Báo cáo thực hành tuần 3

Phùng Ngọc Vinh – 20194719

Bài 1:

TH1: i > j

Edit Execute

ex1.asm

```
#Laboratory Exercise 3, Home Assignment 1

.text
            addi        $s1,$0,5
            addi        $s2,$0,4
            addi        $s3,$0,0

start:
            slt         $t0,$s2,$s1             # j<i i=s1 j=s2
            bne         $t0,$zero,else                      # branch to else if j<i
            addi        $t1,$t1,1               # then part: 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=2*z
endif:
```

Kết quả:

Registers | Coproc 1 | Coproc 0

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	1
$t1	9	0
$t2	10	-1
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	5
$s2	18	4
$s3	19	0
$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	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194340
hi		0
lo		0

TH2: i < j

Edit | Execute

ex1.asm

```
1  #Laboratory Exercise 3, Home Assignment 1
2
3  .text
4              addi        $s1,$0,4
5              addi        $s2,$0,5
6              addi        $s3,$0,0
7
8  start:
9              slt         $t0,$s2,$s1             # j<i i=s1 j=s2
10             bne         $t0,$zero,else                  # branch to else if j<i
11             addi        $t1,$t1,1               # then part: x=x+1
12             addi        $t3,$zero,1             # z=1
13             j           endif                           # skip "else" part
14 else:       addi        $t2,$t2,-1              # begin else part: y=y-1
15             add         $t3,$t3,$t3             # z=2*z
16 endif:
17
```

Kết quả:

Registers | Coproc 1 | Coproc 0

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	1
$t2	10	0
$t3	11	1
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	4
$s2	18	5
$s3	19	0
$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	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194340
hi		0
lo		0

TH3: i = j

ex1.asm

```
1  #Laboratory Exercise 3, Home Assignment 1
2
3  .text
4              addi        $s1,$0,4
5              addi        $s2,$0,4
6              addi        $s3,$0,0
7
8  start:
9              slt         $t0,$s2,$s1             # j<i i=s1 j=s2
10             bne         $t0,$zero,else                  # branch to else if j<i
11             addi        $t1,$t1,1               # then part: x=x+1
12             addi        $t3,$zero,1             # z=1
13             j           endif                           # skip “else” part
14 else:       addi        $t2,$t2,-1              # begin else part: y=y-1
15             add         $t3,$t3,$t3             # z=2*z
16 endif:
17
```

Kết quả:

Registers | Coproc 1 | Coproc 0

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	1
$t2	10	0
$t3	11	1
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	4
$s2	18	4
$s3	19	0
$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	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194340
hi		0
lo		0

Bài 2:

ex2.asm

```
1  #Laboratory 3, Home Assigment 2
2  .data
3  arr: .word  26,7,2001,38,42,-73
4
5  .text
6               addi $s1, $zero, 0      #i = 0
7               la   $s2, arr           #load address A[i]
8               addi $s3, $zero, 6      #n = 6
9               addi $s4, $zero, 1      #step = 1
10              addi $s5, $zero, 0      #sum = 0
11 loop:        slt  $t2, $s1, $s3      # $t2 = i < n? 1 : 0
12              beq  $t2, $zero, endloop
13              add  $t1,$s1,$s1        #$t1=2*$s1
14              add  $t1,$t1,$t1        #$t1=4*$s1
15              add  $t1,$t1,$s2        #$t1 store the address of A[i]
16              lw   $t0,0($t1)                 #load value of A[i] in $t0
17              add  $s5,$s5,$t0        #sum=sum+A[i]
18              add  $s1,$s1,$s4        #i=i+step
19              j    loop               #goto loop
20 endloop:
21
```

