

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

\*\*\*\*\*



**BÁO CÁO**  
**Bài tập thực hành tuần 3**  
**Học phần: Thực hành kiến trúc máy tính**

**Giảng viên hướng dẫn:** Lê Bá Vui

**Sinh viên thực hiện:** Phạm Huy Cảnh - 20194490

**Mã lớp:** 130938

**Hà Nội, tháng 4 năm 2022**

# 1. Assignment 1:

```
HomeAssignment1.asm
1 # Laboratory Exercise 3, Home Assignment 1
2 # Author: Pham Huy Canh
3 .data
4     x: .word 1
5     y: .word 1
6     z: .word 1
7 .text
8     addi $s1, $zero, 3      # Khoi tao i = 3
9     addi $s2, $zero, 4      # Khoi tao j = 4
10    la $a0, x              # Lay dia chi cua x
11    lw $t1, 0($a0)          # Load gia tri cua x vao thanh ghi $t1
12    la $a0, y              # Lay dia chi cua y
13    lw $t2, 0($a0)          # Load gia tri cua y vao thanh ghi $t2
14    la $a0, z              # Lay dia chi cua z
15    lw $t3, 0($a0)          # Load gia tri cua z vao thanh ghi $t3
16
17 start:
18     slt $t0,$s2,$s1        # j<i
19     bne $t0,$zero,else     # branch to else if j<i
20     addi $t1,$t1,1          # then part: x=x+1
21     addi $t3,$zero,1          # z=1
22     j endif                # skip "else" part
23 else:
24     addi $t2,$t2,-1          # begin else part: y=y-1
25     add $t3,$t3,$t3          # z=2*z
26 endif:
27
28
29
```

Giả sử ta có  $x = y = z = 1$  và khởi tạo  $i = 3, j = 4$

- Sau khi chạy các câu lệnh:

addi \$s1, \$zero, 3

addi \$s2, \$zero, 4

The screenshot shows the QEMU debugger interface with three main panes:

- Registers:** Shows the state of various寄存器 (Registers). The \$s1 and \$s2 registers are highlighted in green, both containing the value 3.
- Labels:** Shows the labels defined in the assembly code, such as start, else, endif, and the labels for variables x, y, and z.
- Text Segment:** Shows the assembly code with line numbers and comments. The first two instructions are highlighted in red: addi \$s1, \$zero, 3 and addi \$s2, \$zero, 4.
- Data Segment:** Shows the memory dump with address ranges from 0x10000000 to 0x10010120. The first four bytes at address 0x10000000 are shown as 1, 1, 1, 1.

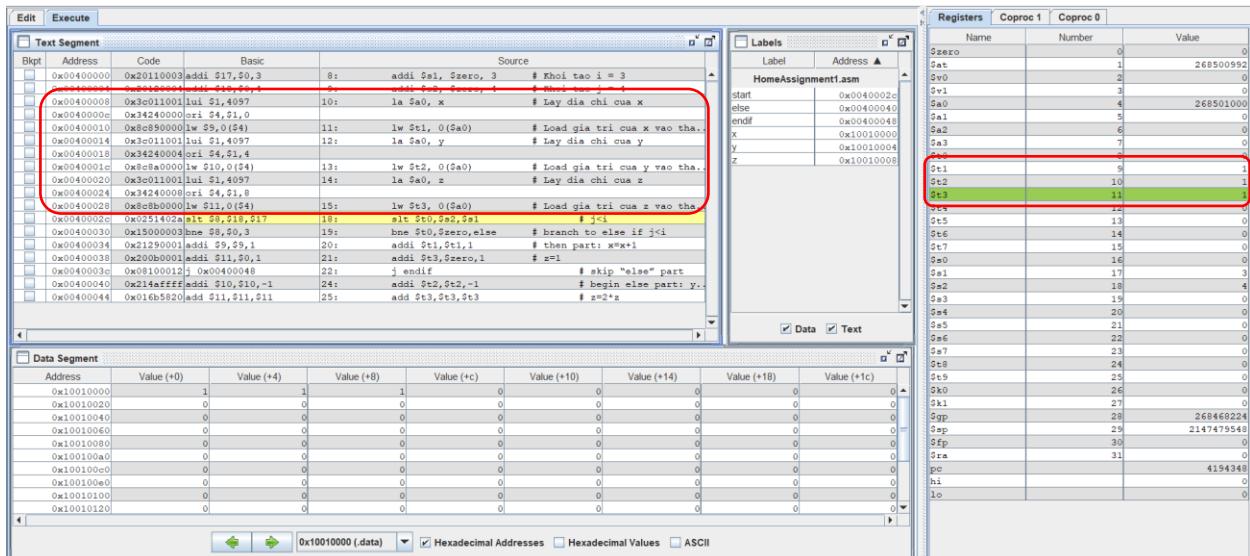
Ta thấy: Hai thanh ghi thay đổi giá trị là  $\$s1 = 3$ ,  $\$s2 = 4$  (Tức là khởi tạo giá trị  $i = 3$  và  $j = 4$ )

### - Sau khi chạy các câu lệnh:

```

la $a0, x
lw $t1, 0($a0)
la $a0, y
lw $t2, 0($a0)
la $a0, z
lw $t3, 0($a0)

```



Ta thấy: Sau khi thực hiện các câu lệnh `la` để lấy địa chỉ của ô nhớ và `lw` để ghi giá trị đang nằm trong ô nhớ đó ra thanh ghi. Kết quả, giá trị x, y, z trong ô nhớ được ghi vào trong các thanh ghi  $\$t1$ ,  $\$t2$ ,  $\$t3$

### - Sau khi chạy lệnh:

slt \$t0, \$s2, \$s1

The screenshot shows the QEMU debugger interface with several panes:

- Text Segment:** Displays assembly code for the `HomeAssignment1.asm` file. A specific instruction at address 0x0400002c is highlighted with a red box: `slt $t0,$s2,$s1`. This indicates that the condition  $s2 < s1$  was true.
- Registers:** Shows the register values. The `$t0` register is highlighted with a red box and has a value of 0.
- Data Segment:** Displays memory dump information for the `.data` section.

Ta thấy: Ở câu lệnh này sẽ thực hiện so sánh  $\$s2 < \$s1$  (tức  $j < i$ ), nếu đúng thì thanh ghi `$t0` được gán giá trị bằng 1, nếu sai thanh ghi `$t0` được gán giá trị bằng 0. Sau khi chạy xong câu lệnh ta được kết quả  $\$t0 = 0$  do  $\$s2 < \$s1$  ( $4 < 3$ ) là sai.

### - Sau khi thực hiện câu lệnh:

bne \$t0, \$zero, else

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x10010008
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
<b>\$t0</b>	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000001
\$t3	11	0x00000001
\$s4	12	0x00000000
\$t5	13	0x00000000
\$s6	14	0x00000000
\$t7	15	0x00000000
\$a0	16	0x00000000
\$a1	17	0x00000003
\$a2	18	0x00000004
\$a3	19	0x00000000
\$a4	20	0x00000000
\$a5	21	0x00000000
\$a6	22	0x00000000
\$a7	23	0x00000000
\$s8	24	0x00000000
\$s9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10000000
\$sp	29	0x7ffffefc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x0400034
hi		0x00000000
lo		0x00000000

Ta thấy: Câu lệnh so sánh thanh ghi \$t0 với thanh ghi \$zero, nếu không bằng nhau thì sẽ nhảy đến label else, ở đây ta thấy thanh ghi pc lúc này có giá trị bằng địa chỉ của câu lệnh ngay sau đó nên lệnh nhảy đến else không được thực hiện vì **\$t0 = 0x00000000 = \$zero**

