

BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH (IT3280) TUẦN 3

Họ và tên: **TRỊNH MINH ĐỨC**

MSSV: 20225813

Assignment 1:

Ta xét hai trường hợp $i = 3, j = 5$ và ngược lại.

- TH1: $i = 3$ và $j = 5$ ($i \leq j$):

```
.text
    addi    $s1, $zero, 3 # khai bao i
    addi    $s2, $zero, 5 # khai bao j
    addi    $t1, $zero, 5 # khai bao x
    addi    $t2, $zero, 5 # khai bao y
    addi    $t3, $zero, 5 # khai bao z

start:
    slt     $t0, $s2, $s1  # j < i
    bne     $t0, $zero, else # means if $t0 = 1 then branch to else
    addi    $t1, $t1, 1    # x=x+1
    addi    $t3, $zero, 1  # z=1
    j       endif         # skip "else" part
else:
    addi    $t2, $t2, -1   # begin else part: y=y-1
    add     $t3, $t3, $t3  # z=z*2
endif:
```

	\$s1	\$s2	\$t0	\$t1	\$t2	\$t3	pc
Ban đầu	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x40000000
Khai báo i	0x00000003	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x40000004
Khai báo j	0x00000003	0x00000005	0x00000000	0x00000000	0x00000000	0x00000000	0x40000008
Khai báo x, y, z	0x00000003	0x00000005	0x00000000	0x00000005	0x00000005	0x00000005	0x40000014

Chạy lệnh slt	0x00000003	0x00000005	0x00000000	0x00000005	0x00000005	0x00000005	0x40000018
Chạy lệnh bne	0x00000003	0x00000005	0x00000000	0x00000005	0x00000005	0x00000005	0x4000001c
Chạy lệnh addi thứ 1 của start	0x00000003	0x00000005	0x00000000	0x00000006	0x00000005	0x00000005	0x40000020
Chạy lệnh addi thứ 2 của start	0x00000003	0x00000005	0x00000000	0x00000006	0x00000005	0x00000001	0x40000024

➔ Việc kiểm thử là đúng khi ta thấy $j \geq i$ (với giá trị thanh ghi \$s1 là 3 (0x00000003) và giá trị thanh ghi \$s2 là 5 (0x00000005)) thì giá trị thanh ghi \$t0 sẽ trả về 0 (địa chỉ 0x00000000) và giá trị thanh ghi này đúng với thanh ghi \$zero nên trình biên dịch không thực hiện lệnh rẽ nhánh mà tiếp tục các lệnh addi, từ đó tính toán và gán giá trị cho thanh ghi \$t1 là 6 (0x00000006) và gán giá trị thanh ghi \$t3 là 1 (0x00000001).

- TH2: $i = 5$ và $j = 3$ ($i > j$):

```
.text
    addi    $s1, $zero, 5 # khai bao i
    addi    $s2, $zero, 3 # khai bao j
    addi    $t1, $zero, 5 # khai bao x
    addi    $t2, $zero, 5 # khai bao y
    addi    $t3, $zero, 5 # khai bao z

start:
    slt     $t0, $s2, $s1 # j < i
    bne     $t0, $zero, else # means if $t0 = 1 then branch to else
    addi    $t1, $t1, 1    # x=x+1
    addi    $t3, $zero, 1  # z=1
    j       endif         # skip "else" part
else:
    addi    $t2, $t2, -1   # begin else part: y=y-1
    add     $t3, $t3, $t3  # z=z*2
endif:
```

	\$s1	\$s2	\$t0	\$t1	\$t2	\$t3	pc
Ban đầu	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x40000000
Khai báo i, j	0x00000005	0x00000003	0x00000000	0x00000000	0x00000000	0x00000000	0x40000008
Khai báo x, y, z	0x00000005	0x00000003	0x00000000	0x00000005	0x00000005	0x00000005	0x40000014
Chạy lệnh slt	0x00000005	0x00000003	0x00000001	0x00000005	0x00000005	0x00000005	0x40000018
Chạy lệnh bne	0x00000005	0x00000003	0x00000001	0x00000005	0x00000005	0x00000005	0x40000028
Chạy lệnh addi thứ 1 của else	0x00000005	0x00000003	0x00000001	0x00000000	0x00000004	0x00000000	0x4000002c
Chạy lệnh add của else	0x00000005	0x00000003	0x00000001	0x00000000	0x00000004	0x0000000a	0x40000030

➔ Việc kiểm thử là đúng khi ta thấy $j < i$ (với giá trị thanh ghi \$s1 là 5 (0x00000005) và giá trị thanh ghi \$s2 là 3 (0x00000003)) thì giá trị thanh ghi \$t0 sẽ trả về 1 (địa chỉ 0x00000001) và giá trị thanh ghi này khác với thanh ghi \$zero nên trình biên dịch đã thực hiện lệnh rẽ nhánh, từ đó tính toán và gán giá trị cho thanh ghi \$t2 là 4 (địa chỉ là 0x00000004) và gán giá trị thanh ghi \$t3 bằng 2 lần giá trị chính nó là 10 (địa chỉ là 0x0000000a) .

Assignment 2:

.data

A: .word 1,2,3,4

.text

addi \$s1,\$zero,0 # i=0

la \$s2,A # địa chỉ biến A

```

addi $s3,$zero,4 # n=4
addi $s4,$zero,1 # step=1
add $s5,$zero,0 # sum=0
loop: slt $t2,$s1,$s3 # i<n
      beq $t2,$zero,endloop
      add $t1,$s1,$s1
      add $t1,$t1,$t1 #t1=4*s1
      add $t1,$t1,$s2 #t1 store the address of A[i]
      lw $t0,0($t1) #load value of A[i] in $t0
      add $s5,$s5,$t0 #sum=sum+A[i]
      add $s1,$s1,$s4
      j loop
endloop:

```

- Đầu tiên ta khởi tạo mảng có 4 phần tử có các giá trị là 1, 2, 3, 4
- Tiếp đến ta khởi tạo các giá trị i, n, step (bước nhảy), step = 4=> sẽ có 4 lần lặp => sau 4 vòng lặp ta sẽ thu được tổng các phần tử trong chuỗi chính là giá trị của thanh ghi \$s5.
- Sự thay đổi của các thanh ghi:

	\$t1	\$t2	\$s1	\$s5
Ban đầu	0x00000000	0x00000000	0x00000000	0x00000000
Vòng loop 1	0x10010000	0x00000001	0x00000001	0x00000001
Vòng loop 2	0x10010004	0x00000001	0x00000002	0x00000003
Vòng loop 3	0x10010008	0x00000001	0x00000003	0x00000006
Vòng loop 4	0x1001000c	0x00000001	0x00000004	0x0000000a
Vòng 'loop' 5	0x1001000c	0x00000000	0x00000004	0x0000000a

- Nhân xét:
 - Vòng lặp dừng lại khi \$t2 = 0x00000000, tương ứng với điều kiện câu lệnh: beq \$t2, \$zero, endloop
 - Kết quả giá trị sum = \$s5 = 0x0000000a = 10 = 1 + 2 + 3 + 4, điều này đúng với tính toán khi ta tính tổng các phần tử của mảng.
 - Thử với mảng gồm các phần tử là 2, 4, 6, 8 ta thu được \$s5 = 0x00000014 = 20, điều này đúng với tính toán khi ta tính tổng các phần tử của mảng.

