

Laboratory 3 Load/ Store , Jump & Branch instructions

Bài 1:

Code

```

B1.asm*
1  #Laboratory Exercise 3, Home Assignment 1
2  addi $s1,$0,2020 #i
3  addi $s2,$0,440 #j
4  start:
5  slt $t0,$s2,$s1 # j<i
6  bne $t0,$zero,else # branch to else if j<i
7  addi $t1,$t1,1 # then part: x=x+1
8  addi $t3,$zero,1 # z=1
9  j endif # skip "else" part
10 else: addi $t2,$t2,-1 # begin else part: y=y-1
11 add $t3,$t3,$t3 # z=2*z
12 endif:
13

```

Run

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	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
\$ep	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194304
hi		0
lo		0

Text Segment					
Bkpt	Address	Code	Basic	Source	
	0x00400000	0x201107e4	addi \$t1,\$0,2020	2:	addi \$s1,\$0,2020 #i
	0x00400004	0x201201b8	addi \$t3,\$0,440	3:	addi \$s2,\$0,440 #j
	0x00400008	0x0251402a	slt \$t0,\$t3,\$t1	5:	slt \$t0,\$s2,\$s1 # j<i
	0x0040000c	0x15000003	bne \$t0,\$0,3	6:	bne \$t0,\$zero,else # branch to else if j<i
	0x00400010	0x21290001	addi \$t1,\$t1,1	7:	addi \$t1,\$t1,1 # then part: x=x+1
	0x00400014	0x200b0001	addi \$t3,\$0,1	8:	addi \$t3,\$zero,1 # z=1
	0x00400018	0x08100009	j 0x00400024	9:	j endif # skip "else" part
	0x0040001c	0x214affff	addi \$t2,\$t2,-1	10:	else: addi \$t2,\$t2,-1 # begin else part:
	0x00400020	0x016b5820	add \$t3,\$t3,\$t3	11:	add \$t3,\$t3,\$t3 # z=2*z

Data Segment					
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)
0x10010000	0	0	0	0	0
0x10010020	0	0	0	0	0
0x10010040	0	0	0	0	0
0x10010060	0	0	0	0	0
0x10010080	0	0	0	0	0
0x100100a0	0	0	0	0	0
0x100100c0	0	0	0	0	0
0x100100e0	0	0	0	0	0
0x10010100	0	0	0	0	0
0x10010120	0	0	0	0	0

0x10010000 (.data) ☒ Hexadecimal Addresses ☐ Hexa

Mars Messages Run I/O

Hoàng Minh Ngọc 20200440

2 câu lệnh đầu tiên ta khởi tạo giá trị cho i và j

Và lưu vào 2 biến s1, s2

Bắt đầu vào chương trình ta chạy lệnh `slt $t0,$s2,$s1`

Ta so sánh 2 số s1 và s2 với nhau

ở lệnh `slt` ta được hiểu là $s2 < s1$ thì $t0 = 1$ ngược lại là $t0$ bằng 0

(sau câu lệnh trên thì giá trị các biến như sau, $s1 = 2020$, $s2 = 440$, $t0 = 1$)

Ta mang $t0$ đi so sánh với số 0 để biết rõ nhanh

ở lệnh `bne $t0,$zero,else` có nghĩa là

nếu như $t0$ và zero bằng nhau thì sẽ thực hiện câu lệnh `else`

vậy khi hoàn thành xong câu lệnh này chương trình lập tức nhảy tới dòng có câu lệnh `else`

trong câu lệnh `else: addi $t2,$t2,-1`

ta thực hiện lệnh cộng $t2 = t2 + (-1)$

và ở đây $t2$ đang mang giá trị bằng 0 nên sau khi thực hiện câu lệnh thì $t2$ mang giá trị là -1

tiếp đến ta thực hiện dòng lệnh `add $t3,$t3,$t3 # z=2*z`

ta cộng $t3 = t3 + t3 = 0$ vì $t3$ ở đây ta chưa khởi tạo giá trị cho nó

kết thúc chương trình

Hoàng Minh Ngọc 20200440

Bài 2:

Code

```

b21.asm*  B1.asm
1  #Laboratory 3, Home Assignment 2
2
3  .data
4  A: .word 1,2,3,4,2020,2021
5
6  .text
7  addi $s1, $0, -1 #khởi tạo giá trị i
8  addi $s3, $0, 5 # khởi tạo giá n
9  addi $s4, $0, 1 #step
10 addi $s5, $0, 0 #SUM
11 la $s2, A
12 loop: add $s1,$s1,$s4 #i=i+step
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 bne $s1,$s3,loop #if i != n, goto loop

```

Run

Text Segment				
Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x2011ffff	addi \$17,\$0,-1	7: addi \$s1, \$0, -1 #khởi tạo giá trị i
<input type="checkbox"/>	0x00400004	0x20130005	addi \$19,\$0,5	8: addi \$s3, \$0, 5 # khởi tạo giá n
<input type="checkbox"/>	0x00400008	0x20140001	addi \$20,\$0,1	9: addi \$s4, \$0, 1 #step
<input type="checkbox"/>	0x0040000c	0x20150000	addi \$21,\$0,0	10: addi \$s5, \$0, 0 #SUM
<input type="checkbox"/>	0x00400010	0x3c011001	lui \$1,4097	11: la \$s2, A
<input type="checkbox"/>	0x00400014	0x34320000	ori \$18,\$1,0	
<input type="checkbox"/>	0x00400018	0x02348820	add \$17,\$17,\$20	12: loop: add \$s1,\$s1,\$s4 #i=i+step
<input type="checkbox"/>	0x0040001c	0x02314820	add \$9,\$17,\$17	13: add \$t1,\$s1,\$s1 #t1=2*s1
<input type="checkbox"/>	0x00400020	0x01294820	add \$9,\$9,\$9	14: add \$t1,\$t1,\$t1 #t1=4*s1
<input type="checkbox"/>	0x00400024	0x01324820	add \$9,\$9,\$18	15: add \$t1,\$t1,\$s2 #t1 store the address of A[i]
<input type="checkbox"/>	0x00400028	0x8d280000	lw \$8,0(\$9)	16: lw \$t0,0(\$t1) #load value of A[i] in \$t0
<input type="checkbox"/>	0x0040002c	0x02a8a820	add \$21,\$21,\$8	17: add \$s5,\$s5,\$t0 #sum=sum+A[i]
<input type="checkbox"/>	0x00400030	0x1633ffff	bne \$17,\$19,-7	18: bne \$s1,\$s3,loop #if i != n, goto loop

