

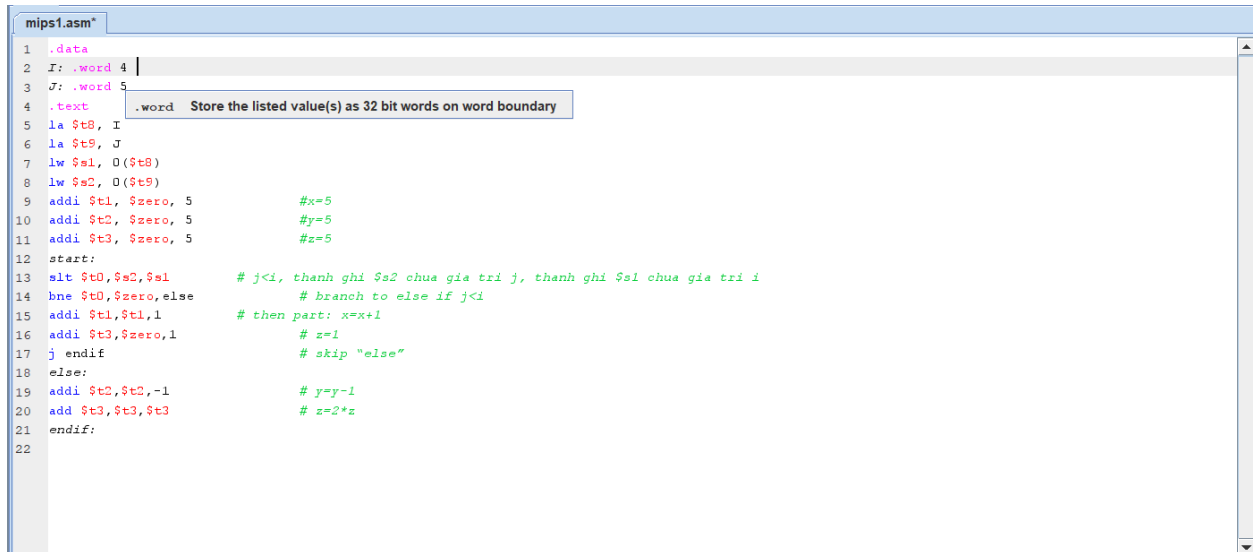
Họ và Tên: Hà Trung Chiến

MSSV: 20225794

- **Assignment 1:**

Đặt $x = y = z = 5$

Trường hợp 1: $I < J$ (theo lý thuyết chương trình sẽ trả ra kết quả $x = 6, y = 5, z = 1$)



```
1 .data
2 I: .word 4
3 J: .word 5
4 .text
5 la $t8, I
6 la $t9, J
7 lw $s1, 0($t8)
8 lw $s2, 0($t9)
9 addi $t1, $zero, 5      #x=5
10 addi $t2, $zero, 5     #y=5
11 addi $t3, $zero, 5     #z=5
12 start:
13 slt $t0, $s2, $s1      # j<i, thanh ghi $s2 chua gia tri j, thanh ghi $s1 chua gia tri i
14 bne $t0, $zero, else   # branch to else if j<i
15 addi $t1, $t1, 1       # then part: x=x+1
16 addi $t3, $zero, 1     # z=1
17 j endif                # skip "else"
18 else:
19 addi $t2, $t2, -1      # y=y-1
20 add $t3, $t3, $t3      # z=2*z
21 endif:
22
```

sau khi slt so sánh 2 giá trị $I = 4 < J = 5$ thanh ghi \$t0 có giá trị bằng 0, sau đó tới câu lệnh bne so sánh \$t0 với \$zero sau đó thực thi câu lệnh ở phía dưới rồi nhảy đến kết thúc mà không thực hiện lệnh else, thực hiện các câu lệnh $x = x + 1, z = 1$.

Kết thúc chương trình cho ra kết quả x, y, z như lý thuyết.