Text Segment						Name			Number		Value	
Bkpt	Address	Code	Basic	Source								
	0x00400000	0x20150000	addi \$t1,\$0,0x00000000	4:	addi \$s5, \$zero, 0	# khoi tao sum					\$v0	0x00000000
	0x00400004	0x20110000	addi \$t7,\$0,0x00000000	5:	addi \$s1, \$zero, 0	# khoi tao i					\$v1	0x00000000
	0x00400008	0x20130004	addi \$t9,\$0,0x00000000	6:	addi \$s3, \$zero, 4	# khoi tao n					\$a0	0x00000000
	0x0040000c	0x20140001	addi \$t0,\$0,0x00000000	7:	addi \$s4, \$zero, 1	# khoi tao step					\$a1	0x00000000
	0x00400010	0x3c011001	lui \$t1,0x000001001	8:	la \$s2, array	# khoi tao array					\$a2	0x00000000
	0x00400014	0x34320000	ori \$t8,\$t1,0x000000000								\$a3	0x00000000
	0x00400018	0x0233502a	slt \$t0,\$t7,\$t9	9: loop:	slt \$t2, \$s1, \$s3	# i<n					\$t0	0x00000008
	0x0040001c	0x11400007	beq \$t0,\$0,0x000000007	10:	beq \$t2, \$zero, endloop						\$t1	0x1001000c
	0x00400020	0x02314820	add \$t9,\$t7,\$t7	11:	add \$t1, \$s1, \$s1						\$t2	0x00000000
											\$t3	0x00000000
											\$t4	0x00000000
											\$t5	0x00000000
											\$t6	0x00000000
											\$t7	0x00000000
											\$s0	0x00000000
											\$s1	0x00000000
											\$s2	0x00000004
											\$s3	0x10010000
											\$s4	0x00000004
											\$s5	0x00000000
											\$s6	0x00000001
											\$s7	0x00000014
											\$s8	0x00000000
											\$s9	0x00000000
											\$t8	0x00000000
											\$t9	0x00000000
											\$k0	0x00000000
											\$k1	0x00000000
											\$gp	0x10008000
											\$sp	0x7ffffeffc
											\$fp	0x00000000
											\$ra	0x00000000
											\$pc	0x0040003c
											\$hi	0x00000000
											\$lo	0x00000000

Assignment 3:

- TH1: test = 1:

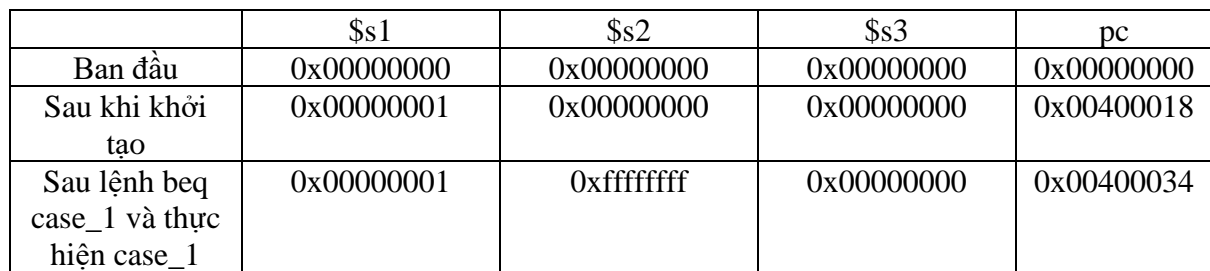
```

.data
    test: .word 1

.text
    la    $s0, test
    lw    $s1, 0($s0)    #load the value of test to register $t1
    li    $t0, 0         #load value for test case
    li    $t1, 1
    li    $t2, 2
    beq   $s1, $t0, case_0
    beq   $s1, $t1, case_1
    beq   $s1, $t2, case_2
    j     default
case_0:  addi $s2, $s2, 1    #a=a+1
    j     continue
case_1:  sub  $s2, $s2, $t1  #a=a-1
    j     continue
case_2:  add  $s3, $s3, $s3  #b=2*b
    j     continue

default:
continue:

```



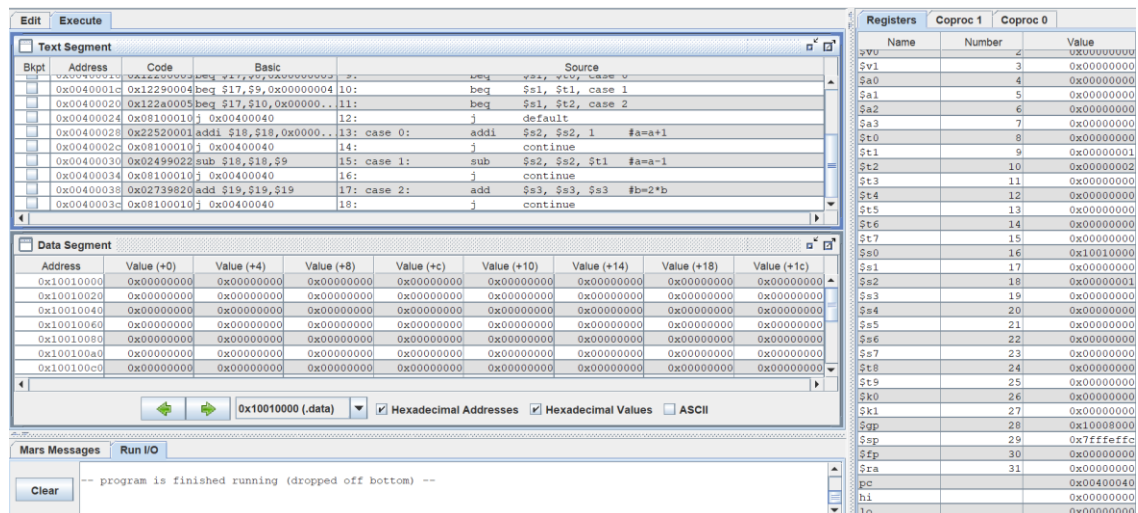
- TH2: test = 0:

```

.data
    test: .word 0

.text
    la      $s0, test
    lw      $s1, 0($s0)    #load the value of test to register $t1
    li      $t0, 0         #load value for test case
    li      $t1, 1
    li      $t2, 2
    beq     $s1, $t0, case_0
    beq     $s1, $t1, case_1
    beq     $s1, $t2, case_2
    j       default
case_0:
    addi    $s2, $s2, 1     #a=a+1
    j       continue
case_1:
    sub     $s2, $s2, $t1   #a=a-1
    j       continue
case_2:
    add     $s3, $s3, $s3   #b=2*b
    j       continue
default:
continue:

```



	\$s1	\$s2	\$s3	pc
Ban đầu	0x00000000	0x00000000	0x00000000	0x00000000
Sau khi khởi tạo	0x00000000	0x00000000	0x00000000	0x00400018
Sau lệnh beq case_0 và thực hiện case_0	0x00000000	0x00000001	0x00000000	0x0040002c

➔ Do $\$s1 = \$t0 = 0$ nên lệnh hàm sẽ thực hiện hàm case_0 và thanh ghi pc nhảy đến giá trị của thẻ case_0 là 0x00400028 thực hiện lệnh cộng thanh ghi \$s2 cho 1 ta được $\$s2 = 1$ (thỏa mãn).

- TH3: test = 2:

```
.data
    test: .word 2

.text
    la    $s0, test
    lw    $s1, 0($s0)    #load the value of test to register $t1
    li    $t0, 0          #load value for test case
    li    $t1, 1
    li    $t2, 2
    beq   $s1, $t0, case_0
    beq   $s1, $t1, case_1
    beq   $s1, $t2, case_2
    j     default
case_0:
    addi  $s2, $s2, 1     #a=a+1
    j     continue
case_1:
    sub   $s2, $s2, $t1   #a=a-1
    j     continue
case_2:
    add   $s3, $s3, $s3   #b=2*b
    j     continue

default:
continue:
```

The screenshot shows the Mars MIPS simulator interface. The Text Segment window displays the assembly code with addresses and comments. The Data Segment window shows the memory layout with registers \$s0 through \$t9. The Register File window on the right shows the current values of the registers. The console at the bottom shows the message 'program is finished running (dropped off bottom)'.