### - Sau khi chạy câu lệnh:

addi \$t1,\$t1,1

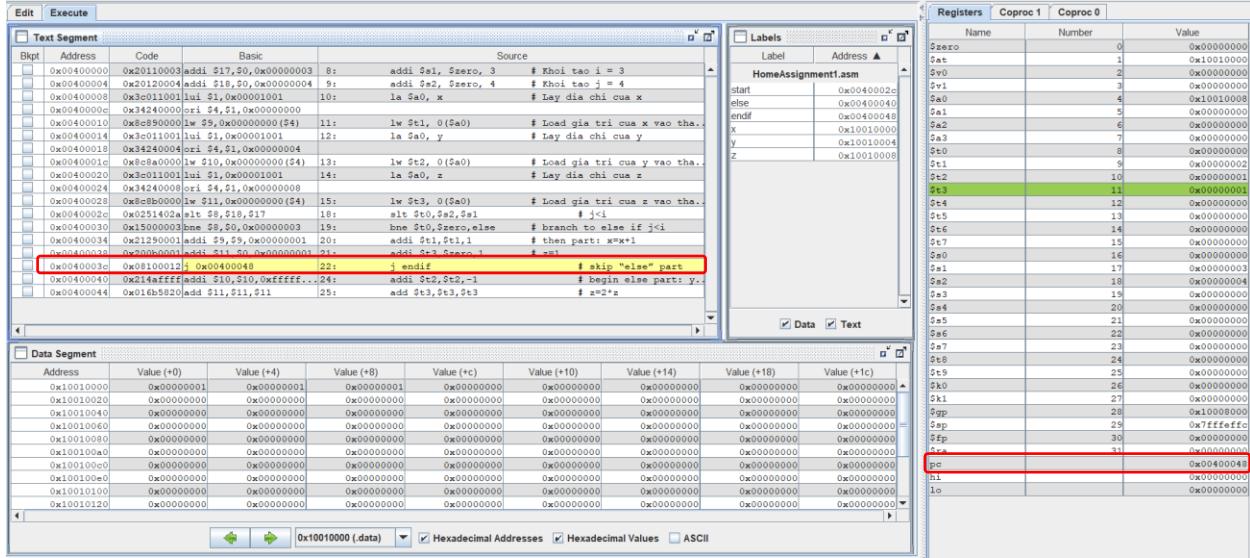
addi \$t3,\$zero,1

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x10010008
\$a1	5	0x00000000
\$a2	6	0x00000000
<b>\$t0</b>	7	0x00000000
\$t1	8	0x00000000
<b>\$t2</b>	9	0x00000001
<b>\$t3</b>	11	0x00000001
\$t4	12	0x00000000
\$t5	13	0x00000000
\$s6	14	0x00000000
\$t7	15	0x00000000
\$a0	16	0x00000000
\$a1	17	0x00000003
\$a2	18	0x00000004
\$a3	19	0x00000000
\$a4	20	0x00000000
\$a5	21	0x00000000
\$a6	22	0x00000000
\$a7	23	0x00000000
\$s8	24	0x00000000
\$s9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10000000
\$sp	29	0x7ffffefc
\$fp	30	0x10000000
\$ra	31	0x00000000
pc		0x0400034
hi		0x00000000
lo		0x00000000

Ta thấy: Hai câu lệnh trên thực hiện hai phép toán  $x = x + 1$  và  $z = 1$ . Lúc này thanh ghi **\$t1 = 0x00000002** và **\$t3 = 0x00000001**

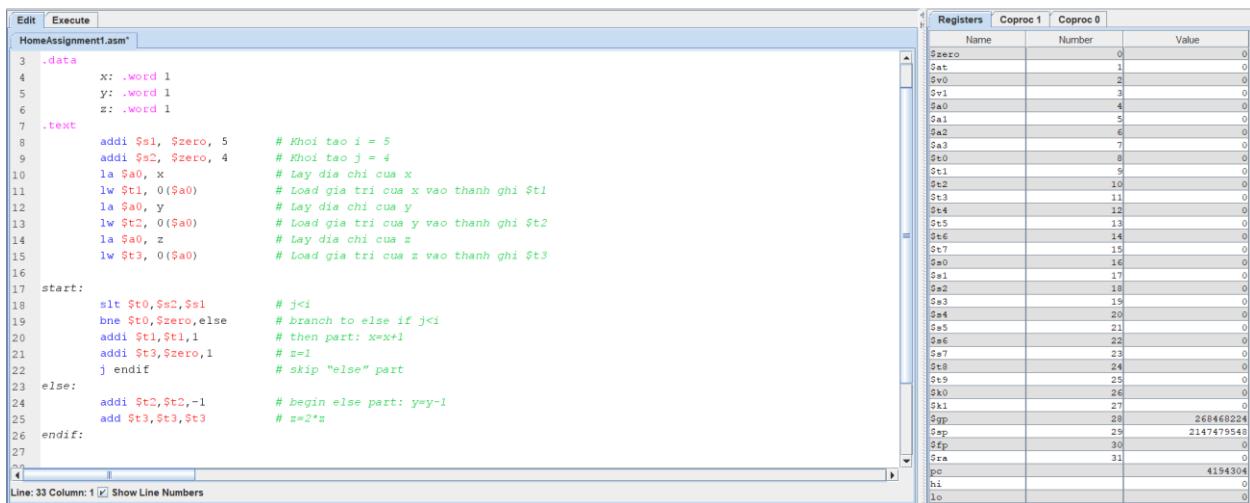
## - Sau khi chạy câu lệnh:

j endif

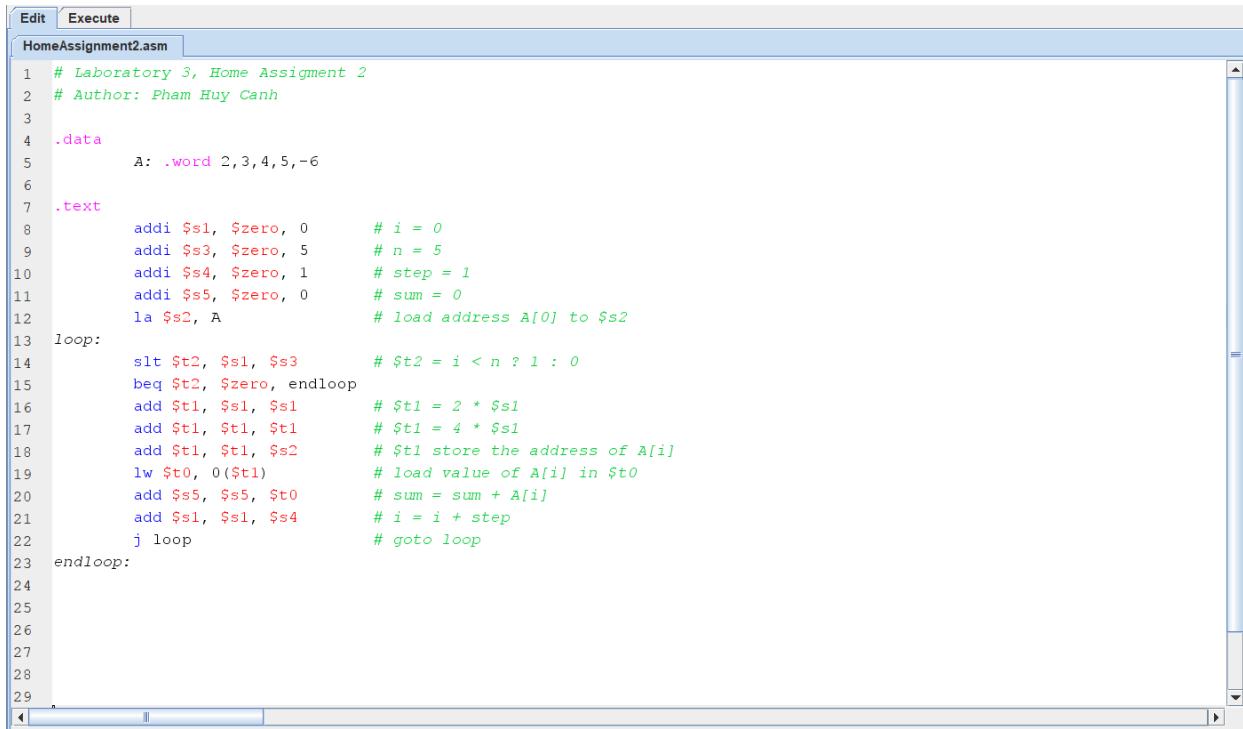


Ta thấy: Câu lệnh này dùng để nhảy đến label `endif`, ở đây thanh ghi `pc` đã trỏ đến địa chỉ của câu nhãn `endif`, bỏ qua các câu lệnh trước đó và kết thúc chương trình.

Khi thay đổi giá trị khởi tạo của `i` và `j` ( $i = 5, j = 4$ ) thì tương tự như vậy sau khi thực hiện câu lệnh thì lúc này `$t0 = 1` khác 0 và sẽ nhảy sang nhãn `else` để thực hiện các câu lệnh sau nhãn đó.



## 2. Assignment 2:



The screenshot shows a text editor window titled "Edit Execute" with the file "HomeAssignment2.asm". The code is written in assembly language and performs the following steps:

- Defines a data section with a word-sized array A containing elements 2, 3, 4, 5, -6.
- Initializes registers \$s1, \$s3, \$s4, \$s5 to 0, and \$s2 to the address of A[0].
- Enters a loop where it checks if \$t2 (i) is less than \$s3 (n). If true, it adds the value at memory location \$t1 to \$s5 (sum).
- If \$t2 is not less than \$s3, it exits the loop.
- Updates \$t1 to \$t1 + 4, \$t2 to \$t2 + 1, and \$s1 to \$s1 + 1.
- Loads the value at memory location \$t1 into \$t0.
- Updates the sum by adding \$t0 to \$s5.
- Updates \$s1 to \$s1 + step.
- Jumps back to the beginning of the loop.