Khởi tạo i

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x20110000	addi $17,$0,0	6: addi $s1, $zero, 0 #i = 0
	0x00400004	0x3c011001	lui $1,4097	7: la $s2, arr #load address A[i]
	0x00400008	0x34320000	ori $18,$1,0	
	0x0040000c	0x20130006	addi $19,$0,6	8: addi $s3, $zero, 6 #n = 6
	0x00400010	0x20140001	addi $20,$0,1	9: addi $s4, $zero, 1 #step = 1
	0x00400014	0x20150000	addi $21,$0,0	10: addi $s5, $zero, 0 #sum = 0
	0x00400018	0x0233502a	slt $10,$17,$19	11: loop: slt $t2, $s1, $s3 # $t2 = i < n? 1 : 0
	0x0040001c	0x11400007	beq $10,$0,7	12: beq $t2, $zero, endloop
	0x00400020	0x02314820	add $9,$17,$17	13: add $t1,$s1,$s1 #$t1=2*$s1
	0x00400024	0x01294820	add $9,$9,$9	14: add $t1,$t1,$t1 #$t1=4*$s1
	0x00400028	0x01324820	add $9,$9,$18	15: add $t1,$t1,$s2 #$t1 store the address of A[i]
	0x0040002c	0x8d280000	lw $8,0($9)	16: lw $t0,0($t1) #load value of A[i] in $t0
	0x00400030	0x02a8a820	add $21,$21,$8	17: add $s5,$s5,$t0 #sum=sum+A[i]
	0x00400034	0x02348820	add $17,$17,$20	18: add $s1,$s1,$s4 #i=i+step
	0x00400038	0x08100006	j 0x00400018	19: j loop #goto loop

Lấy địa chỉ mảng Arr

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x20110000	addi $17,$0,0	6: addi $s1, $zero, 0 #i = 0
	0x00400004	0x3c011001	lui $1,4097	7: la $s2, arr #load address A[i]
	0x00400008	0x34320000	ori $18,$1,0	
	0x0040000c	0x20130006	addi $19,$0,6	8: addi $s3, $zero, 6 #n = 6
	0x00400010	0x20140001	addi $20,$0,1	9: addi $s4, $zero, 1 #step = 1
	0x00400014	0x20150000	addi $21,$0,0	10: addi $s5, $zero, 0 #sum = 0
	0x00400018	0x0233502a	slt $10,$17,$19	11: loop: slt $t2, $s1, $s3 # $t2 = i < n? 1 : 0
	0x0040001c	0x11400007	beq $10,$0,7	12: beq $t2, $zero, endloop
	0x00400020	0x02314820	add $9,$17,$17	13: add $t1,$s1,$s1 #$t1=2*$s1
	0x00400024	0x01294820	add $9,$9,$9	14: add $t1,$t1,$t1 #$t1=4*$s1
	0x00400028	0x01324820	add $9,$9,$18	15: add $t1,$t1,$s2 #$t1 store the address of A[i]
	0x0040002c	0x8d280000	lw $8,0($9)	16: lw $t0,0($t1) #load value of A[i] in $t0
	0x00400030	0x02a8a820	add $21,$21,$8	17: add $s5,$s5,$t0 #sum=sum+A[i]
	0x00400034	0x02348820	add $17,$17,$20	18: add $s1,$s1,$s4 #i=i+step
	0x00400038	0x08100006	j 0x00400018	19: j loop #goto loop