Đầu tiên ta khởi tạo giá trị cho $i = -1$ vì khi $+$ step thì giá trị của i sẽ bắt đầu bằng 0

Ta khởi tạo giá trị của $n = 5$ vì mảng A có 5 phần tử

Khởi tạo giá trị của step bằng 1 vì qua các lệnh giá trị nhảy lên 4 byte

Khởi tạo giá trị $SUM = 0$

Ta thực hiện la \$s2, A

Lấy địa chỉ của mảng A lưu vào trong biến s2

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

```
add $t1,$s1,$s1 #t1=2*s1
```

```
add $t1,$t1,$t1 #t1=4*s1
```

câu lệnh nhảy 4 byte liên tiếp

```
add $t1,$t1,$s2 #t1 store the address of A[i]
```

câu lệnh thực hiện cộng 4 byte vào địa chỉ gốc của A để lấy giá trị của phần tử thứ 2 (trường hợp đầu tiên do $i=0$ nên là sẽ là địa chỉ của A0)

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

```
bne $s1,$s3,loop #if i != n, goto loop
```

ta thực hiện lệnh cộng vào SUM và so sánh xem giá trị i đã đến cuối mảng hay chưa bằng cách so sánh với số phần tử n nếu như $i \neq n$ thì quay lại loop

còn nếu như $i=n$ thì chạy tiếp

và đây là lệnh cuối cùng cho nên sẽ kết thúc chương trình

Bài 3 :

Code

```

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

```

Run

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x24100001	addiu \$16,\$0,1	6: la \$s0, 1 #load the address of test variable
<input type="checkbox"/>	0x00400004	0x22110000	addi \$17,\$16,0	7: addi \$s1, \$s0, 0 #load the value of test to register \$t1
<input type="checkbox"/>	0x00400008	0x21080000	addi \$8,\$8,0	8: addi \$t0,\$t0, 0 #load value for test case
<input type="checkbox"/>	0x0040000c	0x21290001	addi \$9,\$9,1	9: addi \$t1,\$t1, 1
<input type="checkbox"/>	0x00400010	0x214a0002	addi \$10,\$10,2	10: addi \$t2, \$t2, 2
<input type="checkbox"/>	0x00400014	0x12280003	beq \$17,\$8,3	11: beq \$s1, \$t0, case 0
<input type="checkbox"/>	0x00400018	0x12290004	beq \$17,\$9,4	12: beq \$s1, \$t1, case 1
<input type="checkbox"/>	0x0040001c	0x122a0005	beq \$17,\$10,5	13: beq \$s1, \$t2, case 2
<input type="checkbox"/>	0x00400020	0x0810000f	j 0x00400003c	14: j default
<input type="checkbox"/>	0x00400024	0x22520001	addi \$18,\$18,1	15: case 0: addi \$s2, \$s2, 1 #a=a+1
<input type="checkbox"/>	0x00400028	0x0810000f	j 0x00400003c	16: j continue
<input type="checkbox"/>	0x0040002c	0x02499022	sub \$18,\$18,\$9	17: case 1: sub \$s2, \$s2, \$t1 #a=a-1
<input type="checkbox"/>	0x00400030	0x0810000f	j 0x00400003c	18: j continue
<input type="checkbox"/>	0x00400034	0x02739820	add \$19,\$19,\$19	19: case 2: add \$s3, \$s3, \$s3 #b=2*b
<input type="checkbox"/>	0x00400038	0x0810000f	j 0x00400003c	20: j continue

Đầu tiên ta cần khai báo tạo giá trị cho các biến đầu vào ở đây dòng số 6 ta lưu 1 vào biến s0

Lưu s1 = s0 để chuẩn bị biến so sánh

Dòng 8,9,10 ta tạo các case lựa chọn từ 0,1,2 để lát ta so sánh với s1

Dòng 11,12,13 ta bắt đầu rẽ nhánh

Sử dụng lệnh beq

```
beq $s1, $t0, case_0
```

```
beq $s1, $t1, case_1
```

```
beq $s1, $t2, case_2
```

nếu s1 mà bằng t0 thì ta đi đến case_0 tương tự như case 1 case 2

sau khi thực hiện xong ta chạy tới continue để tránh lặp lại các câu lệnh khác

ở đây e để giá trị s0 bằng 1 nên giá trị s1 = 1

sau khi chạy thì máy nhảy tới case 1 vì t1 = s1

và kết quả $s2 = s2 - 1 = -1$ (vì giá trị của s2 trước khi tính toán là bằng 0)

tới đây ta thực hiện câu lệnh “ j continue “ để thực hiện đến lệnh continue và kết thúc chương trình

câu lệnh dòng 14 “ j default “ để khi mà ko có giá trị thuộc trong khoảng từ 0 đến 2 thì ta thực hiện nhảy đến vị trí có câu lệnh continue dòng số 21,22

trước khi tới bài 4 em đã tính toán và phân tích sử dụng 3 lệnh slt, bne, beq để phân tích và thỏa mãn các điều kiện $>$, $<$, \geq , \leq