*Khởi tạo mảng A gồm 5 phần tử {2, 3, 4, 5, -6} và i = 0, n = 5, step = 1,  
sum = 0*

- Sau khi chạy lệnh:

```
addi $s1, $zero, 0
addi $s3, $zero, 5
addi $s4, $zero, 1
addi $s5, $zero, 0
```

The screenshot shows the QEMU debugger interface with several windows:

- Text Segment**: Shows assembly code for `HomeAssignment2.asm`. A specific instruction at address `0x0400010` is highlighted with a red box.
- Labels**: Lists labels from the assembly code, with `$s1`, `$s3`, `$s4`, and `$s5` highlighted in red.
- Registers**: Shows the CPU register state. The registers `$s1`, `$s3`, `$s4`, and `$s5` are highlighted in red.
- Data Segment**: Shows memory dump data for the range `0x10010000 - 0x10010120`.

Các thanh ghi được gán giá trị `$s1`, `$s3`, `$s4`, `$s5`

- Sau khi chạy lệnh:

la `$s2`, A

The screenshot shows the QEMU debugger interface after executing the `la $s2, A` command. The registers pane now shows the following values for the highlighted registers:

Name	Value
<code>\$s1</code>	<code>0x10010000</code>
<code>\$s3</code>	<code>0x00000005</code>
<code>\$s4</code>	<code>0x00000001</code>
<code>\$s5</code>	<code>0x00000000</code>

Địa chỉ của mảng A được nạp vào thanh ghi `$s2`, đây cũng chính là địa chỉ cơ sở `A[0]`

- Sau khi chạy lệnh:

`slt $t2, $s1, $s3`

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$s0	8	0x00000000
\$s1	9	0x00000000
<b>\$t2</b>	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$a0	16	0x00000000
\$a1	17	0x00000000
\$a2	18	0x10010000
\$a3	19	0x00000005
\$s4	20	0x00000001
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10000000
\$sp	29	0x7ffffefc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x0400001c
hi		0x00000000
lo		0x00000000

Labels HomeAssignment2.asm

Label	Address
loop	0x04000018
endloop	0x040003c
A	0x10010000

Text Segment

Bkpt	Address	Code	Basic	Source
	0x04000000	0x20110000	addi \$17,\$0,0x0000...	8: addi \$s1, \$zero, 0 # i = 0
	0x04000004	0x20130005	addi \$19,\$0,0x0000...	9: addi \$s3, \$zero, 5 # n = 5
	0x04000008	0x20140001	addi \$20,\$0,0x0000...	10: addi \$s4, \$zero, 1 # step = 1
	0x0400000c	0x20150000	addi \$21,\$0,0x0000...	11: addi \$s5, \$zero, 0 # sum = 0
	0x04000010	0x3c010001	lui \$1,0x00001001	12: la \$s2, A # load address A[0] to \$s2
	0x04000014	0x11400007	beq \$t2,\$zero,endloop	14: beq \$t2, \$zero, endloop # t2 = i < n ? 1 : 0
	0x04000020	0x02334820	add \$5,\$17,\$17	15: add \$t1, \$s1, \$s1 # t1 = 2 * s1
	0x04000024	0x01294820	add \$9,\$9,29	17: add \$t1, \$t1, \$t1 # t1 = 4 * s1
	0x04000028	0x01324820	add \$9,\$9,\$18	18: add \$t1, \$t1, \$s2 # t1 store the address of A[i]
	0x0400002c	0x08280000	lw \$0,0x00000000(\$9)	19: lw \$t0, 0(\$t1) # load value of A[i] in \$t0
	0x04000030	0x023a8820	add \$21,\$21,\$8	20: add \$s5, \$s5, \$t0 # sum = sum + A[i]
	0x04000034	0x02348820	add \$17,\$17,\$20	21: add \$s1, \$s1, \$s4 # i = i + step
	0x04000038	0x05100006	j 0x04000018	22: j loop # goto loop

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000002	0x00000003	0x00000004	0x00000005	0xfffffff6	0x00000000	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

Câu lệnh kiểm tra  $i < n$  không. Vì  $\$s1 < \$s3$  nên  $\$t2 = 1$

- Sau khi chạy lệnh:

beq \$t2, \$zero, endloop

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$s0	8	0x00000000
\$s1	9	0x00000000
<b>\$t2</b>	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$a0	16	0x00000000
\$a1	17	0x00000000
\$a2	18	0x10010000
\$a3	19	0x00000005
\$s4	20	0x00000001
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10000000
\$sp	29	0x7ffffefc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x0400001c
hi		0x00000000
lo		0x00000000

Labels HomeAssignment2.asm

Label	Address
loop	0x04000018
endloop	0x040003c
A	0x10010000

Text Segment

Bkpt	Address	Code	Basic	Source
	0x04000000	0x20110000	addi \$17,\$0,0x0000...	8: addi \$s1, \$zero, 0 # i = 0
	0x04000004	0x20130005	addi \$19,\$0,0x0000...	9: addi \$s3, \$zero, 5 # n = 5
	0x04000008	0x20140001	addi \$20,\$0,0x0000...	10: addi \$s4, \$zero, 1 # step = 1
	0x0400000c	0x20150000	addi \$21,\$0,0x0000...	11: addi \$s5, \$zero, 0 # sum = 0
	0x04000010	0x3c010001	lui \$1,0x00001001	12: la \$s2, A # load address A[0] to \$s2
	0x04000014	0x11400007	beq \$t2,\$zero,endloop	14: beq \$t2, \$zero, endloop # t2 = i < n ? 1 : 0
	0x04000020	0x02334820	add \$5,\$17,\$17	15: add \$t1, \$s1, \$s1 # t1 = 2 * s1
	0x04000024	0x01294820	add \$9,\$9,29	17: add \$t1, \$t1, \$t1 # t1 = 4 * s1
	0x04000028	0x01324820	add \$9,\$9,\$18	18: add \$t1, \$t1, \$s2 # t1 store the address of A[i]
	0x0400002c	0x08280000	lw \$0,0x00000000(\$9)	19: lw \$t0, 0(\$t1) # load value of A[i] in \$t0
	0x04000030	0x023a8820	add \$21,\$21,\$8	20: add \$s5, \$s5, \$t0 # sum = sum + A[i]
	0x04000034	0x02348820	add \$17,\$17,\$20	21: add \$s1, \$s1, \$s4 # i = i + step
	0x04000038	0x05100006	j 0x04000018	22: j loop # goto loop

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000002	0x00000003	0x00000004	0x00000005	0xfffffff6	0x00000000	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
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Vì  $\$t2$  khác  $\$zero$  nên thanh ghi pc trỏ đến địa chỉ của câu lệnh tiếp theo mà không nhảy đến nhãn `endloop`

- Sau khi chạy lệnh:

add \$t1, \$s1, \$s1

add \$t1, \$t1, \$t1

add \$t1, \$s1, \$s2

Registers:

Name	Number	Value
\$zero	0	0x00000000
\$t0	1	0x10010000
\$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	0x10010000
\$s3	19	0x00000005
\$s4	20	0x00000001
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$s8	24	0x00000000
\$s9	25	0x00000000
\$s10	26	0x00000000
\$s11	27	0x00000000
\$sp	28	0x10008000
\$fp	29	0x7ffffefc
\$ra	30	0x00000000
pc	31	0x00000000
hi		0x040002c
lo		0x00000000

$\$t1$  đang lưu địa chỉ của  $A[i]$

### - Sau khi chạy lệnh:

`lw $t0, 0($t1)`

Registers:

Name	Number	Value
\$zero	0	0x00000000
\$t0	1	0x10010000
\$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	0x10010000
\$s3	19	0x00000005
\$s4	20	0x00000001
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$s8	24	0x00000000
\$s9	25	0x00000000
\$s10	26	0x00000000
\$s11	27	0x00000000
\$sp	28	0x10008000
\$fp	29	0x7ffffefc
\$ra	30	0x00000000
pc	31	0x00000000
hi		0x0400030
lo		0x00000000