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x20110000	addi $17,$0,0	6: addi $s1, $zero, 0 #i = 0
	0x00400004	0x3c011001	lui $1,4097	7: la $s2, arr #load address A[i]
	0x00400008	0x34320000	ori $18,$1,0	
	0x0040000c	0x20130006	addi $19,$0,6	8: addi $s3, $zero, 6 #n = 6
	0x00400010	0x20140001	addi $20,$0,1	9: addi $s4, $zero, 1 #step = 1
	0x00400014	0x20150000	addi $21,$0,0	10: addi $s5, $zero, 0 #sum = 0
	0x00400018	0x0233502a	slt $10,$17,$19	11: loop: slt $t2, $s1, $s3 # $t2 = i < n? 1 : 0
	0x0040001c	0x11400007	beq $10,$0,7	12: beq $t2, $zero, endloop
	0x00400020	0x02314820	add $9,$17,$17	13: add $t1,$s1,$s1 #$t1=2*$s1
	0x00400024	0x01294820	add $9,$9,$9	14: add $t1,$t1,$t1 #$t1=4*$s1
	0x00400028	0x01324820	add $9,$9,$18	15: add $t1,$t1,$s2 #$t1 store the address of A[i]
	0x0040002c	0x8d280000	lw $8,0($9)	16: lw $t0,0($t1) #load value of A[i] in $t0
	0x00400030	0x02a8a820	add $21,$21,$8	17: add $s5,$s5,$t0 #sum=sum+A[i]
	0x00400034	0x02348820	add $17,$17,$20	18: add $s1,$s1,$s4 #i=i+step
	0x00400038	0x08100006	j 0x00400018	19: j loop #goto loop

Khởi tạo số phần tử n

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x20110000	addi $17,$0,0	6: addi $s1, $zero, 0 #i = 0
	0x00400004	0x3c011001	lui $1,4097	7: la $s2, arr #load address A[i]
	0x00400008	0x34320000	ori $18,$1,0	
	0x0040000c	0x20130006	addi $19,$0,6	8: addi $s3, $zero, 6 #n = 6
	0x00400010	0x20140001	addi $20,$0,1	9: addi $s4, $zero, 1 #step = 1
	0x00400014	0x20150000	addi $21,$0,0	10: addi $s5, $zero, 0 #sum = 0
	0x00400018	0x0233502a	slt $10,$17,$19	11: loop: slt $t2, $s1, $s3 # $t2 = i < n? 1 : 0
	0x0040001c	0x11400007	beq $10,$0,7	12: beq $t2, $zero, endloop
	0x00400020	0x02314820	add $9,$17,$17	13: add $t1,$s1,$s1 #$t1=2*$s1
	0x00400024	0x01294820	add $9,$9,$9	14: add $t1,$t1,$t1 #$t1=4*$s1
	0x00400028	0x01324820	add $9,$9,$18	15: add $t1,$t1,$s2 #$t1 store the address of A[i]
	0x0040002c	0x8d280000	lw $8,0($9)	16: lw $t0,0($t1) #load value of A[i] in $t0
	0x00400030	0x02a8a820	add $21,$21,$8	17: add $s5,$s5,$t0 #sum=sum+A[i]
	0x00400034	0x02348820	add $17,$17,$20	18: add $s1,$s1,$s4 #i=i+step
	0x00400038	0x08100006	j 0x00400018	19: j loop #goto loop

Khởi tạo bức nhảy step

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x20110000	addi $17,$0,0	6: addi $s1, $zero, 0 #i = 0
	0x00400004	0x3c011001	lui $1,4097	7: la $s2, arr #load address A[i]
	0x00400008	0x34320000	ori $18,$1,0	
	0x0040000c	0x20130006	addi $19,$0,6	8: addi $s3, $zero, 6 #n = 6
	0x00400010	0x20140001	addi $20,$0,1	9: addi $s4, $zero, 1 #step = 1
	0x00400014	0x20150000	addi $21,$0,0	10: addi $s5, $zero, 0 #sum = 0
	0x00400018	0x0233502a	slt $10,$17,$19	11: loop: slt $t2, $s1, $s3 # $t2 = i < n? 1 : 0
	0x0040001c	0x11400007	beq $10,$0,7	12: beq $t2, $zero, endloop
	0x00400020	0x02314820	add $9,$17,$17	13: add $t1,$s1,$s1 #$t1=2*$s1
	0x00400024	0x01294820	add $9,$9,$9	14: add $t1,$t1,$t1 #$t1=4*$s1
	0x00400028	0x01324820	add $9,$9,$18	15: add $t1,$t1,$s2 #$t1 store the address of A[i]
	0x0040002c	0x8d280000	lw $8,0($9)	16: lw $t0,0($t1) #load value of A[i] in $t0
	0x00400030	0x02a8a820	add $21,$21,$8	17: add $s5,$s5,$t0 #sum=sum+A[i]
	0x00400034	0x02348820	add $17,$17,$20	18: add $s1,$s1,$s4 #i=i+step
	0x00400038	0x08100006	j 0x00400018	19: j loop #goto loop