Ass

a.
u =
x =
y =

$a < b$
slt to, a, b (nếu $a < b$ thì to = 1 ngược lại to = 0).
bne to, 0, code. (nếu to \neq 0 thì goto code.)
ĐK sai
code: ĐK đúng

$a > b$
~~slt slt~~ ~~to, a, b~~ ~~to, b, a~~ (nếu $b < a$ thì to = 1 ngược lại to = 0.
nếu $a < b$ thì to = 0.)
bne to, 0, code.
ĐK sai
code: ĐK đúng

$a \geq b$
slt to, a, b (nếu $a < b$ thì to = 1 ngược lại to = 0)
beq to, 0, code.
ĐK sai
code: ĐK đúng

$a \leq b$
slt to, b, a (nếu $b < a$ thì to = 1.)
beq to, 0, code
ĐK sai
code ĐK đúng

Bài 4:

Trước khi vào ý A ta mặc định khai báo cho toàn bộ 4 ý là $i = 1, j = 2$,

Ý A

Code

```
addi $s1, $0, 1 #khởi tạo giá trị i
addi $s2, $0, 2 #khởi tạo giá trị của j

start:
# điều kiện nếu i < j thì thực hiện câu lệnh x=x+1, z=1, ngược lại y=y-1, z=2*z
slt $t0, $s1, $s2
bne $t0, $0, code
addi $t2, $t2, -1
add $t3, $t3, $t3
j endif
code: addi $t2, $t2, 1
      addi $t3, $0, 1
endif:
```

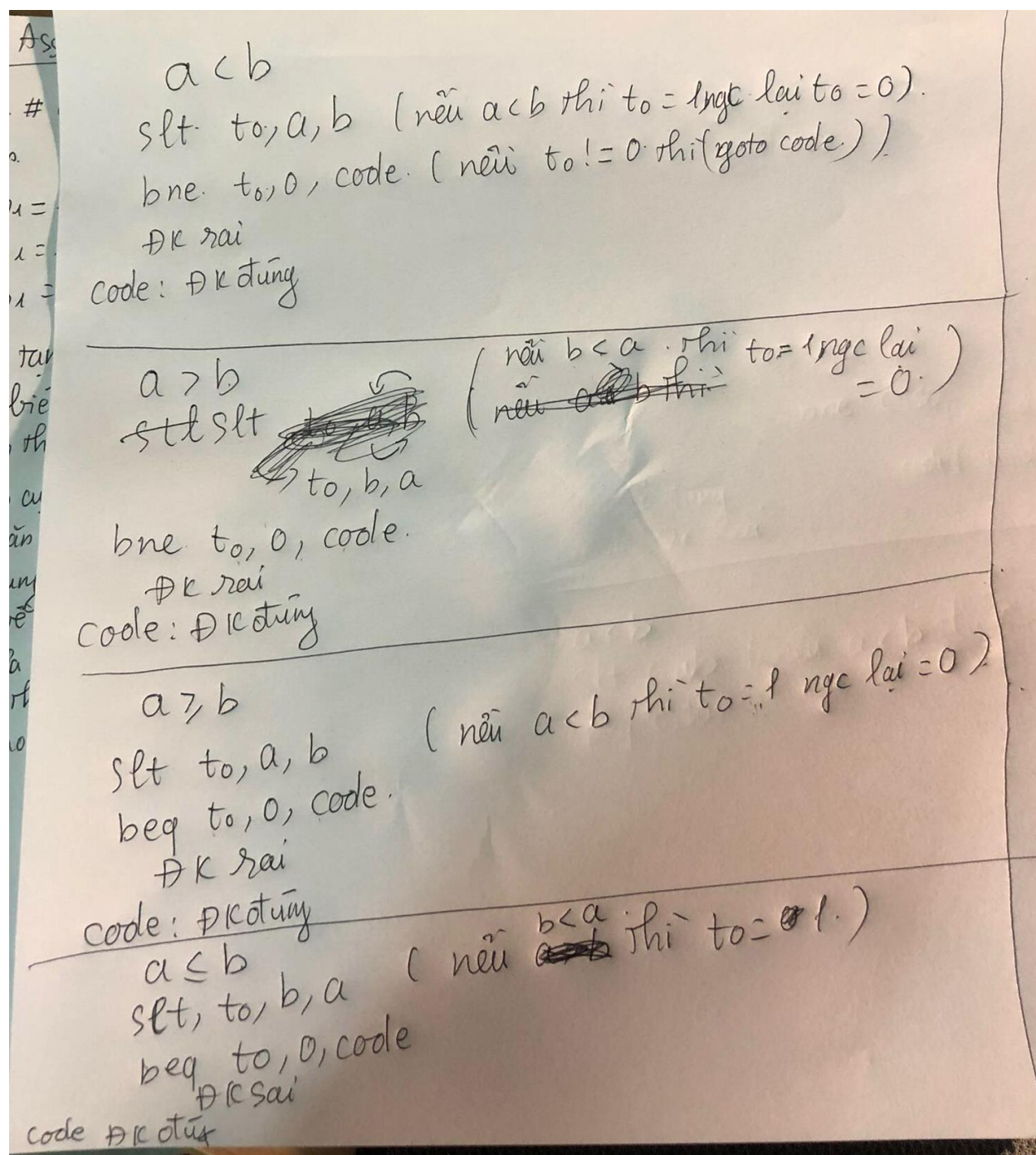
Run

Code	Basic	Source
0110001	addi \$17,\$0,1	1: addi \$s1, \$0, 1 #khởi tạo giá trị i
0120002	addi \$18,\$0,2	2: addi \$s2, \$0, 2 #khởi tạo giá trị của j
232402a	slt \$8,\$17,\$18	6: slt \$t0, \$s1, \$s2
5000003	bne \$8,\$0,3	7: bne \$t0, \$0, code
14affff	addi \$10,\$10,-1	8: addi \$t2, \$t2, -1
16b5820	add \$11,\$11,\$11	9: add \$t3, \$t3, \$t3
3100009	j 0x00400024	10: j endif
14a0001	addi \$10,\$10,1	11: code: addi \$t2, \$t2, 1
00b0001	addi \$11,\$0,1	12: addi \$t3, \$0, 1