Câu lệnh lấy dữ liệu trong ô nhớ có địa chỉ  $\$t1$  và nạp vào  $\$t0$

### - Sau khi chạy lệnh:

`add $s5, $s5, $t0`

`add $s1, $s1, $s4`

<img alt="Screenshot of the QEMU debugger showing assembly code for HomeAssignment2.asm. The registers window shows \$zero=0, \$at=1, \$v0=2, \$v1=3, \$a0=4, \$a1=5, \$a2=6, \$a3=7, \$t0=8, \$t1=9, \$t2=10, \$s1=11, \$s2=12, \$s3=13, \$s4=14, \$s5=15, \$t3=16, \$t4=17, \$t5=18, \$t6=19, \$t7=20, \$t8=21, \$t9=22, \$t10=23, \$t11=24, \$t12=25, \$t13=26, \$t14=27, \$t15=28, \$t16=29, \$t17=30, \$t18=31, \$t19=32, \$t20=33, \$t21=34, \$t22=35, \$t23=36, \$t24=37, \$t25=38, \$t26=39, \$t27=40, \$t28=41, \$t29=42, \$t30=43, \$t31=44, \$t32=45, \$t33=46, \$t34=47, \$t35=48, \$t36=49, \$t37=50, \$t38=51, \$t39=52, \$t40=53, \$t41=54, \$t42=55, \$t43=56, \$t44=57, \$t45=58, \$t46=59, \$t47=60, \$t48=61, \$t49=62, \$t50=63, \$t51=64, \$t52=65, \$t53=66, \$t54=67, \$t55=68, \$t56=69, \$t57=70, \$t58=71, \$t59=72, \$t60=73, \$t61=74, \$t62=75, \$t63=76, \$t64=77, \$t65=78, \$t66=79, \$t67=80, \$t68=81, \$t69=82, \$t70=83, \$t71=84, \$t72=85, \$t73=86, \$t74=87, \$t75=88, \$t76=89, \$t77=90, \$t78=91, \$t79=92, \$t7a=93, \$t7b=94, \$t7c=95, \$t7d=96, \$t7e=97, \$t7f=98, \$t7g=99, \$t7h=100, \$t7i=101, \$t7j=102, \$t7k=103, \$t7l=104, \$t7m=105, \$t7n=106, \$t7o=107, \$t7p=108, \$t7q=109, \$t7r=110, \$t7s=111, \$t7t=112, \$t7u=113, \$t7v=114, \$t7w=115, \$t7x=116, \$t7y=117, \$t7z=118, \$t7aa=119, \$t7ab=120, \$t7ac=121, \$t7ad=122, \$t7ae=123, \$t7af=124, \$t7ag=125, \$t7ah=126, \$t7ai=127, \$t7aj=128, \$t7ak=129, \$t7al=130, \$t7am=131, \$t7an=132, \$t7ao=133, \$t7ap=134, \$t7aq=135, \$t7ar=136, \$t7as=137, \$t7at=138, \$t7au=139, \$t7av=140, \$t7aw=141, \$t7ax=142, \$t7ay=143, \$t7az=144, \$t7ba=145, \$t7ca=146, \$t7da=147, \$t7ea=148, \$t7fa=149, \$t7ga=150, \$t7ha=151, \$t7ia=152, \$t7ja=153, \$t7ka=154, \$t7la=155, \$t7ma=156, \$t7na=157, \$t7oa=158, \$t7pa=159, \$t7qa=160, \$t7ra=161, \$t7ua=162, \$t7va=163, \$t7wa=164, \$t7xa=165, \$t7ya=166, \$t7za=167, \$t7ba=168, \$t7ca=169, \$t7da=170, \$t7ea=171, \$t7fa=172, \$t7ga=173, \$t7ha=174, \$t7ia=175, \$t7ja=176, \$t7ka=177, \$t7la=178, \$t7ma=179, \$t7na=180, \$t7oa=181, \$t7pa=182, \$t7qa=183, \$t7ra=184, \$t7ua=185, \$t7va=186, \$t7wa=187, \$t7xa=188, \$t7ya=189, \$t7za=190, \$t7ba=191, \$t7ca=192, \$t7da=193, \$t7ea=194, \$t7fa=195, \$t7ga=196, \$t7ha=197, \$t7ia=198, \$t7ja=199, \$t7ka=200, \$t7la=201, \$t7ma=202, \$t7na=203, \$t7oa=204, \$t7pa=205, \$t7qa=206, \$t7ra=207, \$t7ua=208, \$t7va=209, \$t7wa=210, \$t7xa=211, \$t7ya=212, \$t7za=213, \$t7ba=214, \$t7ca=215, \$t7da=216, \$t7ea=217, \$t7fa=218, \$t7ga=219, \$t7ha=220, \$t7ia=221, \$t7ja=222, \$t7ka=223, \$t7la=224, \$t7ma=225, \$t7na=226, \$t7oa=227, \$t7pa=228, \$t7qa=229, \$t7ra=230, \$t7ua=231, \$t7va=232, \$t7wa=233, \$t7xa=234, \$t7ya=235, \$t7za=236, \$t7ba=237, \$t7ca=238, \$t7da=239, \$t7ea=240, \$t7fa=241, \$t7ga=242, \$t7ha=243, \$t7ia=244, \$t7ja=245, \$t7ka=246, \$t7la=247, \$t7ma=248, \$t7na=249, \$t7oa=250, \$t7pa=251, \$t7qa=252, \$t7ra=253, \$t7ua=254, \$t7va=255, \$t7wa=256, \$t7xa=257, \$t7ya=258, \$t7za=259, \$t7ba=260, \$t7ca=261, \$t7da=262, \$t7ea=263, \$t7fa=264, \$t7ga=265, \$t7ha=266, \$t7ia=267, \$t7ja=268, \$t7ka=269, \$t7la=270, \$t7ma=271, \$t7na=272, \$t7oa=273, \$t7pa=274, \$t7qa=275, \$t7ra=276, \$t7ua=277, \$t7va=278, \$t7wa=279, \$t7xa=280, \$t7ya=281, \$t7za=282, \$t7ba=283, \$t7ca=284, \$t7da=285, \$t7ea=286, \$t7fa=287, \$t7ga=288, \$t7ha=289, \$t7ia=290, \$t7ja=291, \$t7ka=292, \$t7la=293, \$t7ma=294, \$t7na=295, \$t7oa=296, \$t7pa=297, \$t7qa=298, \$t7ra=299, \$t7ua=300, \$t7va=301, \$t7wa=302, \$t7xa=303, \$t7ya=304, \$t7za=305, \$t7ba=306, \$t7ca=307, \$t7da=308, \$t7ea=309, \$t7fa=310, \$t7ga=311, \$t7ha=312, \$t7ia=313, \$t7ja=314, \$t7ka=315, \$t7la=316, \$t7ma=317, \$t7na=318, \$t7oa=319, \$t7pa=320, \$t7qa=321, \$t7ra=322, \$t7ua=323, \$t7va=324, \$t7wa=325, \$t7xa=326, \$t7ya=327, \$t7za=328, \$t7ba=329, \$t7ca=330, \$t7da=331, \$t7ea=332, \$t7fa=333, \$t7ga=334, \$t7ha=335, \$t7ia=336, \$t7ja=337, \$t7ka=338, \$t7la=339, \$t7ma=340, \$t7na=341, \$t7oa=342, \$t7pa=343, \$t7qa=344, \$t7ra=345, \$t7ua=346, \$t7va=347, \$t7wa=348, \$t7xa=349, \$t7ya=350, \$t7za=351, \$t7ba=352, \$t7ca=353, \$t7da=354, \$t7ea=355, \$t7fa=356, \$t7ga=357, \$t7ha=358, \$t7ia=359, \$t7ja=360, \$t7ka=361, \$t7la=362, \$t7ma=363, \$t7na=364, \$t7oa=365, \$t7pa=366, \$t7qa=367, \$t7ra=368, \$t7ua=369, \$t7va=370, \$t7wa=371, \$t7xa=372, \$t7ya=373, \$t7za=374, \$t7ba=375, \$t7ca=376, \$t7da=377, \$t7ea=378, \$t7fa=379, \$t7ga=380, \$t7ha=381, \$t7ia=382, \$t7ja=383, \$t7ka=384, \$t7la=385, \$t7ma=386, \$t7na=387, \$t7oa=388, \$t7pa=389, \$t7qa=390, \$t7ra=391, \$t7ua=392, \$t7va=393, \$t7wa=394, \$t7xa=395, \$t7ya=396, \$t7za=397, \$t7ba=398, \$t7ca=399, \$t7da=400, \$t7ea=401, \$t7fa=402, \$t7ga=403, \$t7ha=404, \$t7ia=405, \$t7ja=406, \$t7ka=407, \$t7la=408, \$t7ma=409, \$t7na=410, \$t7oa=411, \$t7pa=412, \$t7qa=413, \$t7ra=414, \$t7ua=415, \$t7va=416, \$t7wa=417, \$t7xa=418, \$t7ya=419, \$t7za=420, \$t7ba=421, \$t7ca=422, \$t7da=423, \$t7ea=424