Registers	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	
\$at	1	26850099
\$v0	2	
\$v1	3	
\$a0	4	
\$a1	5	
\$a2	6	
\$a3	7	
\$t0	8	
\$t1	9	
\$t2	10	
\$t3	11	
\$t4	12	
\$t5	13	
\$t6	14	
\$t7	15	
\$s0	16	
\$s1	17	
\$s2	18	
\$s3	19	
\$s4	20	
\$s5	21	
\$s6	22	
\$s7	23	
\$t8	24	26850099
\$t9	25	26850099
\$k0	26	
\$k1	27	
\$gp	28	26846822
\$sp	29	214747954
\$fp	30	
\$ra	31	
pc		419436
hi		
lo		

Trường hợp 2: $I > J$ (theo lý thuyết chương trình sẽ trả ra kết quả $x = 5, y = 4, z = 10$)

```

Edit Execute
mips1.asm*
1 .data
2 I: .word 5
3 J: .word 4
4 .text
5 la $t8, I      .word Store the listed value(s) as 32 bit words on word boundary
6 la $t9, J
7 lw $s1, 0($t8)
8 lw $s2, 0($t9)
9 addi $t1, $zero, 5      #x=5
10 addi $t2, $zero, 5     #y=5
11 addi $t3, $zero, 5     #z=5
12 start:
13 slt $t0, $s2, $s1      # j<i, thanh ghi $s2 chua gia tri j, thanh ghi $s1 chua gia tri i
14 bne $t0, $zero, else   # branch to else if j<i
15 addi $t1, $t1, 1       # then part: x=x+1
16 addi $t3, $zero, 1     # z=1
17 j endif               # skip "else"
18 else:
19 addi $t2, $t2, -1      # y=y-1
20 add $t3, $t3, $t3      # z=2*z
21 endif:
22

```

sau khi slt so sánh 2 giá trị $I = 5 < J = 4$ thanh ghi \$t0 có giá trị bằng 1, sau đó tới câu lệnh bne so sánh \$t0 với \$zero sau đó thực thi câu lệnh ở phía dưới rồi nhảy đến kết thúc mà không thực hiện lệnh else, thực hiện các câu lệnh $y = y - 1, z = 2 * z$.

Kết thúc chương trình cho ra kết quả x, y, z như lý thuyết.

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	1
\$t1	9	5
\$t2	10	4
\$t3	11	10
\$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	268500992
\$t9	25	268500996
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194368
hi		0
lo		0

Trường hợp 3: $I = J$ (hay $I \leq J$) (theo lý thuyết chương trình sẽ trả ra kết quả $x = 6, y = 5, z = 1$)

```

Edit Execute
mips1.asm
1 .data
2 I: .word 5
3 J: .word 5
4 .text
5 la $t8, I      Store the listed value(s) as 32 bit words on word boundary
6 la $t9, J
7 lw $s1, 0($t8)
8 lw $s2, 0($t9)
9 addi $t1, $zero, 5      #x=5
10 addi $t2, $zero, 5      #y=5
11 addi $t3, $zero, 5      #z=5
12 start:
13 slt $t0, $s2, $s1      # j<i, thanh ghi $s2 chua gia tri j, thanh ghi $s1 chua gia tri i
14 bne $t0, $zero, else   # branch to else if j<i
15 addi $t1, $t1, 1      # then part: x=x+1
16 addi $t3, $zero, 1      # z=1
17 j endif               # skip "else"
18 else:
19 addi $t2, $t2, -1      # y=y-1
20 add $t3, $t3, $t3      # z=2*z
21 endif:
22

```

sau khi slt so sánh 2 giá trị $I = 4 < J = 5$ thanh ghi \$t0 có giá trị bằng 0, sau đó tới câu lệnh bne so sánh \$t0 với \$zero sau đó thực thi câu lệnh ở phía dưới rồi nhảy đến kết thúc mà không thực hiện lệnh else, thực hiện các câu lệnh $x = x + 1, z = 1$.

Kết thúc chương trình cho ra kết quả x, y, z như lý thuyết.

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			6
\$t2	10			5
\$t3	11			1
\$t4	12			0
\$t5	13			0
\$t6	14			0
\$t7	15			0
\$s0	16			0
\$s1	17			3
\$s2	18			3
\$s3	19			0
\$s4	20			0
\$s5	21			0
\$s6	22			0
\$s7	23			0
\$t8	24			268500992
\$t9	25			268500996
\$k0	26			0
\$k1	27			0
\$gp	28			268468224
\$sp	29			2147479548
\$fp	30			0
\$ra	31			0
pc				4194368
hi				0
lo				0