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	0x00000001
\$t2	10	0x00000002
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x10010000
\$s1	17	0x00000002
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$s8	24	0x00000000
\$s9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffffc00
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400040
hi		0x00000000
lo		0x00000000

	\$s1	\$s2	\$s3	pc
Ban đầu	0x00000000	0x00000000	0x00000000	0x00000000
Sau khi khởi tạo	0x00000002	0x00000000	0x00000000	0x00400018
Sau lệnh beq case_2 và thực hiện case_2	0x00000002	0x00000000	0x00000000	0x0040003c

→ Như vậy do $s1 = t2 = 2$ nên lệnh hàm sẽ thực hiện hàm case_2 và thanh ghi pc nhảy đến giá trị của thẻ case_0 là 0x00400038 thực hiện lệnh cộng thanh ghi $s3$ với thanh ghi $s3$ ta được $s3 = 0$ (thỏa mãn).

Assignment 4:

a) $i < j$:

- Đề bài trở thành:

if ($i < j$)

$x = x + 1$;

$z = 1$;

else

$y = y - 1$;

$z = 2 * z$;

- Code:

.text

addi $s1, \$zero, 3$ # khai bao i

addi $s2, \$zero, 5$ # khai bao j

addi $t1, \$zero, 5$ # khai bao x

addi $t2, \$zero, 5$ # khai bao y

addi $t3, \$zero, 5$ # khai bao z

la $t4, m$

lw $s4, 0(t4)$ # khai bao m

la $t5, n$

lw $s5, 0(t5)$ # khai bao n

start:

slt \$t0, \$s1, \$s2 # i<j a)

beq \$t0, \$zero, else

addi \$t1, \$t1, 1 # x=x+1

addi \$t3, \$zero, 1 # z=1

j endif

else: addi \$t2, \$t2, -1 # y=y-1

add \$t3, \$t3, \$t3 # z=z*2

endif:

```
addi $s1, $zero, 3 # khai bao i
addi $s2, $zero, 5 # khai bao j
addi $t1, $zero, 5 # khai bao x
addi $t2, $zero, 5 # khai bao y
addi $t3, $zero, 5 # khai bao z
la $t4, m
lw $s4, 0($t4) # khai bao m
la $t5, n
lw $s5, 0($t5) # khai bao n
start:
slt $t0, $s1, $s2 # i<j a)
#ge $t0, $s1, $s2 # i>=j b)
#add $s3, $s1, $s2 # khai bao i+j c+d)
#sle $t0, $s3, $zero # i+j<=0 c)
#add $s6, $s4, $s5 # khai bao m+n d)
#sgt $t0, $s3, $s6 # i+j>m+n d)

#slt $t0, $s2, $s1 # j<i
beq $t0, $zero, else
addi $t1, $t1, 1 # x=x+1
addi $t3, $zero, 1 # z=1
j endif
else: addi $t2, $t2, -1 # y=y-1
add $t3, $t3, $t3 # z=z*2
endif:
```



```

    addi    $t1, $zero, 5    # khai bao x
    addi    $t2, $zero, 5    # khai bao y
    addi    $t3, $zero, 5    # khai bao z
    la      $t4, m
    lw      $s4, 0($t4)      # khai bao m
    la      $t5, n
    lw      $s5, 0($t5)      # khai bao n
start:
    sge     $t0, $s1, $s2    # i>=j b)
    beq     $t0, $zero, else
    addi    $t1, $t1, 1      # x=x+1
    addi    $t3, $zero, 1    # z=1
    j       endif
else:
    addi    $t2, $t2, -1     # y=y-1
    add     $t3, $t3, $t3     # z=z*2
endif:

```

```

addi    $s1, $zero, 3 # khai bao i
addi    $s2, $zero, 5 # khai bao j
addi    $t1, $zero, 5 # khai bao x
addi    $t2, $zero, 5 # khai bao y
addi    $t3, $zero, 5 # khai bao z
la      $t4, m
lw      $s4, 0($t4)    # khai bao m
la      $t5, n
lw      $s5, 0($t5)    # khai bao n

start:

#slt    $t0, $s1, $s2 # i < j a)
sge     $t0, $s1, $s2 # i >= j b)
#add    $s3, $s1, $s2 # khai bao i+j c+d)
#sle    $t0, $s3, $zero # i+j <= 0 c)
#add    $s6, $s4, $s5 # khai bao m+n d)
#sgt    $t0, $s3, $s6 # i+j > m+n d)

#slt    $t0, $s2, $s1 # j < i
beq     $t0, $zero, else
addi    $t1, $t1, 1    # x=x+1
addi    $t3, $zero, 1  # z=1
j       endif
else:   addi    $t2, $t2, -1 # y=y-1
        add     $t3, $t3, $t3 # z=z*2
endif:

```

Text Segment								
Bkpt	Address	Code	Basic	Source		Name	Number	Value
	0x00400000	0x20110003	addi \$17,\$0,0x00000000	5:	addi \$s1, \$zero, 3 # khai bao i	\$v0	2	0x00000000
	0x00400004	0x20120005	addi \$18,\$0,0x00000000	6:	addi \$s2, \$zero, 5 # khai bao j	\$v1	3	0x00000000
	0x00400008	0x20090005	addi \$9,\$0,0x00000005	7:	addi \$t1, \$zero, 5 # khai bao x	\$a0	4	0x00000000
	0x0040000c	0x200a0005	addi \$10,\$0,0x00000000	8:	addi \$t2, \$zero, 5 # khai bao y	\$a1	5	0x00000000
	0x00400010	0x200b0005	addi \$11,\$0,0x00000000	9:	addi \$t3, \$zero, 5 # khai bao z	\$a2	6	0x00000000
	0x00400014	0x3c011001	lui \$1,0x00001001	10:	la \$t4, m	\$a3	7	0x00000000
	0x00400018	0x342c0000	ori \$12,\$1,0x00000000			\$t0	8	0x00000000
	0x0040001c	0x8d940000	lw \$20,0x00000000(\$...	11:	lw \$s4, 0(\$t4) # khai bao m	\$t1	9	0x00000005
	0x00400020	0x3c011001	lui \$1,0x00001001	12:	la \$t5, n	\$t2	10	0x00000004
Data Segment						\$t3	11	0x0000000a
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000003	0x00000004	0x00000000	0x00000000	0x00000000	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
						\$t6	14	0x00000000
						\$t7	15	0x00000000
						\$s0	16	0x00000000
						\$s1	17	0x00000003
						\$s2	18	0x00000005
						\$s3	19	0x00000000
						\$s4	20	0x00000003
						\$s5	21	0x00000004
						\$s6	22	0x00000000
						\$s7	23	0x00000000
						\$t8	24	0x00000000
						\$t9	25	0x00000000
						\$k0	26	0x00000000
						\$k1	27	0x00000000
						\$gp	28	0x10008000
						\$sp	29	0x7ffffcfc
						\$fp	30	0x00000000
						\$ra	31	0x00000000
						pc		0x00400050
						hi		0x00000000
						lo		0x00000000

- Nhận xét:

- Với $i = 3, j = 5$ ($i < j$) ta thu được $x = 5, y = 4, z = 10 \Rightarrow$ Đúng với kết quả tính toán theo lý thuyết.