, \$t7fa=425, \$t7ga=426, \$t7ha=427, \$t7ia=428, \$t7ja=429, \$t7ka=430, \$t7la=431, \$t7ma=432, \$t7na=433, \$t7oa=434, \$t7pa=435, \$t7qa=436, \$t7ra=437, \$t7ua=438, \$t7va=439, \$t7wa=440, \$t7xa=441, \$t7ya=442, \$t7za=443, \$t7ba=444, \$t7ca=445, \$t7da=446, \$t7ea=447, \$t7fa=448, \$t7ga=449, \$t7ha=450, \$t7ia=451, \$t7ja=452, \$t7ka=453, \$t7la=454, \$t7ma=455, \$t7na=456, \$t7oa=457, \$t7pa=458, \$t7qa=459, \$t7ra=460, \$t7ua=461, \$t7va=462, \$t7wa=463, \$t7xa=464, \$t7ya=465, \$t7za=466, \$t7ba=467, \$t7ca=468, \$t7da=469, \$t7ea=470, \$t7fa=471, \$t7ga=472, \$t7ha=473, \$t7ia=474, \$t7ja=475, \$t7ka=476, \$t7la=477, \$t7ma=478, \$t7na=479, \$t7oa=480, \$t7pa=481, \$t7qa=482, \$t7ra=483, \$t7ua=484, \$t7va=485, \$t7wa=486, \$t7xa=487, \$t7ya=488, \$t7za=489, \$t7ba=490, \$t7ca=491, \$t7da=492, \$t7ea=493, \$t7fa=494, \$t7ga=495, \$t7ha=496, \$t7ia=497, \$t7ja=498, \$t7ka=499, \$t7la=500, \$t7ma=501, \$t7na=502, \$t7oa=503, \$t7pa=504, \$t7qa=505, \$t7ra=506, \$t7ua=507, \$t7va=508, \$t7wa=509, \$t7xa=510, \$t7ya=511, \$t7za=512, \$t7ba=513, \$t7ca=514, \$t7da=515, \$t7ea=516, \$t7fa=517, \$t7ga=518, \$t7ha=519, \$t7ia=520, \$t7ja=521, \$t7ka=522, \$t7la=523, \$t7ma=524, \$t7na=525, \$t7oa=526, \$t7pa=527, \$t7qa=528, \$t7ra=529, \$t7ua=530, \$t7va=531, \$t7wa=532, \$t7xa=533, \$t7ya=534, \$t7za=535, \$t7ba=536, \$t7ca=537, \$t7da=538, \$t7ea=539, \$t7fa=540, \$t7ga=541, \$t7ha=542, \$t7ia=543, \$t7ja=544, \$t7ka=545, \$t7la=546, \$t7ma=547, \$t7na=548, \$t7oa=549, \$t7pa=550, \$t7qa=551, \$t7ra=552, \$t7ua=553, \$t7va=554, \$t7wa=555, \$t7xa=556, \$t7ya=557, \$t7za=558, \$t7ba=559, \$t7ca=560, \$t7da=561, \$t7ea=562, \$t7fa=563, \$t7ga=564, \$t7ha=565, \$t7ia=566, \$t7ja=567, \$t7ka=568, \$t7la=569, \$t7ma=570, \$t7na=571, \$t7oa=572, \$t7pa=573, \$t7qa=574, \$t7ra=575, \$t7ua=576, \$t7va=577, \$t7wa=578, \$t7xa=579, \$t7ya=580, \$t7za=581, \$t7ba=582, \$t7ca=583, \$t7da=584, \$t7ea=585, \$t7fa=586, \$t7ga=587, \$t7ha=588, \$t7ia=589, \$t7ja=590, \$t7ka=591, \$t7la=592, \$t7ma=593, \$t7na=594, \$t7oa=595, \$t7pa=596, \$t7qa=597, \$t7ra=598, \$t7ua=599, \$t7va=600, \$t7wa=601, \$t7xa=602, \$t7ya=603, \$t7za=604, \$t7ba=605, \$t7ca=606, \$t7da=607, \$t7ea=608, \$t7fa=609, \$t7ga=610, \$t7ha=611, \$t7ia=612, \$t7ja=613, \$t7ka=614, \$t7la=615, \$t7ma=616, \$t7na=617, \$t7oa=618, \$t7pa=619, \$t7qa=620, \$t7ra=621, \$t7ua=622, \$t7va=623, \$t7wa=624, \$t7xa=625, \$t7ya=626, \$t7za=627, \$t7ba=628, \$t7ca=629, \$t7da=630, \$t7ea=631, \$t7fa=632, \$t7ga=633, \$t7ha=634, \$t7ia=635, \$t7ja=636, \$t7ka=637, \$t7la=638, \$t7ma=639, \$t7na=640, \$t7oa=641, \$t7pa=642, \$t7qa=643, \$t7ra=644, \$t7ua=645, \$t7va=646, \$t7wa=647, \$t7xa=648, \$t7ya=649, \$t7za=650, \$t7ba=651, \$t7ca=652, \$t7da=653, \$t7ea=654, \$t7fa=655, \$t7ga=656, \$t7ha=657, \$t7ia=658, \$t7ja=659, \$t7ka=660, \$t7la=661, \$t7ma=662, \$t7na=663, \$t7oa=664, \$t7pa=665, \$t7qa=666, \$t7ra=667, \$t7ua=668, \$t7va=669, \$t7wa=670, \$t7xa=671, \$t7ya=672, \$t7za=673, \$t7ba=674, \$t7ca=675, \$t7da=676, \$t7ea=677, \$t7fa=678, \$t7ga=679, \$t7ha=680, \$t7ia=681, \$t7ja=682, \$t7ka=683, \$t7la=684, \$t7ma=685, \$t7na=686, \$t7oa=687, \$t7pa=688, \$t7qa=689, \$t7ra=690, \$t7ua=691, \$t7va=692, \$t7wa=693, \$t7xa=694, \$t7ya=695, \$t7za=696, \$t7ba=697, \$t7ca=698, \$t7da=699, \$t7ea=700, \$t7fa=701, \$t7ga=702, \$t7ha=703, \$t7ia=704, \$t7ja=705, \$t7ka=706, \$t7la=707, \$t7ma=708, \$t7na=709, \$t7oa=710, \$t7pa=711, \$t7qa=712, \$t7ra=713, \$t7ua=714, \$t7va=715, \$t7wa=716, \$t7xa=717, \$t7ya=718, \$t7za=719, \$t7ba=720, \$t7ca=721, \$t7da=722, \$t7ea=723, \$t7fa=724, \$t7ga=725, \$t7ha=726, \$t7ia=727, \$t7ja=728, \$t7ka=729, \$t7la=730, \$t7ma=731, \$t7na=732, \$t7oa=733, \$t7pa=734, \$t7qa=735, \$t7ra=736, \$t7ua=737, \$t7va=738, \$t7wa=739, \$t7xa=740, \$t7ya=741, \$t7za=742, \$t7ba=743, \$t7ca=744, \$t7da=745, \$t7ea=746, \$t7fa=747, \$t7ga=748, \$t7ha=749, \$t7ia=750, \$t7ja=751, \$t7ka=752, \$t7la=753, \$t7ma=754, \$t7na=755, \$t7oa=756, \$t7pa=757, \$t7qa=758, \$t7ra=759, \$t7ua=760, \$t7va=761, \$t7wa=762, \$t7xa=763, \$t7ya=764, \$t7za=765, \$t7ba=766, \$t7ca=767, \$t7da=768, \$t7ea=769, \$t7fa=770, \$t7ga=771, \$t7ha=772, \$t7ia=773, \$t7ja=774, \$t7ka=775, \$t7la=776, \$t7ma=777, \$t7na=778, \$t7oa=779, \$t7pa=780, \$t7qa=781, \$t7ra=782, \$t7ua=783, \$t7va=784, \$t7wa=785, \$t7xa=786, \$t7ya=787, \$t7za=788, \$t7ba=789, \$t7ca=790, \$t7da=791, \$t7ea=792, \$t7fa=793, \$t7ga=794, \$t7ha=795, \$t7ia=796, \$t7ja=797, \$t7ka=798, \$t7la=799, \$t7ma=799, \$t7na=800, \$t7oa=801, \$t7pa=802, \$t7qa=803, \$t7ra=804, \$t7ua=805, \$t7va=806, \$t7wa=807, \$t7xa=808, \$t7ya=809, \$t7za=810, \$t7ba=811, \$t7ca=812, \$t7da=813, \$t7ea=814, \$t7fa=815, \$t7ga=816, \$t7ha=817, \$t7ia=818, \$t7ja=819, \$t7ka=820, \$t7la=821, \$t7ma=822, \$t7na=823, \$t7oa=824, \$t7pa=825, \$t7qa=826, \$t7ra=827, \$t7ua=828, \$t7va=829, \$t7wa=830, \$t7xa=831, \$t7ya=832, \$t7za=833, \$t7ba=834, \$t7ca=835, \$t7da=836, \$t7ea=837, \$t7fa=838, \$t7ga=839, \$t7ha=840, \$t7ia=841, \$t7ja=842, \$t7ka=843, \$t7la=844, \$t7ma=845, \$t7na=846, \$t7oa=847, \$t7pa=848, \$