Đầu tiên ta khai báo giá trị cho i và j

Tiếp theo ta so sánh xem các giá trị của i và j như thế nào để $t0 = 1$ hoặc bằng 0

Như công thức em đã lập ra ở trên (ôi trời để lập ra cái này em lú đã man :V mất cả tiếng để phân tích và ghép @@)



Vậy thì ta chỉ cần đổi lại biến số thì sẽ ra được kết quả

Tương tự như ý A thì ý B ta chỉ cần thay lại 2 câu lệnh `slt` và `beq`

Vậy ta có chương trình sau

Code

```
Bai4yA.asm  Bai4yB.asm*
1  addi $s1, $0, 1 #khởi tạo giá trị i
2  addi $s2, $0, 2 #khởi tạo giá trị của j
3
4  start:
5  # điều kiện nếu i>=j thì thực hiện câu lệnh x=x+1,z=1, ngược lại y=y-1,z=2*z
6  slt $t0, $s1, $s2
7  beq $t0, $0, code
8  addi $t2, $t2, -1
9  add $t3, $t3, $t3
10 j endif
11 code: addi $t2, $t2, 1
12       addi $t3, $0, 1
13 endif:
14
```

Run

Text Segment				
Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x20110001	addi \$17,\$0,1	1: addi \$s1, \$0, 1 #khởi tạo giá trị i
<input type="checkbox"/>	0x00400004	0x20120002	addi \$18,\$0,2	2: addi \$s2, \$0, 2 #khởi tạo giá trị của j
<input type="checkbox"/>	0x00400008	0x0232402a	slt \$8,\$17,\$18	6: slt \$t0, \$s1, \$s2
<input type="checkbox"/>	0x0040000c	0x11000003	beq \$8,\$0,3	7: beq \$t0, \$0, code
<input type="checkbox"/>	0x00400010	0x214affff	addi \$10,\$10,-1	8: addi \$t2, \$t2, -1
<input type="checkbox"/>	0x00400014	0x016b5820	add \$11,\$11,\$11	9: add \$t3, \$t3, \$t3
<input type="checkbox"/>	0x00400018	0x08100009	j 0x00400024	10: j endif
<input type="checkbox"/>	0x0040001c	0x214a0001	addi \$10,\$10,1	11: code: addi \$t2, \$t2, 1
<input type="checkbox"/>	0x00400020	0x200b0001	addi \$11,\$0,1	12: addi \$t3, \$0, 1

Đối với ý C ta cần 1 biến trung gian gộp 2 giá trị của `i` và `j` lại

Và ta mang biến tg đi so sánh `<=` với 0

Vậy thì bài toán sẽ chuyển về bài toán so sánh tương tự như trên dùng `slt` và `beq`

Vậy ta có chương trình sau

Code

```

Bai4yA.asm  Bai4yB.asm*  B4yC.asm
1  addi $s1, $0, 1 #khởi tạo giá trị i
2  addi $s2, $0, 2 #khởi tạo giá trị của j
3  add $s6, $s1, $s2 #tạo ra biến trung gian
4  start:
5  # điều kiện nếu i+j<=0 thực hiện câu lệnh x=x+1,z=1, ngược lại y=y-1,z=2*z
6  slt $t0, $0, $s6
7  beq $t0, $0, code
8  addi $t2, $t2, -1
9  add $t3, $t3, $t3
10 j endif
11 code: addi $t2, $t2, 1
12       addi $t3, $0, 1
13 endif:
14

```

Run

Text Segment				
Bkpt	Address	Code	Basic	Sc
<input type="checkbox"/>	0x00400000	0x20110001	addi \$17,\$0,1	1: addi \$s1, \$0, 1 #khởi tạo giá trị i
<input type="checkbox"/>	0x00400004	0x20120002	addi \$18,\$0,2	2: addi \$s2, \$0, 2 #khởi tạo giá trị của j
<input type="checkbox"/>	0x00400008	0x0232b020	add \$22,\$17,\$18	3: add \$s6, \$s1, \$s2 #tạo ra biến trung gian
<input type="checkbox"/>	0x0040000c	0x0016402a	slt \$8,\$0,\$22	6: slt \$t0, \$0, \$s6
<input type="checkbox"/>	0x00400010	0x11000003	beq \$8,\$0,3	7: beq \$t0, \$0, code
<input type="checkbox"/>	0x00400014	0x214affff	addi \$10,\$10,-1	8: addi \$t2, \$t2, -1
<input type="checkbox"/>	0x00400018	0x016b5820	add \$11,\$11,\$11	9: add \$t3, \$t3, \$t3
<input type="checkbox"/>	0x0040001c	0x0810000a	j 0x00400028	10: j endif
<input type="checkbox"/>	0x00400020	0x214a0001	addi \$10,\$10,1	11: code: addi \$t2, \$t2, 1
<input type="checkbox"/>	0x00400024	0x200b0001	addi \$11,\$0,1	12: addi \$t3, \$0, 1

Tương tự ý D ta cộng về trái vào tg1 về phải vào tg2 từ đó dùng câu lệnh slt, bne để so sánh với nhau