Khởi tạo tổng sum

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x20110000	addi $17,$0,0	6: addi $s1, $zero, 0 #i = 0
	0x00400004	0x3c011001	lui $1,4097	7: la $s2, arr #load address A[i]
	0x00400008	0x34320000	ori $18,$1,0	
	0x0040000c	0x20130006	addi $19,$0,6	8: addi $s3, $zero, 6 #n = 6
	0x00400010	0x20140001	addi $20,$0,1	9: addi $s4, $zero, 1 #step = 1
	0x00400014	0x20150000	addi $21,$0,0	10: addi $s5, $zero, 0 #sum = 0
	0x00400018	0x0233502a	slt $10,$17,$19	11: loop: slt $t2, $s1, $s3 # $t2 = i < n? 1 : 0
	0x0040001c	0x11400007	beq $10,$0,7	12: beq $t2, $zero, endloop
	0x00400020	0x02314820	add $9,$17,$17	13: add $t1,$s1,$s1 #$t1=2*$s1
	0x00400024	0x01294820	add $9,$9,$9	14: add $t1,$t1,$t1 #$t1=4*$s1
	0x00400028	0x01324820	add $9,$9,$18	15: add $t1,$t1,$s2 #$t1 store the address of A[i]
	0x0040002c	0x8d280000	lw $8,0($9)	16: lw $t0,0($t1) #load value of A[i] in $t0
	0x00400030	0x02a8a820	add $21,$21,$8	17: add $s5,$s5,$t0 #sum=sum+A[i]
	0x00400034	0x02348820	add $17,$17,$20	18: add $s1,$s1,$s4 #i=i+step
	0x00400038	0x08100006	j 0x00400018	19: j loop #goto loop

Vòng lặp 1: sum = 0 + Arr[0] <=> sum = 0 + 26 = 26

Vòng lặp 2: sum = 26 + Arr[1] <=> sum = 26 + 7 = 33

Vòng lặp 3: sum = 33 + Arr[2] <=> sum = 33 + 2001 = 2034

Vòng lặp 4: sum = 2034 + Arr[3] <=> sum = 2034 + 38 = 2072

Vòng lặp 5: sum = 2072 + Arr[4] <=> sum = 2072 + 42 = 2114

Vòng lặp 6: sum = 2114 + Arr[5] <=> sum = 2114 + -73 = 2041

Kết quả:

Registers | Coproc 1 | Coproc 0

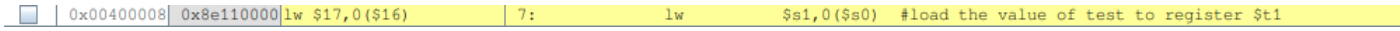
Name	Number	Value
$zero	0	0
$at	1	268500992
$v0	2	0
$v1	3	0
$a0	4	0
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	-73
$t1	9	268501012
$t2	10	0
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	6
$s2	18	268500992
$s3	19	6
$s4	20	1
$s5	21	2041
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194364
hi		0
lo		0

Bài 3:

Lấy địa chỉ của biến

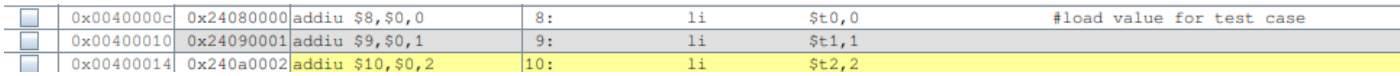
0x00400000	0x3c011001	lui $1,4097	6:	la	$s0,test	#load the address of test variable
0x00400004	0x34300000	ori $16,$1,0				