t7qa=849, \$t7ra=850, \$t7ua=851, \$t7va=852, \$t7wa=853, \$t7xa=854, \$t7ya=855, \$t7za=856, \$t7ba=857, \$t7ca=858, \$t7da=859, \$t7ea=860, \$t7fa=861, \$t7ga=862, \$t7ha=863, \$t7ia=864, \$t7ja=865, \$t7ka=866, \$t7la=867, \$t7ma=868, \$t7na=869, \$t7oa=870, \$t7pa=871, \$t7qa=872, \$t7ra=873, \$t7ua=874, \$t7va=875, \$t7wa=876, \$t7xa=877, \$t7ya=878, \$t7za=879, \$t7ba=880, \$t7ca=881, \$t7da=882, \$t7ea=883, \$t7fa=884, \$t7ga=885, \$t7ha=886, \$t7ia=887, \$t7ja=888, \$t7ka=889, \$t7la=890, \$t7ma=891, \$t7na=892, \$t7oa=893, \$t7pa=894, \$t7qa=895, \$t7ra=896, \$t7ua=897, \$t7va=898, \$t7wa=899, \$t7xa=900, \$t7ya=901, \$t7za=902, \$t7ba=903, \$t7ca=904, \$t7da=905, \$t7ea=906, \$t7fa=907, \$t7ga=908, \$t7ha=909, \$t7ia=910, \$t7ja=911, \$t7ka=912, \$t7la=913, \$t7ma=914, \$t7na=915, \$t7oa=916, \$t7pa=917, \$t7qa=918, \$t7ra=919, \$t7ua=920, \$t7va=921, \$t7wa=922, \$t7xa=923, \$t7ya=924, \$t7za=925, \$t7ba=926, \$t7ca=927, \$t7da=928, \$t7ea=929, \$t7fa=930, \$t7ga=931, \$t7ha=932, \$t7ia=933, \$t7ja=934, \$t7ka=935, \$t7la=936, \$t7ma=937, \$t7na=938, \$t7oa=939, \$t7pa=940, \$t7qa=941, \$t7ra=942, \$t7ua=943, \$t7va=944, \$t7wa=945, \$t7xa=946, \$t7ya=947, \$t7za=948, \$t7ba=949, \$t7ca=950, \$t7da=951, \$t7ea=952, \$t7fa=953, \$t7ga=954, \$t7ha=955, \$t7ia=956, \$t7ja=957, \$t7ka=958, \$t7la=959, \$t7ma=960, \$t7na=961, \$t7oa=962, \$t7pa=963, \$t7qa=964, \$t7ra=965, \$t7ua=966, \$t7va=967, \$t7wa=968, \$t7xa=969, \$t7ya=970, \$t7za=971, \$t7ba=972, \$t7ca=973, \$t7da=974, \$t7ea=975, \$t7fa=976, \$t7ga=977, \$t7ha=978, \$t7ia=979, \$t7ja=980, \$t7ka=981, \$t7la=982, \$t7ma=983, \$t7na=984, \$t7oa=985, \$t7pa=986, \$t7qa=987, \$t7ra=988, \$t7ua=989, \$t7va=990, \$t7wa=991, \$t7xa=992, \$t7ya=993, \$t7za=994, \$t7ba=995, \$t7ca=996, \$t7da=997, \$t7ea=998, \$t7fa=999, \$t7ga=999, \$t7ha=999, \$t7ia=999, \$t7ja=999, \$t7ka=999, \$t7la=999, \$t7ma=999, \$t7na=999, \$t7oa=999, \$t7pa=999, \$t7qa=999, \$t7ra=999, \$t7ua=999, \$t7va=999, \$t7wa=999, \$t7xa=999, \$t7ya=999, \$t7za=999, \$t7ba=999, \$t7ca=999, \$t7da=999, \$t7ea=999, \$t7fa=999, \$t7ga=999, \$t7ha=999, \$t7ia=999, \$t7ja=999, \$t7ka=999, \$t7la=999, \$t7ma=999, \$t7na=999, \$t7oa=999, \$t7pa=999, \$t7qa=999, \$t7ra=999, \$t7ua=999, \$t7va=999, \$t7wa=999, \$t7xa=999, \$t7ya=999, \$t7za=999, \$t7ba=999, \$t7ca=999, \$t7da=999, \$t7ea=999, \$t7fa=999, \$t7ga=999, \$t7ha=999, \$t7ia=999, \$t7ja=999, \$t7ka=999, \$t7la=999, \$t7ma=999, \$t7na=999, \$t7oa=999, \$t7pa=999, \$t7qa=999, \$t7ra=999, \$t7ua=999, \$t7va=999, \$t7wa=999, \$t7xa=999, \$t7ya=999, \$t7za=999, \$t7ba=999, \$t7ca=999, \$t7da=999, \$t7ea=999, \$t7fa=999, \$t7ga=999, \$t7ha=999, \$t7ia=999, \$t7ja=999, \$t7ka=999, \$t7la=999, \$t7ma=999, \$t7na=999, \$t7oa=999, \$t7pa=999, \$t7qa=999, \$t7ra=999, \$t7ua=999, \$t7va=999, \$t7wa=999, \$t7xa=999, \$t7ya=999, \$t7za=999, \$t7ba=999, \$t7ca=999, \$t7da=999, \$t7ea=999, \$t7fa=999, \$t7ga=999, \$t7ha=999, \$t7ia=999, \$t7ja=999, \$t7ka=999, \$t7la=999, \$t7ma=999, \$t7na=999, \$t7oa=999, \$t7pa=999, \$t7qa=999, \$t7ra=999, \$t7ua=999, \$t7va=999, \$t7wa=999, \$t7xa=999, \$t7ya=999, \$t7za=999, \$t7ba=999, \$t7ca=999, \$t7da=999, \$t7ea=999, \$t7fa=999, \$t7ga=999, \$t7ha=999, \$t7ia=999, \$t7ja=999, \$t7ka=999, \$t7la=999, \$t7ma=999, \$t7na=999, \$t7oa=999, \$t7pa=999, \$t7qa=999, \$t7ra=999, \$t7ua=999, \$t7va=999, \$t7wa=999, \$t7xa=999, \$t7ya=999, \$t7za=999, \$t7ba=999, \$t7ca=999, \$t7da=999, \$t7ea=999, \$t7fa=999, \$t7ga=999, \$t7ha=999, \$t7ia=999, \$t7ja=999, \$t7ka=999, \$t7la=999, \$t7ma=999, \$t7na=999, \$t7oa=999, \$t7pa=999, \$t7qa=999, \$t7ra=999, \$t7ua=999, \$t7va=999, \$t7wa=999, \$t7xa=999, \$t7ya=999, \$t7za=999, \$t7ba=999, \$t7ca=999, \$t7da=999, \$t7ea=999, \$t7fa=999, \$t7ga=999, \$t7ha=999, \$t7ia=999, \$t7ja=999, \$t7ka=999, \$t7la=999, \$t7ma=999, \$t7na=999, \$t7oa=999, \$t7pa=999, \$t7qa=999, \$t7ra=999, \$t7ua=999, \$t7va=999, \$t7wa=999, \$t7xa=999, \$t7ya=999, \$t7za=999, \$t7ba=999, \$t7ca=999, \$t7da=999, \$t7ea=999, \$t7fa=999, \$t7ga=999, \$t7ha=999, \$t7ia=999, \$t7ja=999, \$t7ka=999, \$t7la=999, \$t7ma=999, \$t7na=999, \$t7oa=999, \$t7pa=999, \$t7qa=999, \$t7ra=999, \$t7ua=999, \$t7va=999, \$t7wa=999, \$t7xa=999, \$t7ya=999, \$t7za=999, \$t7ba=999, \$t7ca=999, \$t7da=999, \$t7ea=999, \$t7fa=999, \$t7ga=999, \$t7ha=999, \$t7ia=999, \$t7ja=999, \$t7ka=999, \$t7la=999, \$t7ma=999, \$t7na=999, \$t7oa=999, \$t7pa=999, \$t7qa=999, \$t7ra=999, \$t7ua=999, \$t7va=999, \$t7wa=999, \$t7xa=999, \$t7ya=999, \$t7za=999, \$t7ba=999, \$t7ca=999, \$t7da=999, \$t7ea=999, \$t7fa=999, \$t7ga=999, \$t7ha=999, \$t7ia=999, \$t7ja=999, \$t7ka=999, \$t7la=999, \$t7ma=999, \$t7na=999, \$t7oa=999, \$t7pa=999, \$t7qa=999, \$t7ra=999, \$t7ua=999, \$t7va=999, \$t7wa=999, \$t7xa=999, \$t7ya=999, \$t7za=999, \$t7ba=999, \$t7ca=999, \$t7da=999, \$t7ea=999, \$t7fa=999, \$t7ga=999, \$t7ha=999, \$t7ia=999, \$t7ja=999, \$t7ka=999, \$t7la=999, \$t7ma=999, \$t7na=999, \$t7oa=999, \$t7pa=999, \$t7qa=999, \$t7ra=999, \$t7ua=999, \$t7va=999, \$t7wa=999, \$t7xa=999, \$t7ya=999, \$t7za=999, \$t7ba=999, \$t7ca=999, \$t7da=999, \$t7