- Assignment 2:

```

mips1.asm*
1  .data
2  A: .word 1,2,3,4,5,6,7,8,9,10
3  .text
4  addi $s3, $zero, 10      #tao so phan tu cua mang
5  addi $s4, $zero, 1       #tao buoc nhay
6  la $s2, A                #Truy nhap thanh ghi $s2 vao dia chi A
7  addi $s6, $zero, 0       #sum = 0
8  addi $s1, $zero, 0       #i = 0
9  loop: slt $t2, $s1, $s3   # $t2 = i < n ? 1 : 0
10 beq $t2, $zero, endloop
11 add $t1, $s1, $s1        # $t1 = 2 * $s1
12 add $t1, $t1, $t1        # $t1 = 4 * $s1
13 add $t1, $t1, $s2        # $t1 store the address of A[i]
14 lw $t0, 0($t1)           # load value of A[i] in $t0
15 add $s6, $s6, $t0        # sum = sum + A[i]
16 add $s1, $s1, $s4        # i = i + step
17 j loop                  # goto loop
18 endloop:
19

```

Ban đầu khởi tạo mảng A có 10 phần tử, \$s6 là thanh ghi lưu tổng các phần tử của mảng. Sau đó thực hiện chạy vòng lặp, ban đầu kiểm tra điều kiện so sánh \$s1 (thứ tự hiện tại của phần tử) và \$s3 (số phần tử lớn nhất trong mảng). Nếu \$s1 < \$s3 tức giá trị \$t2 = 1 lần lượt thực hiện

các câu lệnh ở bên dưới tính tổng lưu vào thanh ghi \$6 và tăng thứ tự của phần tử hiện tại lên 1 đơn vị “\$s1 = \$s1 + \$s4 (bước nhảy của mảng(1))” rồi nhảy đến “loop” và thực hiện tương tự như trên cho đến khi duyệt hết các phần tử của mảng A.

Khi thực hiện xong thu được kết quả lưu ở thanh ghi \$s6 là 55 đúng với lý thuyết.

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	10
\$t1	9	268501028
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	10
\$s2	18	268500992
\$s3	19	10
\$s4	20	1
\$s5	21	0
\$s6	22	55
\$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

- **Assignment 3:**

Đặt a = b = 6 lưu vào thanh ghi \$s2 và \$s3.

Theo lý thuyết:

Nếu test = 0, a = 7

Nếu test = 1, a = 5

Nếu test = 2, b = 12

```

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

```

Sau khi chạy chương trình, tùy thuộc vào giá trị test được lưu trong \$s1 lệnh beq lần lượt so sánh với các giá trị \$t0, \$t1, \$t2 nếu đúng sẽ thực hiện case tương ứng.

Ở đây chọn test = 1 => thực hiện case 1: thực hiện thay đổi giá trị của a = a – 1 = 5 (đúng với lý thuyết) sau đó nhảy tới continue. Vì không có lệnh nào nên giá trị của các thanh ghi không thay đổi nữa.

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	5
\$s3	19	6
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$s8	24	0
\$s9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	4194376
\$0		0
\$1		0
\$2		0

Thực hiện tương tự với các trường hợp còn lại đều thu được kết quả đúng.

• Assignment 4:

a. $i < j$

```

Edit Execute
mips1.asm
1 .data
2 I: .word 3
3 J: .word 5
4 .text
5 la $t8, I      .word Store the listed value(s) as 32 bit words on word boundary
6 la $t9, J
7 lw $s1, 0($t8)
8 lw $s2, 0($t9)
9 addi $t1, $zero, 5      #x=5
10 addi $t2, $zero, 5      #y=5
11 addi $t3, $zero, 5      #z=5
12 start:
13 slt $t0, $s2, $s1      # j<i, thanh ghi $s2 chua gia tri j, thanh ghi $s1 chua gia tri i
14 bne $t0, $zero, else   # branch to else if j<i
15 addi $t1, $t1, 1      # then part: x=x+1
16 addi $t3, $zero, 1      # z=1
17 j endif               # skip "else"
18 else:
19 addi $t2, $t2, -1      # y=y-1
20 add $t3, $t3, $t3      # z=2*z
21 endif:
22

```

Kết quả:

Khi thay đổi điều kiện thành $i < j$ thì giá trị lưu ở thanh ghi \$t0 = 0 không thỏa mãn điều kiện nên không thực hiện “else”