Code

```

1  addi $s1, $0, 1 #khởi tạo giá trị i
2  addi $s2, $0, 2 #khởi tạo giá trị của j
3  add $s6, $s1, $s2 #tạo ra biến trung gian thứ nhất
4
5  addi $s3, $0, 3 #khởi tạo giá trị m
6  addi $s4, $0, 4 #khởi tạo giá trị của n
7  add $s5, $s3, $s4 #tạo ra biến trung gian thứ hai
8
9  start:
0  # điều kiện nếu i+j<=m+n thực hiện câu lệnh x=x+1,z=1, ngược lại y=y-1,z=2*z
1  slt $t0, $5, $s6
2  beq $t0, $0, code
3  addi $t2, $t2, -1
4  add $t3, $t3, $t3
5  j endif
6  code:  addi $t2, $t2, 1
7         addi $t3, $0, 1
8  endif:
9

```

Run

Text Segment				
Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x20110001	addi \$17,\$0,1	1: addi \$s1, \$0, 1 #khởi tạo giá trị i
<input type="checkbox"/>	0x00400004	0x20120002	addi \$18,\$0,2	2: addi \$s2, \$0, 2 #khởi tạo giá trị của j
<input type="checkbox"/>	0x00400008	0x0232b020	add \$22,\$17,\$18	3: add \$s6, \$s1, \$s2 #tạo ra biến trung gian thứ nhất
<input type="checkbox"/>	0x0040000c	0x20130003	addi \$19,\$0,3	5: addi \$s3, \$0, 3 #khởi tạo giá trị m
<input type="checkbox"/>	0x00400010	0x20140004	addi \$20,\$0,4	6: addi \$s4, \$0, 4 #khởi tạo giá trị của n
<input type="checkbox"/>	0x00400014	0x0274a820	add \$21,\$19,\$20	7: add \$s5, \$s3, \$s4 #tạo ra biến trung gian thứ hai
<input type="checkbox"/>	0x00400018	0x00b6402a	slt \$8,\$5,\$22	11: slt \$t0, \$5, \$s6
<input type="checkbox"/>	0x0040001c	0x11000003	beq \$8,\$0,3	12: beq \$t0, \$0, code
<input type="checkbox"/>	0x00400020	0x214affff	addi \$10,\$10,-1	13: addi \$t2, \$t2, -1
<input type="checkbox"/>	0x00400024	0x016b5820	add \$11,\$11,\$11	14: add \$t3, \$t3, \$t3
<input type="checkbox"/>	0x00400028	0x0810000d	j 0x00400034	15: j endif
<input type="checkbox"/>	0x0040002c	0x214a0001	addi \$10,\$10,1	16: code: addi \$t2, \$t2, 1
<input type="checkbox"/>	0x00400030	0x200b0001	addi \$11,\$0,1	17: addi \$t3, \$0, 1

Do các điều kiện đúng và sai ở nguyên vị trí vậy nên ta chỉ cần đổi bộ 3 lệnh slt, bne, beq như trên hình là được

Bài 5: ở hình trên ta có thể mô tả lại các lệnh khi sử dụng câu lệnh so sánh

Với ý A ta chỉ cần gán lại i và n và đặt vào đó câu lệnh slt

Từ đó ta có chương trình sau

Code

```
1
2  .data
3  A: .word 1,2,3,4,5,6,10,20,25,36,187
4
5  .text
6  addi $s1, $0, -1 #khởi tạo giá trị i
7  addi $s3, $0, 8  #khởi tạo giá trị của n
8  addi $s4, $0, 1  #bước nhảy
9  addi $s5, $0, 0  #SUM
10 la $s2, A
11
12 loop: add $s1, $s1, $s4 #i=i+step
13 add $t1, $s1, $s1 #t1=2*s1
14 add $t1, $t1, $t1 #t1=4*s1
15 add $t1, $t1, $s2 #nhận địa chỉ của A[i]
16 lw $t0,0($t1) #nhận giá trị của A[i]
17 add $s5, $s5, $t0 # sum = sum + A[i]
18 slt $t6, $s1,$s3 # nếu i<n thì t6=1 ngược lại =0
19 bne $s1, $s3, loop # nếu s1=1 thì chạy tới loop
20
```

Run

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x2011ffff	addi \$17,\$0,-1	6: addi \$s1, \$0, -1 #khởi tạo giá trị i
<input type="checkbox"/>	0x00400004	0x20130008	addi \$19,\$0,8	7: addi \$s3, \$0, 8 #khởi tạo giá trị của n
<input type="checkbox"/>	0x00400008	0x20140001	addi \$20,\$0,1	8: addi \$s4, \$0, 1 #bước nhảy
<input type="checkbox"/>	0x0040000c	0x20150000	addi \$21,\$0,0	9: addi \$s5, \$0, 0 #SUM
<input type="checkbox"/>	0x00400010	0x3c011001	lui \$1,4097	10: la \$s2, A
<input type="checkbox"/>	0x00400014	0x34320000	ori \$18,\$1,0	
<input type="checkbox"/>	0x00400018	0x02348820	add \$17,\$17,\$20	12: loop: add \$s1, \$s1, \$s4 #i=i+step
<input type="checkbox"/>	0x0040001c	0x02314820	add \$9,\$17,\$17	13: add \$t1, \$s1, \$s1 #t1=2*s1
<input type="checkbox"/>	0x00400020	0x01294820	add \$9,\$9,\$9	14: add \$t1, \$t1, \$t1 #t1=4*s1
<input type="checkbox"/>	0x00400024	0x01324820	add \$9,\$9,\$18	15: add \$t1, \$t1, \$s2 #nhận địa chỉ của A[i]
<input type="checkbox"/>	0x00400028	0x8d280000	lw \$8,0(\$9)	16: lw \$t0,0(\$t1) #nhận giá trị của A[i]
<input type="checkbox"/>	0x0040002c	0x02a8a820	add \$21,\$21,\$8	17: add \$s5, \$s5, \$t0 # sum = sum + A[i]
<input type="checkbox"/>	0x00400030	0x0233702a	slt \$14,\$17,\$19	18: slt \$t6, \$s1,\$s3 # nếu i<n thì t6=1 ngược lại =0
<input type="checkbox"/>	0x00400034	0x1633ffff	bne \$17,\$19,-8	19: bne \$s1, \$s3, loop # nếu s1=1 thì chạy tới loop