Lấy giá trị của biến



0x00400008	0x8e110000	lw $17,0($16)	7:	lw	$s1,0($s0)	#load the value of test to register $t1

Lấy giá trị của test case



0x0040000c	0x24080000	addiu $8,$0,0	8:	li	$t0,0	#load value for test case
0x00400010	0x24090001	addiu $9,$0,1	9:	li	$t1,1	
0x00400014	0x240a0002	addiu $10,$0,2	10:	li	$t2,2	

Câu lệnh rẽ nhánh

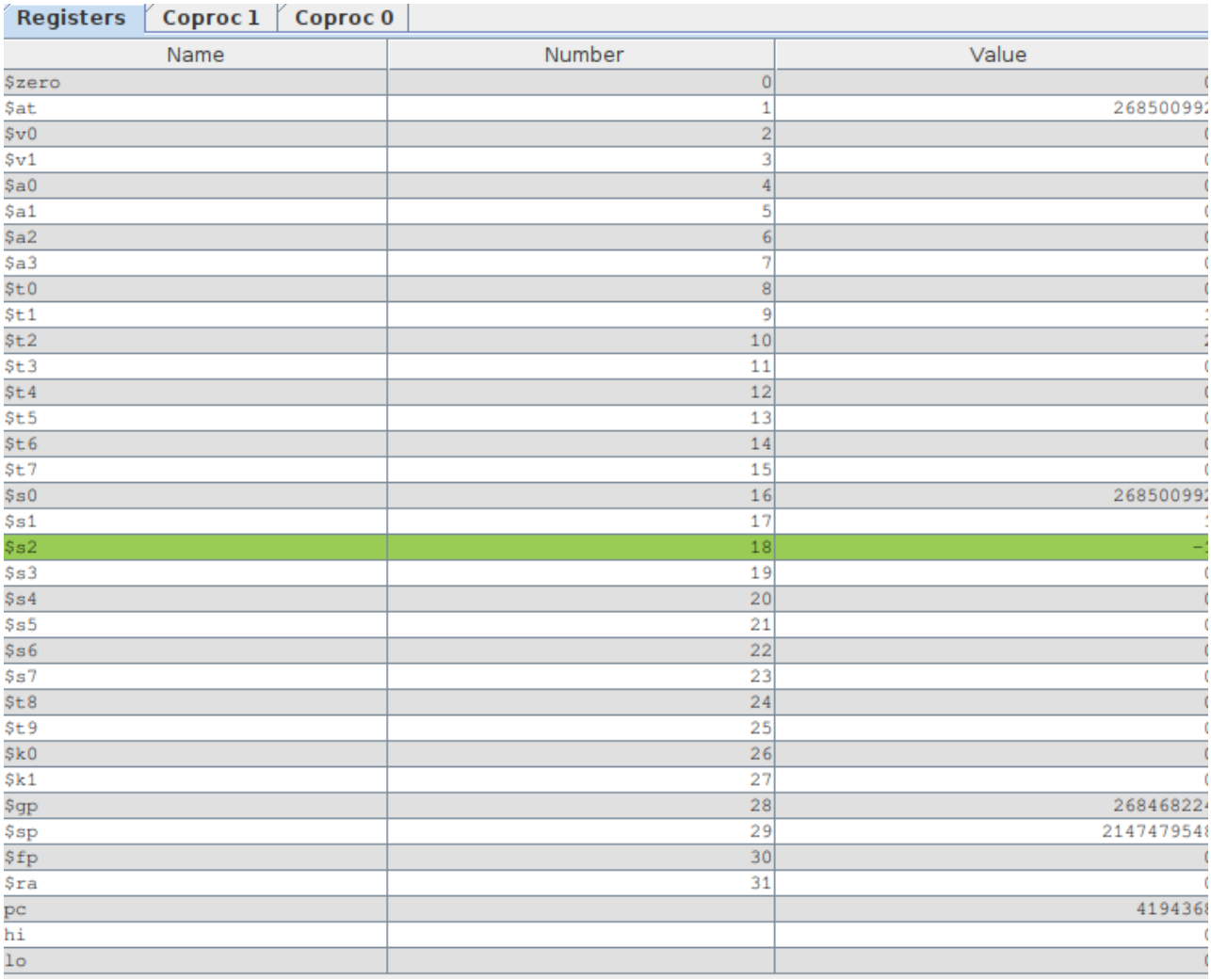
0x00400018	0x12280003	beq $17,$8,3	11:	beq	$s1,$t0,case_0
0x0040001c	0x12290004	beq $17,$9,4	12:	beq	$s1,$t1,case_1
0x00400020	0x122a0005	beq $17,$10,5	13:	beq	$s1,$t2,case_2

