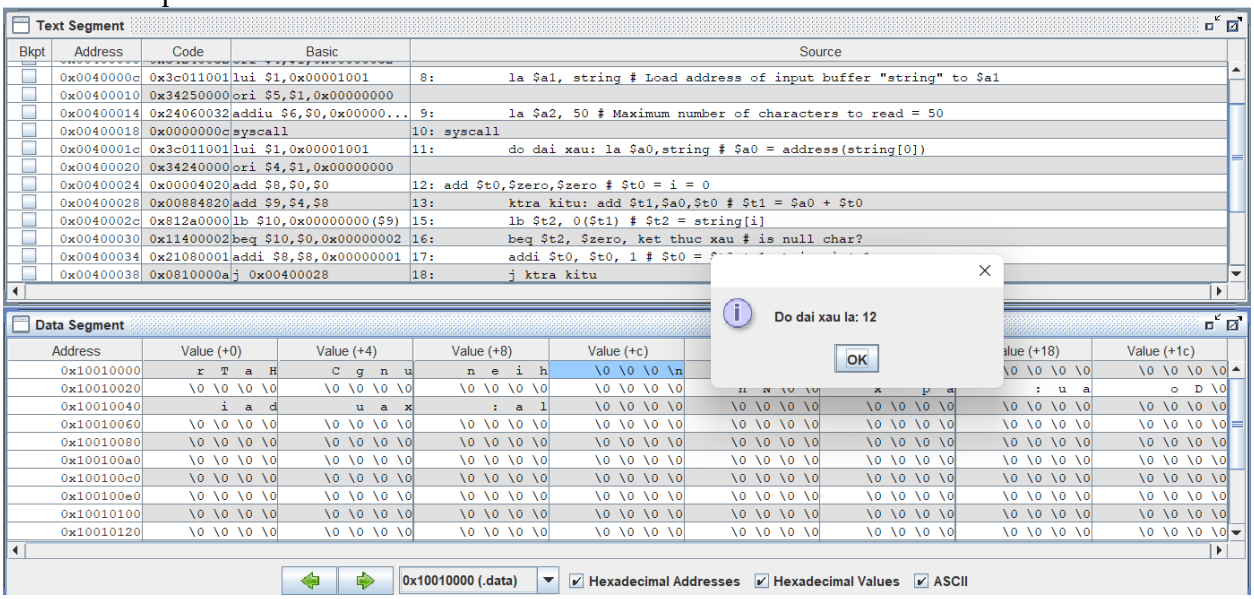
Assignment 4:

```
1  .data
2          string: .space 50
3          Message1: .asciiz "Nhap xau: "
4          Message2: .asciiz "Do dai xau la: "
5  .text
6  nhap_xau: li $v0, 54 # Get a string from dialog
7          la $a0, Message1 # Load address of the Message1 to $a0
8          la $a1, string # Load address of input buffer "string" to $a1
9          la $a2, 50 # Maximum number of characters to read = 50
10 syscall
11         do_dai_xau: la $a0,string # $a0 = address(string[0])
12 add $t0,$zero,$zero # $t0 = i = 0
13         ktra_kitu: add $t1,$a0,$t0 # $t1 = $a0 + $t0
14         # = address(string[i])
15         lb $t2, 0($t1) # $t2 = string[i]
16         beq $t2, $zero, ket_thuc_xau # is null char?
17         addi $t0, $t0, 1 # $t0 = $t0 + 1 -> i = i + 1
18         j ktra_kitu
19 ket_thuc_xau:
20 in_xau:
21         addi $t0, $t0, -1
22         li $v0, 56
23         la $a0, Message2
24         move $a1, $t0
25         syscall
```

- Nhập xâu:



→ Kết quả:



→ Nhận xét:

- Nhập xâu “HaTrungChien” có 12 kí tự ta thu được kết quả độ dài xâu là 12.
- Giá trị của thanh ghi $t0 tăng từ 0 đến 12 thì dừng. Là vì khi kiểm tra tới kí tự NULL ở câu lệnh (beq $t2, $zero, end_of_str) thì nhảy đến ket_thuc_xâu.
- Kết quả thu được tương tự khi kiểm tra những xâu khác => đúng với lý thuyết.