Đầu tiên tương tự như bài 2, ta khai báo và khởi tạo các giá trị cho I, n, mảng A, step và tổng

Ta so sánh $I < n$ nếu như $i < n$ là đúng thì $t6 = 1$

Lúc này ta dùng câu lệnh bne để điều khiển vòng lặp

Tương tự như ý B

Ta chỉ cần thay 2 dòng cuối bởi lệnh slt, beq như trên hình em đã phân tích

Code

```

1
2 .data
3 A: .word 1,2,3,4,5,6,10,20,25,36,187
4
5 .text
6 addi $s1, $0, -1 #khởi tạo giá trị i
7 addi $s3, $0, 8 #khởi tạo giá trị của n
8 addi $s4, $0, 1 #bước nhảy
9 addi $s5, $0, 0 #SUM
0 la $s2, A
1
2 loop: add $s1, $s1, $s4 #i=i+step
3 add $t1, $s1, $s1 #t1=2*s1
4 add $t1, $t1, $t1 #t1=4*s1
5 add $t1, $t1, $s2 #nhận địa chỉ của A[i]
6 lw $t0,0($t1) #nhận giá trị của A[i]
7 add $s5, $s5, $t0 # sum = sum + A[i]
8
9 slt $t6, $s3, $s1 # nếu i<=n thì t6=1 ngược lại =0
0 beq $t6, 0, loop # nếu s1=1 thì chạy tới loop
1

```

Run

Text Segment				
Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x2011ffff	addi \$t1,\$0,-1	6: addi \$s1, \$0, -1 #khởi tạo giá trị i
<input type="checkbox"/>	0x00400004	0x20130008	addi \$t1,\$0,8	7: addi \$s3, \$0, 8 #khởi tạo giá trị của n
<input type="checkbox"/>	0x00400008	0x20140001	addi \$t1,\$0,1	8: addi \$s4, \$0, 1 #bước nhảy
<input type="checkbox"/>	0x0040000c	0x20150000	addi \$t1,\$0,0	9: addi \$s5, \$0, 0 #SUM
<input type="checkbox"/>	0x00400010	0x3c011001	lui \$1,4097	10: la \$s2, A
<input type="checkbox"/>	0x00400014	0x34320000	ori \$18,\$1,0	
<input type="checkbox"/>	0x00400018	0x02348820	add \$t1,\$t1,\$t1	12: loop: add \$s1, \$s1, \$s4 #i=i+step
<input type="checkbox"/>	0x0040001c	0x02314820	add \$t1,\$t1,\$t1	13: add \$t1, \$s1, \$s1 #t1=2*s1
<input type="checkbox"/>	0x00400020	0x01294820	add \$t1,\$t1,\$t1	14: add \$t1, \$t1, \$t1 #t1=4*s1
<input type="checkbox"/>	0x00400024	0x01324820	add \$t1,\$t1,\$t1	15: add \$t1, \$t1, \$s2 #nhận địa chỉ của A[i]
<input type="checkbox"/>	0x00400028	0x8d280000	lw \$t0,0(\$t1)	16: lw \$t0,0(\$t1) #nhận giá trị của A[i]
<input type="checkbox"/>	0x0040002c	0x02a8a820	add \$t1,\$t1,\$t1	17: add \$s5, \$s5, \$t0 # sum = sum + A[i]
<input type="checkbox"/>	0x00400030	0x0271702a	slt \$t6,\$t1,\$t1	19: slt \$t6, \$s3, \$s1 # nếu i<=n thì t6=1 ngược lại =0
<input type="checkbox"/>	0x00400034	0x20010000	addi \$t1,\$0,0	20: beq \$t6, 0, loop # nếu s1=1 thì chạy tới loop
<input type="checkbox"/>	0x00400038	0x102efff7	beq \$1,\$14,-9	

Với ý C ta chỉ cần mang sum đi so sánh với 0, tương tự như bài tính toán trên

Vậy ta có chương trình sau

Code

```

1
2 .data
3 A: .word 1,2,3,4,5,6,10,20,25,36,187
4 .text
5 addi $s1, $0, -1 #khởi tạo giá trị i
6 addi $s3, $0, 8 #khởi tạo giá trị của n
7 addi $s4, $0, 1 #bước nhảy
8 addi $s5, $0, 0 #SUM
9 la $s2, A
10
11 loop: add $s1, $s1, $s4 #i=i+step
12 add $t1, $s1, $s1 #t1=2*s1
13 add $t1, $t1, $t1 #t1=4*s1
14 add $t1, $t1, $s2 #nhận địa chỉ của A[i]
15 lw $t0,0($t1) #nhận giá trị của A[i]
16 add $s5, $s5, $t0 # sum = sum + A[i]
17
18 slt $t6, $s5, $0 # nếu SUM>=0 thì t6=1 ngược lại =0
19 beq $t6, 0, loop # nếu s1=1 thì chạy tới loop
20