Với test = 0, kết quả:

Registers | Coproc 1 | Coproc 0

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	0
$s2	18	1
$s3	19	0
$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	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194368
hi		0
lo		0

Với test = 1, kết quả:



Registers | Coproc 1 | Coproc 0

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	0
$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	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194368
hi		0
lo		0

Với test = 2, kết quả:

Registers | Coproc 1 | Coproc 0

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	2
$s2	18	0
$s3	19	0
$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	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194368
hi		0
lo		0

Bài 4:

a. i < j

```
.text
    addi  $s1, $zero, 6    # i = 6
    addi  $s2, $zero, 5     # j = 5
    addi  $t1, $zero, 1     # x = 1
    addi  $t2, $zero, 2     # y = 2
    add   $t3, $zero, 3     # z = 3
    start:
        slt   $t0, $s1, $s2      # i < j
```

```
        bnez  $t0, else          # branch to else if i >= j
        addi  $t1, $t1, 1        # then part: x = x + 1
        addi  $t3, $zero, 1      # z = 1
        j     endif              # skip else part
    else:                        # begin else part
        addi  $t2, $t2, -1       # y = y - 1
        add   $t3, $t3, $t3        # z = 2 * z
    endif:
```

b. i >= j

```
.text
    addi  $s1, $zero, 6     # i = 6
    addi  $s2, $zero, 5     # j = 5
    addi  $t1, $zero, 1     # x = 1
    addi  $t2, $zero, 2     # y = 2
    add   $t3, $zero, 3     # z = 3
    start:
        sge   $t0, $s1, $s2      # i >= j
        bnez  $t0, else          # branch to else if i < j
        addi  $t1, $t1, 1        # then part: x = x + 1
        addi  $t3, $zero, 1      # z = 1
        j     endif              # skip else part
    else:                        # begin else part
        addi  $t2, $t2, -1       # y = y - 1
```

```
        add   $t3, $t3, $t3        # z = 2 * z
    endif:
```

c. i + j <= 0

```
.text
    addi  $s1, $zero, 6     # i = 6
    addi  $s2, $zero, 5     # j = 5
    addi  $t1, $zero, 1     # x = 1
    addi  $t2, $zero, 2     # y = 2
    add   $t3, $zero, 3     # z = 3
    add   $t4, $s1, $s2  # i + j
    start:
        sle   $t0, $t4, $0      # i+j <= 0
        bnez  $t0, else          # branch to else if i + j > 0
        addi  $t1, $t1, 1        # then part: x = x + 1
        addi  $t3, $zero, 1      # z = 1
        j     endif              # skip else part
    else:                        # begin else part
        addi  $t2, $t2, -1       # y = y - 1
        add   $t3, $t3, $t3        # z = 2 * z
    endif:
```

d. i + j > m + n

```
.text
    addi  $s1, $zero, 6     # i = 6
```

```
    addi  $s2, $zero, 5     # j = 5
    addi  $s3, $zero, 7     # m = 7
    addi  $s4, $zero, 8     # n = 8
    addi  $t1, $zero, 1     # x = 1
    addi  $t2, $zero, 2     # y = 2
    add   $t3, $zero, 3     # z = 3
    add   $t4, $s1, $s2  # i + j
    add   $t5, $s3, $s4  # m + n
    start:
        sgt   $t0, $t4, $t5      # i+j > m + n
        bnez  $t0, else          # branch to else if i + j <= m
+ n
        addi  $t1, $t1, 1        # then part: x = x + 1
        addi  $t3, $zero, 1      # z = 1
        j     endif              # skip else part
    else:                        # begin else part
        addi  $t2, $t2, -1       # y = y - 1
        add   $t3, $t3, $t3        # z = 2 * z
    endif:
```