- Với $i = 5$ và $j = 3$ ($i < j$) thì ta thu được $x = 6, y = 5, z = 1 \Rightarrow$ Đúng với kết quả tính toán theo lý thuyết.

c) $i + j \leq 0$:

- Code:
- Đề bài trở thành:

if ($i + j \leq 0$)

$x = x + 1$;

$z = 1$;

else

$y = y - 1$;

$z = 2 * z$;

.text

addi \$s1, \$zero, 3 # khai bao i

addi \$s2, \$zero, 5 # khai bao j

addi \$t1, \$zero, 5 # khai bao x

addi \$t2, \$zero, 5 # khai bao y

addi \$t3, \$zero, 5 # khai bao z

la \$t4, m

lw \$s4, 0(\$t4) # khai bao m

la \$t5, n

lw \$s5, 0(\$t5) # khai bao n

start:

add \$s3, \$s1, \$s2 # khai bao $i+j$ c+d)

sle \$t0, \$s3, \$zero # $i+j \leq 0$ c)

```

    beq    $t0, $zero, else

    addi   $t1, $t1, 1    # x=x+1

    addi   $t3, $zero, 1  # z=1

    j      endif

else:     addi   $t2, $t2, -1  # y=y-1

          add    $t3, $t3, $t3  # z=z*2

endif:

```

```

    addi   $s1, $zero, 3  # khai bao i
    addi   $s2, $zero, 5  # khai bao j
    addi   $t1, $zero, 5  # khai bao x
    addi   $t2, $zero, 5  # khai bao y
    addi   $t3, $zero, 5  # khai bao z
    la     $t4, m
    lw     $s4, 0($t4)     # khai bao m
    la     $t5, n
    lw     $s5, 0($t5)     # khai bao n
start:
    #slt   $t0, $s1, $s2  # i<j  a)
    #ge    $t0, $s1, $s2  # i>=j b)
    add    $s3, $s1, $s2  # khai bao i+j c+d)
    sle    $t0, $s3, $zero # i+j<=0 c)
    #add   $s6, $s4, $s5  # khai bao m+n d)
    #sgt   $t0, $s3, $s6  # i+j>m+n d)

    #slt   $t0, $s2, $s1  # j<i
    beq    $t0, $zero, else
    addi   $t1, $t1, 1    # x=x+1
    addi   $t3, $zero, 1  # z=1
    j      endif
else:     addi   $t2, $t2, -1  # y=y-1
          add    $t3, $t3, $t3  # z=z*2
endif:

```

Text Segment							
Bkpt	Address	Code	Basic	Source			
	0x00400000	0x20110003	addi \$t7,\$0,0x00000000	5: addi \$t1, \$zero, 3 # khai bao i			
	0x00400004	0x20120005	addi \$t8,\$0,0x00000000	6: addi \$t2, \$zero, 5 # khai bao j			
	0x00400008	0x20090005	addi \$t9,\$0,0x00000005	7: addi \$t1, \$zero, 5 # khai bao x			
	0x0040000c	0x200a0005	addi \$t0,\$0,0x00000000	8: addi \$t2, \$zero, 5 # khai bao y			
	0x00400010	0x200b0005	addi \$t1,\$0,0x00000000	9: addi \$t3, \$zero, 5 # khai bao z			
	0x00400014	0x3c011001	lui \$t1,0x00001001	10: la \$t4, m			
	0x00400018	0x342c0000	ori \$t2,\$t1,0x00000000				
	0x0040001c	0x8d940000	lw \$t0,0x00000000(\$t1)	11: lw \$s4, 0(\$t4) # khai bao m			
	0x00400020	0x3c011001	lui \$t1,0x00001001	12: la \$t5, n			

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000004	0x00000000	0x00000000	0x00000000	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

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	0x00000005
\$t2	10	0x00000004
\$t3	11	0x0000000a
\$t4	12	0x10010000
\$t5	13	0x10010004
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000003
\$s2	18	0x00000005
\$s3	19	0x00000000
\$s4	20	0x00000003
\$s5	21	0x00000004
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7fffffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400054
hi		0x00000000
lo		0x00000000

- Nhận xét:

- Với $i = 3, j = 5$ ($i + j > 0$), kết quả thu được sau khi chạy tất cả các lệnh là $x = 5, y = 4, z = 10 \Rightarrow$ Đúng với điều kiện của đề bài và đúng với kết quả tính toán theo lý thuyết.
- Với $i = 0, j = -1$ ($i + j < 0$), kết quả thu được sau khi chạy tất cả các lệnh là $x = 6, y = 5, z = 1 \Rightarrow$ Đúng với điều kiện của đề bài và đúng với kết quả tính toán theo lý thuyết.

d) $i + j > m + n$:

- Đề bài trở thành:

if ($i+j>m+n$)

$x=x+1$;

$z=1$;

else

$y=y-1$;

$z=2*z$;

- Code:

.data

m: .word 3

n: .word 4

.text

addi \$s1, \$zero, 3 # khai bao i

addi \$s2, \$zero, 5 # khai bao j

addi \$t1, \$zero, 5 # khai bao x

addi \$t2, \$zero, 5 # khai bao y

addi \$t3, \$zero, 5 # khai bao z

la \$t4, m

lw \$s4, 0(\$t4) # khai bao m

la \$t5, n

lw \$s5, 0(\$t5) # khai bao n

start:

add \$s3, \$s1, \$s2 # khai bao i+j c+d)

add \$s6, \$s4, \$s5 # khai bao m+n d)

sgt \$t0, \$s3, \$s6 # i+j>m+n d)

#slt \$t0, \$s2, \$s1 # j<i

beq \$t0, \$zero, else

addi \$t1, \$t1, 1 # x=x+1

addi \$t3, \$zero, 1 # z=1

j endif

else: addi \$t2, \$t2, -1 # y=y-1

add \$t3, \$t3, \$t3 # z=z*2

endif:

```
.data
    m: .word 3
    n: .word 4

.text

    addi    $s1, $zero, 3 # khai bao i
    addi    $s2, $zero, 5 # khai bao j
    addi    $t1, $zero, 5 # khai bao x
    addi    $t2, $zero, 5 # khai bao y
    addi    $t3, $zero, 5 # khai bao z
    la      $t4, m
    lw      $s4, 0($t4)    # khai bao m
    la      $t5, n
    lw      $s5, 0($t5)    # khai bao n

start:
    #slt     $t0, $s1, $s2 # i<j a)
    #sge     $t0, $s1, $s2 # i>=j b)
    add      $s3, $s1, $s2 # khai bao i+j c+d)
    #sle     $t0, $s3, $zero # i+j<=0 c)
    add      $s6, $s4, $s5 # khai bao m+n d)
    sgt      $t0, $s3, $s6 # i+j>m+n d)

    #slt     $t0, $s2, $s1 # j<i
    beq      $t0, $zero, else
    addi     $t1, $t1, 1    # x=x+1
    addi     $t3, $zero, 1 # z=1

    j        endif

else:
    addi     $t2, $t2, -1  # y=y-1
    add      $t3, $t3, $t3 # z=z*2

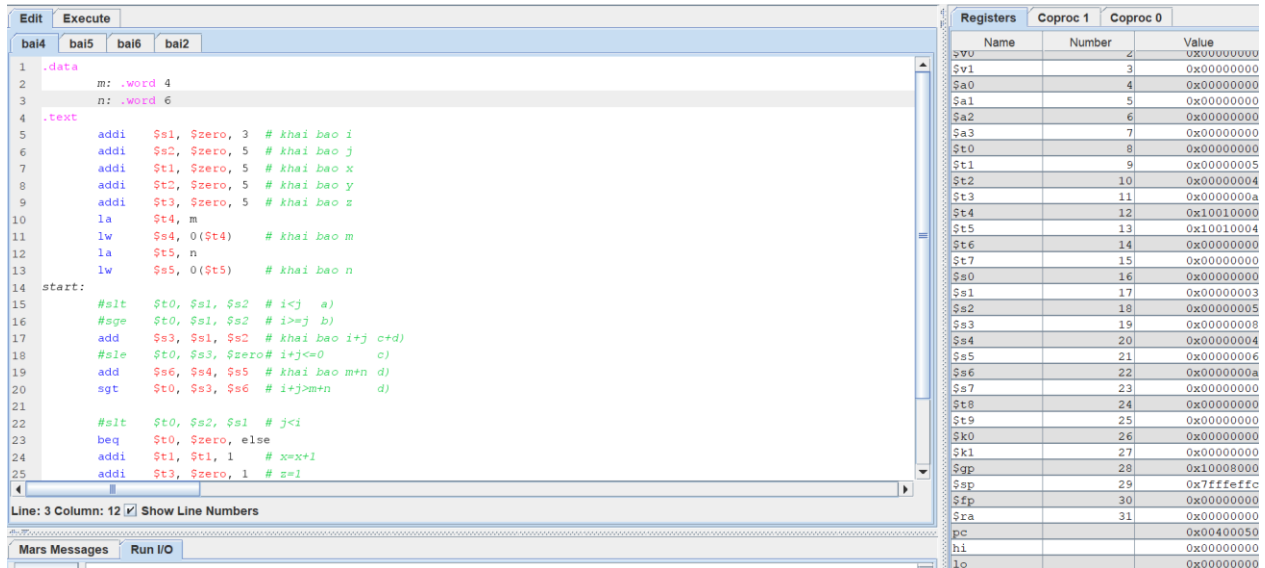
endif:
```