### 3. Assignment 3:

```

HomeAssignment3.asm
1 # Laboratory Exercise 3, Home Assignment 3
2 # Author: Pham Huy Canh
3 .data
4     test: .word 1
5 .text
6     la $s0, test      # load the address of test variable
7     lw $s1, 0($s0)    # load the value of test to register $t1
8     li $t0, 0          # load value for test case
9     li $t1, 1
10    li $t2, 2
11    li $s2, 2
12    li $s3, 1
13    beq $s1, $t0, case_0
14    beq $s1, $t1, case_1
15    beq $s1, $t2, case_2
16    j default
17 case_0:
18     addi $s2, $s2, 1      # a=a+1
19     j continue
20 case_1:
21     sub $s2, $s2, $t1      # a=a-1
22     j continue
23 case_2:
24     add $s3, $s3, $s3      # b=2*b
25     j continue
26 default:
27 continue:

```

Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	1
\$t2	10	2
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	268500992
\$s1	17	1
\$s2	18	1
\$s3	19	1
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268440824
\$sp	29	2147419548
\$fp	30	0
\$ra	31	0
pc		4194376
hi		0
lo		0

Dữ liệu của *test* được nạp vào trong *\$s1*, ta so sánh *\$s1* với *\$t0*, *\$t1*, *\$t2*

*\$s1* = *\$t0* => nhảy tới *case\_0*

*\$s1* = *\$t1* => nhảy tới *case\_1*

*\$s1* = *\$t2* => nhảy tới *case\_2*

Khi cả ba đều không bằng *\$s1* khi chạy đến lệnh *j default* thì lệnh sẽ nhảy đến nhãn *default*

Trường hợp này *\$s1* = *\$t1* = 1 nên sẽ thực hiện lệnh trong *case\_1*

=> Kết quả thu được: *\$s2* = *\$s2* - 1 = 2 - 1 = 1

### 4. Assignment 4:

a.  $i < j$

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	268500592
\$v0	2	0
\$v1	3	0
\$a0	4	268501000
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	1
\$t2	10	0
\$t3	11	2
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$a0	16	0
\$a1	17	5
\$a2	18	4
\$a3	19	5
\$a4	20	3
\$a5	21	0
\$a6	22	0
\$a7	23	0
\$t0	24	0
\$k0	25	0
\$k1	26	0
\$gp	27	0
\$sp	28	268468224
\$fp	29	2147479548
\$ra	30	0
pc	31	0
hi		4194384
lo		0

```

Edit Execute
HomeAssignment1.asm
2 # Author: Pham Huy Canh
3 .data
4     x: .word 1
5     y: .word 1
6     z: .word 1
7 .text
8     addi $s1, $zero, 5      # Khoi tao i = 5
9     addi $s2, $zero, 4      # Khoi tao j = 4
10    addi $s3, $zero, 5      # Khoi tao j = 4
11    addi $s4, $zero, 3      # Khoi tao j = 4
12    la $a0, x              # Lay dia chi cua x
13    lw $t1, 0($a0)          # Load gia tri cua x vao thanh ghi $t1
14    la $a0, y              # Lay dia chi cua y
15    lw $t2, 0($a0)          # Load gia tri cua y vao thanh ghi $t2
16    la $a0, z              # Lay dia chi cua z
17    lw $t3, 0($a0)          # Load gia tri cua z vao thanh ghi $t3
18
19
20 start:
21     slt $t4, $s1, $s2      # i<j
22     beq $t4, $zero, else    # branch to else if i>=j
23     addi $t1,$t1,1          # then part: x=x+1
24     addi $t3,$zero,1          # z=1
25     j endif                # skip "else" part
26 else:
27     addi $t2,$t2,-1          # begin else part: y=y-1
28     add $t3,$t3,$t3          # z=z*x
29 endif:
30

```

b.  $i \geq j$

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	268500592
\$v0	2	0
\$v1	3	0
\$a0	4	268501000
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	1
\$t2	10	1
\$t3	11	1
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$a0	16	0
\$a1	17	5
\$a2	18	4
\$a3	19	5
\$a4	20	3
\$a5	21	0
\$a6	22	0
\$t7	23	0
\$t0	24	0
\$k0	25	0
\$k1	26	0
\$gp	27	0
\$sp	28	268468224
\$fp	29	2147479548
\$ra	30	0
pc	31	0
hi		4194384
lo		0

```

Edit Execute
HomeAssignment1.asm
1 # Laboratory Exercise 3, Home Assignment 1
2 # Author: Pham Huy Canh
3 .data
4     x: .word 1
5     y: .word 1
6     z: .word 1
7 .text
8     addi $s1, $zero, 5      # Khoi tao i = 5
9     addi $s2, $zero, 4      # Khoi tao j = 4
10    addi $s3, $zero, 5      # Khoi tao j = 4
11    addi $s4, $zero, 3      # Khoi tao j = 4
12    la $a0, x              # Lay dia chi cua x
13    lw $t1, 0($a0)          # Load gia tri cua x vao thanh ghi $t1
14    la $a0, y              # Lay dia chi cua y
15    lw $t2, 0($a0)          # Load gia tri cua y vao thanh ghi $t2
16    la $a0, z              # Lay dia chi cua z
17    lw $t3, 0($a0)          # Load gia tri cua z vao thanh ghi $t3
18
19
20 start:
21     slt $t4, $s1, $s2      # i<j
22     bne $t4, $zero, else    # branch to else if i<j
23     addi $t1,$t1,1          # then part: x=x+1
24     addi $t3,$zero,1          # z=1
25     j endif                # skip "else" part
26 else:
27     addi $t2,$t2,-1          # begin else part: y=y-1
28     add $t3,$t3,$t3          # z=z*x
29 endif:
30

```

c.  $i+j \leq 0$

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	0
\$v1	3	0
\$a0	4	268501000
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	1
\$t2	10	0
\$t3	11	2
\$t4	12	0
\$t5	13	1
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	5
\$s2	18	4
\$s3	19	5
\$s4	20	3
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	26846824
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194392
hi		0
lo		0

```

Edit Execute
HomeAssignment1.asm
3 .data
4     x: .Word 1
5     y: .Word 1
6     z: .Word 1
7 .text
8     addi $s1, $zero, 5      # Khoi tao i = 5
9     addi $s2, $zero, 4      # Khoi tao j = 4
10    addi $s3, $zero, 5      # Khoi tao j = 4
11    addi $s4, $zero, 3      # Khoi tao j = 4
12    la $a0, x              # Lay dia chi cua x
13    lw $t1, 0($a0)          # Load gia tri cua x vao thanh ghi $t1
14    la $a0, y              # Lay dia chi cua y
15    lw $t2, 0($a0)          # Load gia tri cua y vao thanh ghi $t2
16    la $a0, z              # Lay dia chi cua z
17    lw $t3, 0($a0)          # Load gia tri cua z vao thanh ghi $t3
18
19
20 start:
21     add $t4, $s1, $s2      # t1=i+j
22     sgt $t5, $t4, 0         # t1 > 0
23     bne $t5, $zero, else   # t2 != 0 branch to else (i+j > 0)
24     addi $t1,$t1,1          # then part: xx+x1
25     addi $t3,$zero,1          # z=1
26     j endif                # skip "else" part
27 else:
28     addi $t2,$t2,-1        # begin else part: y=y-1
29     add $t3,$t3,$t3          # z=z*x
30 endif:
31