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	6
\$t2	10	5
\$t3	11	1
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	3
\$s2	18	5
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	268500992
\$t9	25	268500996
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194368
hi		0
lo		0

b. $i \geq j$

```

mips1.asm
1 data
2 I: .word 5
3 J: .word 3
4 .text
5 la $t8, I
6 la $t9, J
7 lw $s1, 0($t8)
8 lw $s2, 0($t9)
9 addi $t1, $zero, 5      #x=5
10 addi $t2, $zero, 5     #y=5
11 addi $t3, $zero, 5     #z=5
12 start:
13 slt $t0,$s2,$s1        # j<i, thanh ghi $s2 chua gia tri j, thanh ghi $s1 chua gia tri i
14 bne $t0,$zero,else     # branch to else if j<i
15 addi $t1,$t1,1         # then part: x=x+1
16 addi $t3,$zero,1       # z=1
17 j endif               # skip "else"
18 else:
19 addi $t2,$t2,-1        # y=y-1
20 add $t3,$t3,$t3        # z=2*z
21 endif:
22

```

Kết quả:

Khi thay đổi điều kiện thành $i \geq j$ thì giá trị lưu ở thanh ghi \$t0 = 1 thỏa mãn điều kiện nên thực hiện “else”.

Registers	Coproc 1	Coproc 0	
CPU registers	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		1
\$t1	9		5
\$t2	10		4
\$t3	11		1
\$t4	12		0
\$t5	13		0
\$t6	14		0
\$t7	15		0
\$s0	16		0
\$s1	17		5
\$s2	18		3
\$s3	19		0
\$s4	20		0
\$s5	21		0
\$s6	22		0
\$s7	23		0
\$t8	24		268500992
\$t9	25		268500996
\$k0	26		0
\$k1	27		0
\$gp	28		268468224
\$sp	29		2147479548
\$fp	30		0
\$ra	31		0
\$lo			4194368
\$hi			0
\$lo			0

c. $i + j \leq 0$


```

mips1.asm
1  .data
2  J: .word -10
3  J: .word 5
4  .text
5  la $t8, J
6  la $t9, J
7  lw $s1, 0($t8)
8  lw $s2, 0($t9)
9  addi $t1, $zero, 5      #x=5
10 addi $t2, $zero, 5      #y=5
11 addi $t3, $zero, 5      #z=5
12 add $s7, $s1, $s2      # tính tổng I + J
13 start:
14 slt $t0, $s7, $zero     # j<i, thanh ghi $s2 chua gia tri j, thanh ghi $s1 chua gia tri i
15 bne $t0, $zero, else    # branch to else if j<i
16 addi $t1, $t1, 1        # then part: x=x+1
17 addi $t3, $zero, 1      # z=1
18 j endif                # skip "else"
19 else:
20 addi $t2, $t2, -1        # y=y-1
21 add $t3, $t3, $t3        # z=2*z
22 endif:
23

```

Kết quả thu được đúng với lý thuyết

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		1
\$t1	9		5
\$t2	10		4
\$t3	11		10
\$t4	12		0
\$t5	13		0
\$t6	14		0
\$t7	15		0
\$s0	16		0
\$s1	17		-10
\$s2	18		5
\$s3	19		0
\$s4	20		0
\$s5	21		0
\$s6	22		0
\$s7	23		-5
\$t8	24		268500992
\$t9	25		268500996
\$k0	26		0
\$k1	27		0
\$gp	28		268468224
\$sp	29		2147479548
\$fp	30		0
\$ra	31		0
pc			4194372
hi			0
lo			0

d. $i + j > m + n$

```

mips1.asm
1  .data
2  I: .word 6
3  J: .word 5
4  .text
5  la $t8, I
6  la $t9, J
7  lw $s1, 0($t8)
8  lw $s2, 0($t9)
9  addi $t1, $zero, 5      #x=5
10 addi $t2, $zero, 5      #y=5
11 addi $t3, $zero, 5      #z=5
12 addi $t4, $zero, 5      #n=5
13 addi $t5, $zero, 5      #n=5
14 add $s6, $t4, $t5        # tính tổng m + n
15 add $s7, $s1, $s2        # tính tổng I + J
16 start:
17 slt $t0, $s6, $s7        # j+i>m+n
18 bne $t0, $zero, else
19 addi $t1, $t1, 1          # then part: x=x+1
20 addi $t3, $zero, 1        # z=1
21 j endif                  # skip "else"
22 else:
23 addi $t2, $t2, -1         # y=y-1
24 add $t3, $t3, $t3        # z=2*z
25 endif:
26