The screenshot displays the Mars IDE interface. The main window shows assembly code with comments in Vietnamese. Below the code, there are two panels: 'Text Segment' and 'Data Segment'. The 'Text Segment' panel shows the mapping of instructions to memory addresses and their basic and source codes. The 'Data Segment' panel shows the memory layout for variables, including their addresses, values, and offsets. On the right side, there is a table listing variables and their values.

Name	Number	Value
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000001
\$t1	9	0x00000006
\$t2	10	0x00000005
\$t3	11	0x00000001
\$t4	12	0x10010000
\$t5	13	0x10010004
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000003
\$s2	18	0x00000005
\$s3	19	0x00000008
\$s4	20	0x00000003
\$s5	21	0x00000004
\$s6	22	0x00000007
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400050
hi		0x00000000
lo		0x00000000

- Nhận xét:

- Xét $i = 3, j = 5, m = 3, n = 4$, lúc này $8 = i + j > m + n = 7$, kết quả của x, y, z lần lượt bằng 6, 5, 1 \Rightarrow Đúng với điều kiện của đề bài và đúng với kết quả tính toán theo lý thuyết.
- Xét $i = 3, j = 5, m = 4, n = 6$ lúc này $8 = i + j < m + n = 10$, kết quả của x, y, z lần lượt bằng 5, 4, 10 \Rightarrow Đúng với điều kiện của đề bài và đúng với kết quả tính toán theo lý thuyết.



Assignment 5:

a) $i \leq n$

- Code:

.data

array: .word 1, 2, 3, 4, 5

.text

addi \$s5, \$zero, 0 # khai tao sum

addi \$s1, \$zero, 0 # khai tao i

addi \$s3, \$zero, 4 # khai tao n

addi \$s4, \$zero, 1 # khai tao step

```

        la      $s2, array      # khoi tao array
loop:   sle     $t2, $s1, $s3    # i<=n

        beq     $t2, $zero, endloop

        add     $t1, $s1, $s1

        add     $t1, $t1, $t1    # t1 = 4*i dia chi address them vao

        add     $t1, $t1, $s2

        lw      $t0, 0($t1)     # load gia tri A[i] vao $t0

        add     $s5, $s5, $t0   # sum = sum + A[i]

        add     $s1, $s1, $s4   # i = i + step

        j       loop

```

endloop:

```

.data
        array: .word 1, 2, 3, 4, 5      # them mot phan tu nua vao mang co gia tri la 5
.text
        addi    $s5, $zero, 0          # khoi tao sum
        addi    $s1, $zero, 0          # khoi tao i
        addi    $s3, $zero, 4          # khoi tao n
        addi    $s4, $zero, 1          # khoi tao step
        la      $s2, array              # khoi tao array
loop:   sle     $t2, $s1, $s3           # i<=n
        beq     $t2, $zero, endloop
        add     $t1, $s1, $s1
        add     $t1, $t1, $t1          # t1 = 4*i dia chi address them vao
        add     $t1, $t1, $s2
        lw      $t0, 0($t1)           # load gia tri A[i] vao $t0
        add     $s5, $s5, $t0          # sum = sum + A[i]
        add     $s1, $s1, $s4          # i = i + step
        j       loop
endloop:

```

Text Segment							
Bkpt	Address	Code	Basic	Source	Name	Number	Value
	0x00400000	0x20150000	addi \$t1,\$0,0x00000000	4: addi \$s5,\$zero,0 # khai tao sum	\$zero	0	0x00000000
	0x00400004	0x20110000	addi \$t1,\$0,0x00000000	5: addi \$s1,\$zero,0 # khai tao i	\$at	1	0x00000001
	0x00400008	0x20130004	addi \$t3,\$0,0x00000004	6: addi \$s3,\$zero,4 # khai tao n	\$v0	2	0x00000000
	0x0040000c	0x20140001	addi \$t2,\$0,0x00000001	7: addi \$s4,\$zero,1 # khai tao step	\$v1	3	0x00000000
	0x00400010	0x3c011001	lui \$t1,0x00001001	8: la \$s2,array # khai tao array	\$a0	4	0x00000000
	0x00400014	0x34320000	ori \$t8,\$1,0x00000000		\$a1	5	0x00000000
	0x00400018	0x0271502a	sle \$t0,\$t8,\$t7	9: loop: sle \$t2,\$s1,\$s3 # i<=n	\$a2	6	0x00000000
	0x0040001c	0x34010001	ori \$t1,\$0,0x00000001		\$a3	7	0x00000000
	0x00400020	0x002a5023	subu \$t0,\$t1,\$t0		\$t0	8	0x00000005
	0x00400024	0x11400007	beq \$t0,\$0,0x00000007	10: beq \$t2,\$zero, endloop	\$t1	9	0x10010010
	0x00400028	0x02314820	add \$t9,\$t7,\$t7	11: add \$t1,\$s1,\$s1	\$t2	10	0x00000000
	0x0040002c	0x01294820	add \$t9,\$t9,\$t9	12: add \$t1,\$t1,\$t1 # t1 = 4*i dia chi address them vao	\$t3	11	0x00000000
	0x00400030	0x01324820	add \$t9,\$t9,\$t9	13: add \$t1,\$t1,\$s2	\$t4	12	0x00000000
					\$t5	13	0x00000000
					\$t6	14	0x00000000
					\$t7	15	0x00000000
					\$s0	16	0x00000000
					\$s1	17	0x00000005
					\$s2	18	0x10010000
					\$s3	19	0x00000004
					\$s4	20	0x00000001
					\$s5	21	0x0000000f
					\$s6	22	0x00000000
					\$s7	23	0x00000000
					\$t8	24	0x00000000
					\$t9	25	0x00000000
					\$k0	26	0x00000000
					\$k1	27	0x00000000
					\$fp	28	0x10000000
					\$sp	29	0xffffffff
					\$fp	30	0x00000000
					\$ra	31	0x00000000
					pc		0x00400044
					hi		0x00000000
					lo		0x00000000