```


Run

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x2011ffff	addi \$17,\$0,-1	5: addi \$s1, \$0, -1 #khởi tạo giá trị i
<input type="checkbox"/>	0x00400004	0x20130008	addi \$19,\$0,8	6: addi \$s3, \$0, 8 #khởi tạo giá trị của n
<input type="checkbox"/>	0x00400008	0x20140001	addi \$20,\$0,1	7: addi \$s4, \$0, 1 #bước nhảy
<input type="checkbox"/>	0x0040000c	0x20150000	addi \$21,\$0,0	8: addi \$s5, \$0, 0 #SUM
<input type="checkbox"/>	0x00400010	0x3c011001	lui \$1,4097	9: la \$s2, A
<input type="checkbox"/>	0x00400014	0x34320000	ori \$18,\$1,0	
<input type="checkbox"/>	0x00400018	0x02348820	add \$17,\$17,\$20	11: loop: add \$s1, \$s1, \$s4 #i=i+step
<input type="checkbox"/>	0x0040001c	0x02314820	add \$9,\$17,\$17	12: add \$t1, \$s1, \$s1 #t1=2*s1
<input type="checkbox"/>	0x00400020	0x01294820	add \$9,\$9,\$9	13: add \$t1, \$t1, \$t1 #t1=4*s1
<input type="checkbox"/>	0x00400024	0x01324820	add \$9,\$9,\$18	14: add \$t1, \$t1, \$s2 #nhận địa chỉ của A[i]
<input type="checkbox"/>	0x00400028	0x8d280000	lw \$8,0(\$9)	15: lw \$t0,0(\$t1) #nhận giá trị của A[i]
<input type="checkbox"/>	0x0040002c	0x02a8a820	add \$21,\$21,\$8	16: add \$s5, \$s5, \$t0 # sum = sum + A[i]
<input type="checkbox"/>	0x00400030	0x02a0702a	slt \$14,\$21,\$0	18: slt \$t6, \$s5, \$0 # nếu SUM>=0 thì t6=1 ngược lại =0
<input type="checkbox"/>	0x00400034	0x20010000	addi \$1,\$0,0	19: beq \$t6, 0, loop # nếu s1=1 thì chạy tới loop
<input type="checkbox"/>	0x00400038	0x102efff7	beq \$1,\$14,-9	

Đối với ý D thì tương tự như ý C

Ta mang $A[i]$ đi so sánh với 0

$A[i]$ chính là giá trị của lw t0

Đối với bài này ta chỉ cần sử dụng 1 câu lệnh duy nhất là nếu $A[i] == 0$ thì ta dừng chương trình

Ta có code sau

Code


```

.data
A: .word 1,2,3,4,5,6,10,20,25,36,187

.text

addi $s1, $0, -1 #khởi tạo giá trị i
addi $s3, $0, 8 #khởi tạo giá trị của n
addi $s4, $0, 1 #bước nhảy
addi $s5, $0, 0 #SUM
la $s2, A

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 #nhận địa chỉ của A[i]
lw $t0, 0($t1) #nhận giá trị của A[i]
add $s5, $s5, $t0 # sum = sum + A[i]

slt $t6, $s5, $0 # nếu SUM>=0 thì t6=1 ngược lại =0
bne $t6, 0, loop # nếu s1=1 thì chạy tới loop

```

Run

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x2011ffff	addi \$17,\$0,-1	5: addi \$s1, \$0, -1 #khởi tạo giá trị i
<input type="checkbox"/>	0x00400004	0x20130008	addi \$19,\$0,8	6: addi \$s3, \$0, 8 #khởi tạo giá trị của n
<input type="checkbox"/>	0x00400008	0x20140001	addi \$20,\$0,1	7: addi \$s4, \$0, 1 #bước nhảy
<input type="checkbox"/>	0x0040000c	0x20150000	addi \$21,\$0,0	8: addi \$s5, \$0, 0 #SUM
<input type="checkbox"/>	0x00400010	0x3c011001	lui \$1,4097	9: la \$s2, A
<input type="checkbox"/>	0x00400014	0x34320000	ori \$18,\$1,0	
<input type="checkbox"/>	0x00400018	0x02348820	add \$17,\$17,\$20	11: loop: add \$s1, \$s1, \$s4 #i=i+step
<input type="checkbox"/>	0x0040001c	0x02314820	add \$9,\$17,\$17	12: add \$t1, \$s1, \$s1 #t1=2*s1
<input type="checkbox"/>	0x00400020	0x01294820	add \$9,\$9,\$9	13: add \$t1, \$t1, \$t1 #t1=4*s1
<input type="checkbox"/>	0x00400024	0x01324820	add \$9,\$9,\$18	14: add \$t1, \$t1, \$s2 #nhận địa chỉ của A[i]
<input type="checkbox"/>	0x00400028	0x8d280000	lw \$8,0(\$9)	15: lw \$t0,0(\$t1) #nhận giá trị của A[i]
<input type="checkbox"/>	0x0040002c	0x02a8a820	add \$21,\$21,\$8	16: add \$s5, \$s5, \$t0 # sum = sum + A[i]
<input type="checkbox"/>	0x00400030	0x02a0702a	slt \$14,\$21,\$0	18: slt \$t6, \$s5, \$0 # nếu SUM>=0 thì t6=1 ngược lại =0
<input type="checkbox"/>	0x00400034	0x20010000	addi \$1,\$0,0	19: bne \$t6, 0, loop # nếu s1=1 thì chạy tới loop
<input type="checkbox"/>	0x00400038	0x142efff7	bne \$1,\$14,-9	

Đối với Bài 6 ta cần minh họa lại bài toán yêu cầu tìm phần tử có giá trị tuyệt đối lớn nhất trong 1 mảng đã cho và biết địa chỉ cơ sở của mảng đó

Vậy bài toán sẽ có biết cho số phần tử của 1 mảng, địa chỉ cơ sở

Vậy ta có 2 trường hợp có thể xảy ra

Đó là $A[i] < 0$ và $A[i] \geq 0$

Đối với $A[i] \geq 0$ ta mang đi so sánh với MAX

Còn với $A[i] < 0$ ta thực hiện lệnh mang đi đổi dấu của chúng rồi đem đi so sánh với MAX

Đối với biến đặc biệt MAX ra lưu giá trị của MAX chính là giá trị của địa chỉ cơ sở tương ứng với nó là $A[0]$