Assignment 5:

mips1.asm

```
.data
string: .space 20
Message1: .asciiz "Nhap ky tu thu "
Message2: .asciiz ": "
Message3: .asciiz "\n"
Message4: .asciiz "Chuoi ky tu sau khi dao la: "

.text
        li $s0, 20      #do dai xau toi da
        li $s1, 0       #load i
        la $s2, string  #load address(string(0)) = $s2
        li $s3, 10
doc_kitu:
        beq $s1, $s0, ket_thuc_doc      #if i = max_length -> end
        li $v0, 4
        la $a0, Message1
        syscall         #in Message1
        addi $t1, $s1, 1        #i = i+1
        li $v0, 1
        move $a0, $t1
        syscall                 #in so thu tu ki tu
        li $v0, 4
        la $a0, Message2
        syscall                 #in Message2
        li $v0, 12
        syscall                 #doc ki tu
        move $t0, $v0
```

mips1.asm

```
        li $v0, 12
        syscall                 #doc ki tu
        move $t0, $v0
        beq $t0, $s3, ket_thuc_doc      # $t0 = 10 -> exit
        li $v0, 4
        la $a0, Message3
        syscall                 #in Message3
        add $s5, $s2, $s1       #$s5 = s2 + s1 -> $s5 = a(s(0)) + i
        sb $t0, 0($s5)
        addi $s1, $s1, 1        #i = i+1
        j doc_kitu
ket_thuc_doc:
        li $v0, 4
        la $a0, Message4
        syscall
in_xau:
        li $v0, 11
        lb $a0, 0($s5)
        syscall                 #in kitu
        beq $s5, $s2, exit
        addi $s5, $s5, -1
        j in_xau
exit:
        li $v0, 10
        syscall
```

- TH1: Độ dài xâu bé hơn 20

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x24100014	addiu $16,$0,20	9: li $s0, 20 #do dai xau toi da
	0x00400004	0x24110000	addiu $17,$0,0	10: li $s1, 0 #load i
	0x00400008	0x3c011001	lui $1,4097	11: la $s2, string #load address(string(0)) = $s2
	0x0040000c	0x34320000	ori $18,$1,0	
	0x00400010	0x2413000a	addiu $19,$0,10	12: li $s3, 10
	0x00400014	0x12300018	beq $17,$16,24	14: beq $s1, $s0, ket_thuc_doc #if i = max_length -> end
	0x00400018	0x24020004	addiu $2,$0,4	15: li $v0, 4
	0x0040001c	0x3c011001	lui $1,4097	16: la $a0, Message1
	0x00400020	0x34240014	ori $4,$1,20	
	0x00400024	0x0000000c	syscall	17: syscall #in Message1
	0x00400028	0x22290001	addi $9,$17,1	18: addi $t1, $s1, 1 #i = i+1
	0x0040002c	0x24020001	addiu $2,$0,1	19: li $v0, 1

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	r T a H	C g n u	n e i h	\0 \0 \0 \0	\0 \0 \0 \0	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	u a s	i h k	o a d
0x10010040	: a l	\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

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Mars Messages Run I/O

Clear

```
Nhap ky tu thu 6: n
Nhap ky tu thu 7: g
Nhap ky tu thu 8: C
Nhap ky tu thu 9: h
Nhap ky tu thu 10: i
Nhap ky tu thu 11: e
Nhap ky tu thu 12: n
Nhap ky tu thu 13:
Chuoi ky tu sau khi dao la: neihCgnurTaH
```

→ Nhập Xâu “HaTrungChien” thu được kết quả “neihCgnurTaH” đúng với lí thuyết.

- TH2: Độ dài xâu lớn hơn 20

mips1.asm

```
27        move $t0, $v0
28        beq $t0, $s3, ket_thuc_doc     # $t0 = 10 -> exit
29        li $v0, 4
30        la $a0, Message3
31        syscall                 #in Message3
32        add $s5, $s2, $s1       #$s5 = s2 + s1 -> $s5 = a(s(0)) + i
33        sb $t0, 0($s5)
34        addi $s1, $s1, 1        #i = i+1
35        j doc_kitu
36    ket_thuc_doc:
37        li $v0, 4
38        la $a0, Message4
39        syscall
40    in_xau:
41        li $v0, 11
42        lb $a0, 0($s5)
43        syscall                 #in kitu
44        beq $s5, $s2, exit
45        addi $s5, $s5, -1
46        j in_xau
47    exit:
48        li $v0, 10
49        syscall
50
51
```

Line: 20 Column: 17 Show Line Numbers

Mars Messages Run I/O

Clear

```
Nhap ky tu thu 13: n
Nhap ky tu thu 14: o
Nhap ky tu thu 15: p
Nhap ky tu thu 16: q
Nhap ky tu thu 17: r
Nhap ky tu thu 18: s
Nhap ky tu thu 19: t
Nhap ky tu thu 20: y
Chuoi ky tu sau khi dao la: ytsrqponmlkihgfedcba
```

→ Chỉ có thể nhập được tối đa 20 kí tự. ta thu được kết quả như hình.