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000001	0x00000002	0x00000003	0x00000004	0x00000005	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
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

- Nhân xét: Ta tang vòng lặp lên một lần, do đó ta khai báo thêm một phần tử nữa có giá trị = 5 vào mảng. Thanh ghi \$s5 có giá trị bằng sum giờ có giá trị là 0x0000000f = 15, đúng với phép tính $1 + 2 + 3 + 4 + 5 = 15$.

b) sum >= 0

- Code:

.data

array: .word -1, 0, 0, 0, 0 # them mot phan tu nua vao mang co gia tri la 5

.text

addi \$s5, \$zero, 0 # khai tao sum

addi \$s1, \$zero, 0 # khai tao i

addi \$s3, \$zero, 4 # khai tao n

addi \$s4, \$zero, 1 # khai tao step

la \$s2, array # khai tao array

loop: sle \$t2, \$s1, \$s3 # i<=n

beq \$t2, \$zero, endloop

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

add \$t1, \$t1, \$t1 # t1 = 4*i dia chi address them vao

```
add    $t1, $t1, $s2

lw     $t0, 0($t1)    # load gia tri A[i] vao $t0

add    $s5, $s5, $t0   # sum = sum + A[i]

sge    $t3, $s5, $zero # sum >= 0 ? 1 : 0

beq    $t3, $zero, endloop # neu sum < 0 thi endloop

add    $s1, $s1, $s4   # i = i + step

j      loop

endloop:
```

```

.data
    array: .word -1, 0, 0, 0, 0    # them mot phan tu nua vao mang co gia tri la 5

.text

    addi    $s5, $zero, 0    # khoi tao sum
    addi    $s1, $zero, 0    # khoi tao i
    addi    $s3, $zero, 4    # khoi tao n
    addi    $s4, $zero, 1    # khoi tao step
    la      $s2, array       # khoi tao array
loop:
    sle     $t2, $s1, $s3    # i <= n
    beq     $t2, $zero, endloop
    add     $t1, $s1, $s1
    add     $t1, $t1, $t1    # t1 = 4*i dia chi address them vao
    add     $t1, $t1, $s2
    lw      $t0, 0($t1)      # load gia tri A[i] vao $t0
    add     $s5, $s5, $t0    # sum = sum + A[i]
    sge     $t3, $s5, $zero  # sum >= 0 ? 1 : 0
    beq     $t3, $zero, endloop # neu sum < 0 thi endloop
    add     $s1, $s1, $s4    # i = i + step
    j       loop
endloop:

```

Text Segment						Name			Number		Value
Bkpt	Address	Code	Basic	Source							
	0x00400000	0x20150000	addi \$21,\$0,0x00000000	4:	addi \$s5, \$zero, 0	# khoi tao sum	\$zero		0		0x00000000
	0x00400004	0x20110000	addi \$17,\$0,0x00000000	5:	addi \$s1, \$zero, 0	# khoi tao i	\$at		1		0x00000001
	0x00400008	0x20130004	addi \$19,\$0,0x00000004	6:	addi \$s3, \$zero, 4	# khoi tao n	\$v0		2		0x00000000
	0x0040000c	0x20140001	addi \$20,\$0,0x00000001	7:	addi \$s4, \$zero, 1	# khoi tao step	\$v1		3		0x00000000
	0x00400010	0x3c011001	lui \$1,0x00001001	8:	la \$s2, array	# khoi tao array	\$a0		4		0x00000000
	0x00400014	0x34320000	ori \$16,\$1,0x00000000				\$a1		5		0x00000000
	0x00400018	0x0271502a	sll \$10,\$19,\$17	9: loop: sle \$t2, \$s1, \$s3	# i <= n		\$a2		6		0x00000000
	0x0040001c	0x34010001	ori \$1,\$0,0x00000001				\$a3		7		0x00000000
	0x00400020	0x002a5023	subu \$10,\$1,\$10				\$t0		8		0xffffffff
	0x00400024	0x1140000b	beq \$10,\$0,0x0000000b	10:	beq \$t2, \$zero, endloop		\$t1		9		0x10010000
	0x00400028	0x02314820	add \$9,\$17,\$17	11:	add \$t1, \$s1, \$s1		\$t2		10		0x00000001
	0x0040002c	0x01294820	add \$9,\$9,\$9	12:	add \$t1, \$t1, \$t1	# t1 = 4*i dia chi address them vao	\$t3		11		0x00000000
	0x00400030	0x01324820	add \$9,\$9,\$18	13:	add \$t1, \$t1, \$s2		\$t4		12		0x00000000
							\$t5		13		0x00000000
							\$t6		14		0x00000000
							\$t7		15		0x00000000
							\$s0		16		0x00000000
							\$s1		17		0x00000000
							\$s2		18		0x10010000
							\$s3		19		0x00000004
							\$s4		20		0x00000001
							\$s5		21		0xffffffff
							\$s6		22		0x00000000
							\$s7		23		0x00000000
							\$t8		24		0x00000000
							\$t9		25		0x00000000
							\$k0		26		0x00000000
							\$t1		27		0x00000000
							\$gp		28		0x7ffff800
							\$sp		29		0x7ffff8fc
							\$fp		30		0x00000000
							\$ra		31		0x00000000
							\$pc				0x00400054
							\$hi				0x00000000
							\$lo				0x00000000

Dãy -1, 0, 0, 0, 0

Text Segment							
Bkpt	Address	Code	Basic	Source	Name	Number	Value
	0x00400000	0x20150000	addi \$t1,\$0,0x00000000	4: addi \$s5, \$zero, 0 # khoi tao sum	\$zero	0	0x00000000
	0x00400004	0x20110000	addi \$t7,\$0,0x00000000	5: addi \$s1, \$zero, 0 # khoi tao i	\$at	1	0x00000001
	0x00400008	0x20130004	addi \$t9,\$0,0x00000004	6: addi \$s3, \$zero, 4 # khoi tao n	\$v0	2	0x00000000
	0x0040000c	0x20140001	addi \$t0,\$0,0x00000001	7: addi \$s4, \$zero, 1 # khoi tao step	\$v1	3	0x00000000
	0x00400010	0x3c011001	lui \$t1,0x000001001	8: la \$s2, array # khoi tao array	\$a0	4	0x00000000
	0x00400014	0x34320000	ori \$t8,\$1,0x00000000		\$a1	5	0x00000000
	0x00400018	0x0271502a	sle \$t0,\$t9,\$t7	9: loop: sle \$t2, \$s1, \$s3 # i<=n	\$a2	6	0x00000000
	0x0040001c	0x34010001	ori \$t1,\$0,0x00000001		\$a3	7	0x00000000
	0x00400020	0x002a5023	subu \$t0,\$t1,\$t0		\$t0	8	0x00000005
	0x00400024	0x1140000b	beq \$t0,\$0,0x0000000b	10: beq \$t2, \$zero, endloop	\$t1	9	0x10010010
	0x00400028	0x02314820	add \$9,\$t7,\$t7	11: add \$t1, \$s1, \$s1	\$t2	10	0x00000000
	0x0040002c	0x01294820	add \$9,\$9,\$9	12: add \$t1, \$t1, \$t1 # t1 = 4*i dia chi address them vao	\$t3	11	0x00000001
	0x00400030	0x01324820	add \$9,\$9,\$t8	13: add \$t1, \$t1, \$s2	\$t4	12	0x00000000
					\$t5	13	0x00000000
					\$t6	14	0x00000000
					\$t7	15	0x00000000
					\$s0	16	0x00000000
					\$s1	17	0x00000005
					\$s2	18	0x10010000
					\$s3	19	0x00000004
					\$s4	20	0x00000001
					\$s5	21	0x0000000f
					\$s6	22	0x00000000
					\$s7	23	0x00000000
					\$t8	24	0x00000000
					\$t9	25	0x00000000
					\$k0	26	0x00000000
					\$k1	27	0x00000000
					\$gp	28	0x10000000
					\$sp	29	0xffffffe
					\$fp	30	0x00000000
					\$ra	31	0x00000000
					\$pc		0x00400054
					\$hi		0x00000000
					\$lo		0x00000000