Và đến cuối cùng ta nhận được giá trị của MAX

Code

```
1
2 .data
3 A: .word 1,3,5,2,-6,-9,5,-4,2,8,-20,62,-198,53
4 .text
5 addi $s1, $0, -1 #khởi tạo giá trị i
6 addi $s3, $0, 14 #khởi tạo giá trị của n
7 addi $s4, $0, 1 #bước nhảy
8 la $s2, A
9 lw $s5, 0($s2) #giá trị MAX = A[0]
10 add $t7,$0,$0 # MAX khi khai báo có địa chỉ là A[0]
11 loop: add $s1, $s1, $s4 #i=i+step
12 add $t1, $s1, $s1 #t1=2*s1
13 add $t1, $t1, $t1 #t1=4*s1
14 add $t1, $t1, $s2 #nhận địa chỉ của A[i]
15 lw $s7,0($t1) #nhận giá trị của A[i]
16 slt $t4,$s7,$0 # câu lệnh so sánh A[i] với 0, nếu nhỏ hơn 0 ta đổi dấu còn nếu lớn hơn 0 thì ta chạy tới next
17 bne $t4,$0, code
18 add $s6, $s7, $0
19 j next
20 code:
21 sub $s6,$0,$s7
22 next:
23 slt $t4, $s5, $s6 # ta thực hiện lệnh so sánh MAX với A[i]
24 bne $t4, $0, kqdung # nếu như MAX<A[i] ta nhảy tới kết quả đúng, thực hiện lệnh gán giá trị MAX = A[i],
25 #nếu sai ta nhảy về kqsai để chạy tiếp chương trình
26 j kqsai
27 kqdung:
28 add $s5, $s6, $0
29 add $t7, $s1, $0
30 kqsai:
31 # ta thực hiện kiểm tra xem đã chạy hết phần tử chưa, nếu chưa chạy hết ta thực hiện quay lại vòng lặp loop
32 slt $t4, $s3,$s1
33 beq $t4, $0, loop
34
```

Run

Text Segment				
Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x2011ffff	addi \$17,\$0,-1	5: addi \$s1, \$0, -1 #khởi tạo giá trị i
<input type="checkbox"/>	0x00400004	0x2013000e	addi \$19,\$0,14	6: addi \$s3, \$0, 14 #khởi tạo giá trị của n
<input type="checkbox"/>	0x00400008	0x20140001	addi \$20,\$0,1	7: addi \$s4, \$0, 1 #bước nhảy
<input type="checkbox"/>	0x0040000c	0x3c011001	lui \$1,4097	8: la \$s2, A
<input type="checkbox"/>	0x00400010	0x34320000	ori \$18,\$1,0	
<input type="checkbox"/>	0x00400014	0x8e550000	lw \$21,0(\$18)	9: lw \$s5, 0(\$s2) #giá trị MAX = A[0]
<input type="checkbox"/>	0x00400018	0x02348820	add \$17,\$17,\$20	10: loop: add \$s1, \$s1, \$s4 #i=i+step
<input type="checkbox"/>	0x0040001c	0x02314820	add \$9,\$17,\$17	11: add \$t1, \$s1, \$s1 #t1=2*s1
<input type="checkbox"/>	0x00400020	0x01294820	add \$9,\$9,\$9	12: add \$t1, \$t1, \$t1 #t1=4*s1
<input type="checkbox"/>	0x00400024	0x01324820	add \$9,\$9,\$18	13: add \$t1, \$t1, \$s2 #nhận địa chỉ của A[i]
<input type="checkbox"/>	0x00400028	0x8d370000	lw \$23,0(\$9)	14: lw \$s7,0(\$t1) #nhận giá trị của A[i]
<input type="checkbox"/>	0x0040002c	0x02e0602a	slt \$12,\$23,\$0	15: slt \$t4,\$s7,\$0
<input type="checkbox"/>	0x00400030	0x15800002	bne \$12,\$0,2	16: bne \$t4,\$0, code
<input type="checkbox"/>	0x00400034	0x02e0b020	add \$22,\$23,\$0	17: add \$s6, \$s7, \$0
<input type="checkbox"/>	0x00400038	0x08100010	j 0x00400040	18: j next
<input type="checkbox"/>	0x0040003c	0x0017b022	sub \$22,\$0,\$23	20: sub \$s6,\$0,\$s7
<input type="checkbox"/>	0x00400040	0x02b6602a	slt \$12,\$21,\$22	22: slt \$t4, \$s5, \$s6
<input type="checkbox"/>	0x00400044	0x15800001	bne \$12,\$0,1	23: bne \$t4, \$0, kqdung
<input type="checkbox"/>	0x00400048	0x08100015	j 0x00400054	24: j kqsai
<input type="checkbox"/>	0x0040004c	0x02c0a820	add \$21,\$22,\$0	26: add \$s5, \$s6, \$0
<input type="checkbox"/>	0x00400050	0x02207820	add \$15,\$17,\$0	27: add \$t7, \$s1, \$0
<input type="checkbox"/>	0x00400054	0x0271602a	slt \$12,\$19,\$17	29: slt \$t4, \$s3,\$s1
<input type="checkbox"/>	0x00400058	0x1180ffe5	beq \$12,\$0,-17	30: beq \$t4, \$0, loop

Sau khi chạy chương trình thì ta nhận được giá trị \$s5 = 198 là giá trị max nhất và \$t7 chính là phần tử thứ A[i] chứa giá trị MAX đó

Phần giải thích em đã viết ở phần comment trong bài

Kết quả của lệnh chạy trên

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	268501052
\$t2	10	0
\$t3	11	0
\$t4	12	1
\$t5	13	0
\$t6	14	0
\$t7	15	12
\$s0	16	0
\$s1	17	15
\$s2	18	268500992
\$s3	19	14
\$s4	20	1
\$s5	21	198
\$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		4194400
hi		0
lo		0

Ati Max

Max