```

d.  $i+j > m+n$

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	268500952
\$v0	2	0
\$v1	3	0
\$a0	4	268501000
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	2
\$t2	10	1
\$t3	11	1
\$t4	12	0
\$t5	13	0
\$t6	14	1
\$t7	15	0
\$s0	16	0
\$s1	17	5
\$s2	18	4
\$s3	19	5
\$s4	20	3
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	26846824
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194392
hi		0
lo		0

```

Edit Execute
HomeAssignment1.asm
2 # Author: Pham Huy Canh
3 .data
4     x: .Word 1
5     y: .Word 1
6     z: .Word 1
7 .text
8     addi $s1, $zero, 5      # Khoi tao i = 5
9     addi $s2, $zero, 4      # Khoi tao j = 4
10    addi $s3, $zero, 5      # Khoi tao j = 4
11    addi $s4, $zero, 3      # Khoi tao j = 4
12    la $a0, x              # Lay dia chi cua x
13    lw $t1, 0($a0)          # Load gia tri cua x vao thanh ghi $t1
14    la $a0, y              # Lay dia chi cua y
15    lw $t2, 0($a0)          # Load gia tri cua y vao thanh ghi $t2
16    la $a0, z              # Lay dia chi cua z
17    lw $t3, 0($a0)          # Load gia tri cua z vao thanh ghi $t3
18
19
20 start:
21     add $t4, $s1, $s2      # t4=i+j
22     add $t5, $s6, $s7      # t5=m+n
23     slt $t6, $t5, $t4      # t5<t4
24     beq $t6, $zero, else   # m+n >= i+j branch to else
25     addi $t1,$t1,1          # then part: xx+x1
26     addi $t3,$zero,1          # z=1
27     j endif                # skip "else" part
28 else:
29     addi $t2,$t2,-1        # begin else part: y=y-1
30     add $t3,$t3,$t3          # z=z*x
31 endif:
32

```

## 5. Assignment 5:

a.  $i < n$

The screenshot shows a debugger interface with two main panes. The left pane displays the assembly code for 'HomeAssignment2.asm' with comments explaining the purpose of each instruction. The right pane shows the register state at the end of the execution.

```

Edit Execute
HomeAssignment2.asm
1 # Laboratory 3, Home Assignment 2
2 # Author: Pham Huy Canh
3
4 .data
5 A: .word 2,3,4,5,-6
6
7 .text
8 addi $s1, $zero, 0      # i = 0
9 addi $s3, $zero, 5      # n = 5
10 addi $s4, $zero, 1      # step = 1
11 addi $s5, $zero, 0      # sum = 0
12 la $s2, A              # load address A[0] to $s2
13 loop:
14 slt $t2, $s1, $s3      # $t2 = i < n ? 1 : 0
15 bneq $t2, $zero, endloop
16 add $t1, $s1, $s1      # $t1 = 2 * $s1
17 add $t1, $t1, $s1      # $t1 = 4 * $s1
18 add $t1, $t1, $s2      # $t1 store the address of A[i]
19 lw $t0, 0($t1)          # load value of A[i] in $t0
20 add $s5, $s5, $t0      # sum = sum + A[i]
21 add $s1, $s1, $s4      # i = i + step
22 j loop                # goto loop
23 endloop:
24
25
26
27
28
29
30

```

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	268500592
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	268501008
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$a0	16	0
\$a1	17	5
\$a2	18	268500592
\$a3	19	5
\$a4	20	1
\$a5	21	8
\$a6	22	0
\$a7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479540
\$fp	30	0
\$ra	31	0
pc		4194364
hi		0
lo		0

Kết quả Sum = \$s5 = 8

b.  $i \leq n$

The screenshot shows a debugger interface with two main panes. The left pane displays the assembly code for 'HomeAssignment2.asm' with comments explaining the purpose of each instruction. The right pane shows the register state at the end of the execution.

```

Edit Execute
HomeAssignment2.asm
1 # Laboratory 3, Home Assignment 2
2 # Author: Pham Huy Canh
3
4 .data
5 A: .word 2,3,4,5,-6
6
7 .text
8 addi $s1, $zero, 0      # i = 0
9 addi $s3, $zero, 5      # n = 5
10 addi $s4, $zero, 1      # step = 1
11 addi $s5, $zero, 0      # sum = 0
12 la $s2, A              # load address A[0] to $s2
13 loop:
14 slt $t2, $s1, $s3      # $t2 = i < n ? 1 : 0
15 bneq $t2, $zero, endloop
16 add $t1, $s1, $s1      # $t1 = 2 * $s1
17 add $t1, $t1, $s1      # $t1 = 4 * $s1
18 add $t1, $t1, $s2      # $t1 store the address of A[i]
19 lw $t0, 0($t1)          # load value of A[i] in $t0
20 add $s5, $s5, $t0      # sum = sum + A[i]
21 add $s1, $s1, $s4      # i = i + step
22 j loop                # goto loop
23 endloop:
24
25
26
27
28
29
30

```

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	268500592
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	268501012
\$t2	10	1
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$a0	16	0
\$a1	17	6
\$a2	18	268500592
\$a3	19	5
\$a4	20	1
\$a5	21	8
\$a6	22	0
\$a7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479540
\$fp	30	0
\$ra	31	0
pc		4194364
hi		0
lo		0

Kết quả Sum = \$s5 = 8

c.  $\sum \geq 0$

The screenshot shows the assembly code for "HomeAssignment2.asm" and the state of the processor registers.

**Registers:**

Name	Number	Value
\$zero	0	0
\$at	1	268500952
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	268500952
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	268500952
\$s3	19	5
\$s4	20	1
\$s5	21	2
\$s6	22	0
\$s7	23	0
\$s8	24	0
\$s9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479540
\$fp	30	0
\$ra	31	0
pc		4194364
hi		0
lo		0

Kết quả Sum = \$s5 = 2

d. A[i] == 0

The screenshot shows the assembly code for "HomeAssignment2.asm" and the state of the processor registers.

**Registers:**

Name	Number	Value
\$zero	0	0
\$at	1	0
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	5
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$s8	24	0
\$s9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479540
\$fp	30	0
\$ra	31	0
pc		4194364
hi		0
lo		0

Kết quả Sum = \$s5 = 5

## 6. Assignment 6:

Registers    Coproc 1    Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	0
\$v1	3	0
\$a0	4	268500992
\$a1	5	268501016
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	24
\$t2	10	0
\$t3	11	0
\$t4	12	1
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	35
\$s1	17	1
\$s2	18	7
\$s3	19	7
\$s4	20	35
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$fp	28	268440224
\$sp	29	2147479540
\$fp	30	0
\$ra	31	0
pc		4194388
hi		0
lo		0

```

Edit Execute
HomeAssignment1.asm Assignment6.asm
1 # Laboratory Exercise 3, Assignment 6
2 # Author: Pham Huy Canh
3 .data
4     A: .word 1, -3, -10, 6, -29, 3, -39
5     message: .asciz "Tri tuyet doi lon nhat la: "
6 .text
7     addi $s0, $zero, 0      # max = 0
8     la $a0, A
9     lw $s1, 0($a0)          # A[0]
10    addi $s2, $zero, 0       # i = 0
11    addi $s3, $zero, 7       # n = 7
12
13    loop: slt $t2, $s2, $s3   # i < n
14    beq $t2, $zero, endloop # i >= n branch to endloop
15    sll $t1, $s2, 2          # t1 = i * 4
16    add $a1, $a0, $t1          # a1 = a0 + 4
17    lw $s4, 0($a1)          # s4 = A[i]
18    if_nhohon:
19        bgez $s4, if_lonhon # s4 > 0 branch to if_lonhon
20        sub $s4, $zero, $s4      # s4 = 0 - s4
21        j if_lonhon
22    if_lonhon:
23        slt $t4, $s0, $s4      # max < s4
24        bne $t4, $zero, max     # max < s4 branch to max
25        j reloop                # jump reloop
26    max: add $s0, $zero, $s4      # max = 0 + s4
27        j reloop                # jump reloop
28    reloop: addi $s2, $s2, 1      # i = i + 1
29        j loop                  # jump loop
30 endloop:

```