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000001	0x00000002	0x00000003	0x00000004	0x00000005	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
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Dãy 1, 2, 3, 4, 5

- Nhân xét:

- Với dãy -1, 0, 0, 0, 0 thì vòng lặp đã ngay lập tức dừng khi sum = -1 < 0 => Thỏa mãn.
- Với dãy 1, 2, 3, 4, 5 thì vòng lặp vẫn chạy bình thường và cho ra kết quả đúng là 0x0000000f = 15.

c) A[i] != 0

- Code:

.data

array: .word 1, 2, 3, 0, 5 # them mot phan tu nua vao mang co gia tri la 5

.text

addi \$s5, \$zero, 0 # khoi tao sum

addi \$s1, \$zero, 0 # khoi tao i

addi \$s3, \$zero, 4 # khoi tao n

addi \$s4, \$zero, 1 # khoi tao step

la \$s2, array # khoi tao array

loop:


```

sle    $t2, $s1, $s3  # i<=n

beq    $t2, $zero, endloop

add    $t1, $s1, $s1

add    $t1, $t1, $t1  # t1 = 4*i dia chi address them vao

add    $t1, $t1, $s2

lw     $t0, 0($t1)    # load gia tri A[i] vao $t0

beq    $t0, $zero, endloop # A[i] = 0 thi dung loop

add    $s5, $s5, $t0  # sum = sum + A[i]

add    $s1, $s1, $s4  # i = i + step

j      loop

```

endloop:

```

.data
    array: .word 1, 2, 3, 0, 5    # them mot phan tu nua vao mang co gia tri la 5

.text
    addi    $s5, $zero, 0  # khoi tao sum
    addi    $s1, $zero, 0  # khoi tao i
    addi    $s3, $zero, 4  # khoi tao n
    addi    $s4, $zero, 1  # khoi tao step
    la      $s2, array     # khoi tao array
loop:
    sle     $t2, $s1, $s3  # i<=n
    beq     $t2, $zero, endloop
    add     $t1, $s1, $s1
    add     $t1, $t1, $t1  # t1 = 4*i dia chi address them vao
    add     $t1, $t1, $s2
    lw      $t0, 0($t1)    # load gia tri A[i] vao $t0
    beq     $t0, $zero, endloop # A[i] = 0 thi dung loop
    add     $s5, $s5, $t0  # sum = sum + A[i]
    #sge     $t3, $s5, $zero # sum >= 0 ? 1 : 0
    #beq     $t3, $zero, endloop # neu sum < 0 thi endloop
    add     $s1, $s1, $s4  # i = i + step
    j      loop
endloop:

```

Text Segment							
Byte	Address	Code	Basic	Source	Name	Number	Value
0x00400000	0x20150000	addi	\$t1,\$0,0x00000000	4: addi \$s5,\$zero,0 # khoi tao sum	\$zero	0	0x00000000
0x00400004	0x20110000	addi	\$t1,\$0,0x00000000	5: addi \$s1,\$zero,0 # khoi tao i	\$at	1	0x00000001
0x00400008	0x20130004	addi	\$t1,\$0,0x00000004	6: addi \$s3,\$zero,4 # khoi tao n	\$v0	2	0x00000006
0x0040000c	0x20140001	addi	\$t2,\$0,0x00000001	7: addi \$s4,\$zero,1 # khoi tao step	\$v1	3	0x00000000
0x00400010	0x3c011001	lui	\$t1,0x00001001	8: la \$s2,array # khoi tao array	\$a0	4	0x00000000
0x00400014	0x34320000	ori	\$t1,\$t1,0x00000000		\$a1	5	0x00000000
0x00400018	0x0271502a	sle	\$t2,\$s1,\$s3 # i<=n		\$a2	6	0x00000000
0x0040001c	0x34010001	ori	\$t1,\$t1,0x00000001	9: loop: sle \$t2,\$s1,\$s3 # i<=n	\$a3	7	0x00000000
0x00400020	0x002a5023	subu	\$t0,\$t1,\$t0		\$t0	8	0x00000000
0x00400024	0x11400008	beq	\$t0,\$t0,0x00000008	10: beq \$t2,\$zero, endloop	\$t1	9	0x1001000c
0x00400028	0x02314820	add	\$s,\$t1,\$t1	11: add \$t1,\$s1,\$s1	\$t2	10	0x00000001
0x0040002c	0x01294820	add	\$s,\$s,\$s	12: add \$t1,\$t1,\$t1 # t1 = 4*i dia chi address them vao	\$t3	11	0x00000000
0x00400030	0x01324820	add	\$s,\$s,\$s	13: add \$t1,\$t1,\$s2	\$t4	12	0x00000000
					\$t5	13	0x00000000
					\$t6	14	0x00000000
					\$t7	15	0x00000000
					\$a0	16	0x00000000
					\$s1	17	0x00000003
					\$s2	18	0x1001000b
					\$s3	19	0x00000004
					\$s4	20	0x00000001
					\$s5	21	0x00000006
					\$s6	22	0x00000000
					\$s7	23	0x00000000
					\$t8	24	0x00000000
					\$t9	25	0x00000000
					\$k0	26	0x00000000
					\$t1	27	0x00000000
					\$gp	28	0x10008000
					\$sp	29	0x7ffffcfc
					\$fp	30	0x00000000
					\$ra	31	0x00000000
					pc		0x00400048
					hi		0x00000000
					lo		0x00000000

- Nhân xét:

- Xét mảng gồm 5 phần tử có các giá trị 1, 2, 3, 0, 5. Vòng lặp đã dừng lại khi A[3] = 0, và tổng lúc này ta tính được là 1 + 2 + 3 = 6, mà \$s5 = 0x00000006 = 6 => Thỏa mãn đề bài.
- Xét mảng khác có giá trị 1, 0, 3, 4, 5. Vòng lặp đã dừng lại khi A[1] = 0, tổng lúc này ta tính được là 1, mà \$s5 = 0x00000001 => Thỏa mãn.