Bài 5:

a. i < n

```
.data
arr: .word       26,7,2001,38,42,-73
```

```
.text
addi $s1, $zero, 0#i = 0
la   $s2, arr    #load address A[i]
addi $s3, $zero, 6#n = 6
addi $s4, $zero, 1#step = 1
addi $s5, $zero, 0#sum = 0
loop:       slt  $t2, $s1, $s3   # $t2 = i < n? 1 : 0
beq  $t2, $zero, endloop
add  $t1,$s1,$s1  #$t1=2*$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 #i=i+step
j    loop          #goto loop
Endloop:
b.
.data
arr: .word       26,7,2001,38,42,-73

.text
addi $s1, $zero, 0#i = 0
```

```
la   $s2, arr    #load address A[i]
addi $s3, $zero, 6#n = 6
addi $s4, $zero, 1#step = 1
addi $s5, $zero, 0#sum = 0
loop:       sle  $t2, $s1, $s3  # $t2 = i <= n? 1 : 0
beq  $t2, $zero, endloop
add  $t1,$s1,$s1  #$t1=2*$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 #i=i+step
j    loop          #goto loop
endloop:
c.
.data
arr: .word       26,7,2001,38,42,-73

.text
addi $s1, $zero, 0#i = 0
la   $s2, arr    #load address A[i]
addi $s3, $zero, 6#n = 6
addi $s4, $zero, 1#step = 1
```

```
addi $s5, $zero, 0#sum = 0
loop:       sle  $t2, $s5, $0    # $t2 = sum <= 0? 1 : 0
beq  $t2, $zero, endloop
add  $t1,$s1,$s1  #$t1=2*$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 #i=i+step
j    loop          #goto loop
endloop:
```

d.

```
.data

arr: .word       26,0,2001,38,42,-73

.text

addi $s1, $zero, 0#i = 0
la   $s2, arr    #load address A[i]
addi $s3, $zero, 6#n = 6
addi $s4, $zero, 1#step = 1
addi $s5, $zero, 0#sum = 0
```

```
loop:
add  $t1,$s1,$s1  #$t1=2*$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
beq  $t0,$0,endloop    #if A[i] == 0 then endloop
add  $s5,$s5,$t0  #sum=sum+A[i]
add  $s1,$s1,$s4 #i=i+step
j    loop          #goto loop

endloop:
```

Bài 6:

```
.data
    arr: .word 26, 7, -2001, -42, -73, -48, -500, 499, 123, 145
.text
    addi  $s1, $zero, 0    # i = 0
    la    $s2, arr         # load address A[i]
    addi  $s3, $zero, 10   # n = 10
    addi  $s4, $zero, 1    # step = 1
    addi  $s5, $zero, 0    # max = 0
```

```
   loop:
        add  $s1, $s1, $s4    # i = i + step
        add  $t1,$s1,$s1  #$t1=2*$s1
add  $t1,$t1,$t1      #$t1=4*$s1
        add  $t1, $t1, $s2    # t1 store address of A[i]
        lw   $t0, 0($t1)      # load value of A[i] in $t0
        abs  $t2, $t0
        start:
            sgt  $t3, $t2, $s5    # if A[i] > max
            beqz $t3, else        # if not then else
            add  $s5, $t2, $zero  # max = A[i]
            j endif
        else:
        endif:
            blt  $s1, $s3, loop   # i <= n -> loop
```