Text Segment							
Byte	Address	Code	Basic	Source	Name	Number	Value
0x00400000	0x20150000	addi	\$t1,\$0,0x00000000	4: addi \$s5,\$zero,0 # khoi tao sum	\$zero	0	0x00000000
0x00400004	0x20110000	addi	\$t1,\$0,0x00000000	5: addi \$s1,\$zero,0 # khoi tao i	\$at	1	0x00000001
0x00400008	0x20130004	addi	\$t1,\$0,0x00000004	6: addi \$s3,\$zero,4 # khoi tao n	\$v0	2	0x00000006
0x0040000c	0x20140001	addi	\$t2,\$0,0x00000001	7: addi \$s4,\$zero,1 # khoi tao step	\$v1	3	0x00000000
0x00400010	0x3c011001	lui	\$t1,0x00001001	8: la \$s2,array # khoi tao array	\$a0	4	0x00000000
0x00400014	0x34320000	ori	\$t1,\$t1,0x00000000		\$a1	5	0x00000000
0x00400018	0x0271502a	sle	\$t2,\$s1,\$s3 # i<=n		\$a2	6	0x00000000
0x0040001c	0x34010001	ori	\$t1,\$t1,0x00000001	9: loop: sle \$t2,\$s1,\$s3 # i<=n	\$a3	7	0x00000000
0x00400020	0x002a5023	subu	\$t0,\$t1,\$t0		\$t0	8	0x00000000
0x00400024	0x11400008	beq	\$t0,\$t0,0x00000008	10: beq \$t2,\$zero, endloop	\$t1	9	0x1001000c
0x00400028	0x02314820	add	\$s,\$t1,\$t1	11: add \$t1,\$s1,\$s1	\$t2	10	0x00000001
0x0040002c	0x01294820	add	\$s,\$s,\$s	12: add \$t1,\$t1,\$t1 # t1 = 4*i dia chi address them vao	\$t3	11	0x00000000
0x00400030	0x01324820	add	\$s,\$s,\$s	13: add \$t1,\$t1,\$s2	\$t4	12	0x00000000
					\$t5	13	0x00000000
					\$t6	14	0x00000000
					\$t7	15	0x00000000
					\$a0	16	0x00000000
					\$s1	17	0x00000001
					\$s2	18	0x1001000b
					\$s3	19	0x00000004
					\$s4	20	0x00000001
					\$s5	21	0x00000006
					\$s6	22	0x00000000
					\$s7	23	0x00000000
					\$t8	24	0x00000000
					\$t9	25	0x00000000
					\$k0	26	0x00000000
					\$k1	27	0x00000000
					\$gp	28	0x10008000
					\$sp	29	0x7ffffcfc
					\$fp	30	0x00000000
					\$ra	31	0x00000000
					pc		0x00400048
					hi		0x00000000
					lo		0x00000000

Assignment 6:

- Code:

.data

```
array: .word 3, 6, -2, -8, -15, 10 # khoi tao mang va cac phan tu
```

```
.text
```

```
addi    $s1, $zero, 0    # khoi tao i
```

```
addi    $s2, $zero, 6    # khoi tao n
```

```
addi    $s3, $zero, 1    # khoi tao step
```

```
la      $s4, array        # khoi tao array
```

```
addi    $s5, $zero, 0    # khoi tao max
```

```
loop:
```

```
slt      $t2, $s1, $s2
```

```
beq      $t2, $zero, endloop
```

```
add      $t1, $s1, $s1
```

```
add      $t1, $t1, $t1    # t1 = 4 * i
```

```
add      $t1, $t1, $s4
```

```
lw       $t0, 0($t1)      # gia tri cua A[i] luu vao $t0
```

```
addi     $t5, $t0, 0      # luu gia tri A[i] vao $t5
```

```
if:
```

```
slti     $t3, $t0, 0      # A[i] < 0 ? 1 : 0
```

```
beq      $t3, $zero, else # A[i] >= 0 thi else
```

```
sub      $t0, $zero, $t0   # A[i] = 0 - A[i]
```

```
j        endif
```

```
else:    addi    $t0, $t0, 0
```

```
endif:
```

if1:

```
#sge    $t4, $t0, $s5    # A[i] >= max ? 1 : 0

slt     $t4, $t0, $s5    # A[i] < max ? 1 : 0

bne     $t4, $zero, endif1    # neu A[i] < max thi ket thuc if1

#beq    $t4, $zero, endif1    # neu A[i] < max thi ket thuc if1

add     $s5, $zero, $t0    # gan max = A[i]

addi    $s6, $t5, 0        # %s6 la gia tri nguyen ban cua A[i]
```

endif1:

```
add     $s1, $s1, $s3    # i += 1

j       loop
```

endloop:

```
.data
    array: .word 3, 6, -2, -8, -15, 10    # khoi tao mang va cac phan tu

.text

addi    $s1, $zero, 0    # khoi tao i
addi    $s2, $zero, 6    # khoi tao n
addi    $s3, $zero, 1    # khoi tao step
la      $s4, array        # khoi tao array
addi    $s5, $zero, 0    # khoi tao max

loop:
    slt    $t2, $s1, $s2
    beq    $t2, $zero, endloop
    add    $t1, $s1, $s1
    add    $t1, $t1, $t1    # t1 = 4 * i
    add    $t1, $t1, $s4
    lw     $t0, 0($t1)      # gia tri cua A[i] luu vao $t0
    addi   $t5, $t0, 0      # luu gia tri A[i] vao $t5

if:
    slti   $t3, $t0, 0      # A[i] < 0 ? 1 : 0
    beq    $t3, $zero, else    # A[i] >= 0 thi else
    sub    $t0, $zero, $t0      # A[i] = 0 - A[i]
    j      endif
else:
    addi   $t0, $t0, 0
endif:
```

```

ifl:
    #sge    $t4, $t0, $s5 # A[i] >= max ? 1 : 0
    slt     $t4, $t0, $s5 # A[i] < max ? 1 : 0
    bne     $t4, $zero, endifl # neu A[i] < max thi ket thuc ifl
    #beq     $t4, $zero, endifl # neu A[i] < max thi ket thuc ifl
    add     $s5, $zero, $t0 # gan max = A[i]
    addi    $s6, $t5, 0 # %s6 la gia tri nguyen ban cua A[i]
endifl:

    add     $s1, $s1, $s3 # i += 1

    j      loop
endloop:

```

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	10
\$t1	9	268501012
\$t2	10	0
\$t3	11	0
\$t4	12	1
\$t5	13	10
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	6
\$s2	18	6
\$s3	19	1
\$s4	20	268500992
\$s5	21	15
\$s6	22	-15
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0

- Nhận xét:

- Xét mảng 6 phần tử gồm các giá trị: 3, 6, -2, -8, -15, 10, ta tìm được giá trị tuyệt đối lớn nhất của mảng này là $|-15| = 15$. Mặt khác, max được lưu vào thanh ghi $\$s5 = 15$, còn giá trị ban đầu là $A[4] = -15$ thì được lưu vào thanh ghi $\$s6 = -15 \Rightarrow$ Kết quả sau khi chạy chương trình đã đúng với kết quả lý thuyết.
- Thử với mảng 6 phần tử khác gồm các giá trị: -4, 1, -2, 14, 16, -11 thì giá trị trên thanh ghi $\$s5 = 16$, $\$s6 = 16 \Rightarrow$ Kết quả sau khi chạy chương trình đã đúng với kết quả lý thuyết.

```

.data
    array: .word -4, 1, -2, 14, 16, -11 # khai tạo mảng và các phần tử

.text
    addi $s1, $zero, 0 # khai tạo i
    addi $s2, $zero, 6 # khai tạo n
    addi $s3, $zero, 1 # khai tạo step
    la $s4, array # khai tạo array
    addi $s5, $zero, 0 # khai tạo max

loop:
    slt $t2, $s1, $s2
    beq $t2, $zero, endloop
    add $t1, $s1, $s1
    add $t1, $t1, $t1 # t1 = 4 * i
    add $t1, $t1, $s4
    lw $t0, 0($t1) # giá trị của A[i] lưu vào $t0
    addi $t5, $t0, 0 # lưu giá trị A[i] vào $t5

if:
    slti $t3, $t0, 0 # A[i] < 0 ? 1 : 0
    beq $t3, $zero, else # A[i] >= 0 thì else
    sub $t0, $zero, $t0 # A[i] = 0 - A[i]
    j endif
else:
    addi $t0, $t0, 0
endif:

```

\$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	11
\$t1	9	268501012
\$t2	10	0
\$t3	11	1
\$t4	12	1
\$t5	13	-11
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	6
\$s2	18	6
\$s3	19	1
\$s4	20	268500992
\$s5	21	16
\$s6	22	16
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0

2 Column: 37 ☒ Show Line Numbers

Messages Run I/O