```

Kết quả đúng với thực nghiệm

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	1	
\$t1	9	5	
\$t2	10	4	
\$t3	11	10	
\$t4	12	5	
\$t5	13	5	
\$t6	14	0	
\$t7	15	0	
\$s0	16	0	
\$s1	17	6	
\$s2	18	5	
\$s3	19	0	
\$s4	20	0	
\$s5	21	0	
\$s6	22	10	
\$s7	23	11	
\$t8	24	268500992	
\$t9	25	268500996	
\$k0	26	0	
\$k1	27	0	
\$gp	28	268468224	
\$sp	29	2147479548	
\$fp	30	0	
\$ra	31	0	
pc		4194384	
hi		0	
lo		0	

- Assignment 5:

a. $i < n$

The screenshot shows the MIPS simulator with the following assembly code:

```

1 .data
2 A: .word 1,2,3,4,5,6,7,8,9,10
3 .text
4 addi $s3, $zero, 10 #tao so phan tu cua mang
5 addi $s4, $zero, 1 #tao huoc nhay
6 la $a2, A #Truy nhap thanh ghi $a2 vao dia chi A
7 addi $s5, $zero, 0 #sum = 0
8 addi $t1, $zero, 0 #i = 0
9 loop:
10 slt $t2, $t1, $s3 # $t2 = i < n? 1 : 0
11 beq $t2, $zero, endloop
12 add $t1, $t1, $s4 # $t1 = $t1 + $s4
13 add $t1, $t1, $t1 # $t1 = 4 * $s4
14 add $t1, $t1, $a2 # $t1 store the address of A[i]
15 lw $t0, 0($t1) #load value of A[i] in $t0
16 add $s5, $s5, $t0 #sum = sum + A[i]
17 add $t1, $t1, $s4 #i = i + step
18 j loop #goto loop
19 endloop:

```

The register window shows the following values:

Register	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	268501020
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	268500992
\$s1	17	0
\$s2	18	268500992
\$s3	19	10
\$s4	20	1
\$s5	21	55
\$s6	22	0
\$s7	23	0
\$s8	24	0
\$s9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$ep	29	2147479546
\$fp	30	0
\$ra	31	4194364
\$pc		
\$hi		
\$lo		

b. $i \leq n$

The screenshot shows the MIPS simulator with the following assembly code:

```

1 .data
2 A: .word 1,2,3,4,5,6,7,8,9,10
3 .text
4 addi $s3, $zero, 10 #tao so phan tu cua mang
5 addi $s4, $zero, 1 #tao huoc nhay
6 la $a2, A #Truy nhap thanh ghi $a2 vao dia chi A
7 addi $s5, $zero, 0 #sum = 0
8 addi $t1, $zero, 0 #i = 0
9 loop:
10 slt $t2, $t1, $s3 # $t2 = i < n? 1 : 0
11 beq $t2, $zero, endloop
12 add $t1, $t1, $s4 # $t1 = $t1 + $s4
13 add $t1, $t1, $t1 # $t1 = 4 * $s4
14 add $t1, $t1, $a2 # $t1 store the address of A[i]
15 lw $t0, 0($t1) #load value of A[i] in $t0
16 add $s5, $s5, $t0 #sum = sum + A[i]
17 add $t1, $t1, $s4 #i = i + step
18 j loop #goto loop
19 endloop:

```

The register window shows the following values:

Register	Number	Value
\$zero	0	0
\$at	1	1
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	268501032
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	11
\$s2	18	268500992
\$s3	19	10
\$s4	20	1
\$s5	21	55
\$s6	22	0
\$s7	23	0
\$s8	24	0
\$s9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$ep	29	2147479548
\$fp	30	0
\$ra	31	0
\$pc		4194372
\$hi		0
\$lo		0

c. $\text{sum} > 0$

The screenshot shows the MIPS simulator with the following assembly code:

```

1 .data
2 A: .word 1,2,3,-7,5,6,7,8,9,10
3 .text
4 addi $s3, $zero, 10 #tao so phan tu cua mang
5 addi $s4, $zero, 1 #tao huoc nhay
6 la $a2, A #Truy nhap thanh ghi $a2 vao dia chi A
7 addi $s5, $zero, 0 #sum = 0
8 addi $t1, $zero, 0 #i = 0
9 loop:
10 add $t1, $t1, $s4 # $t1 = $t1 + $s4
11 add $t1, $t1, $t1 # $t1 = 4 * $s4
12 add $t1, $t1, $a2 # $t1 store the address of A[i]
13 lw $t0, 0($t1) #load value of A[i] in $t0
14 add $s5, $s5, $t0 #sum = sum + A[i]
15 add $t1, $t1, $s4 #i = i + step
16 slt $t2, $zero, $s5
17 bne $t2, $zero, loop
18 endloop:

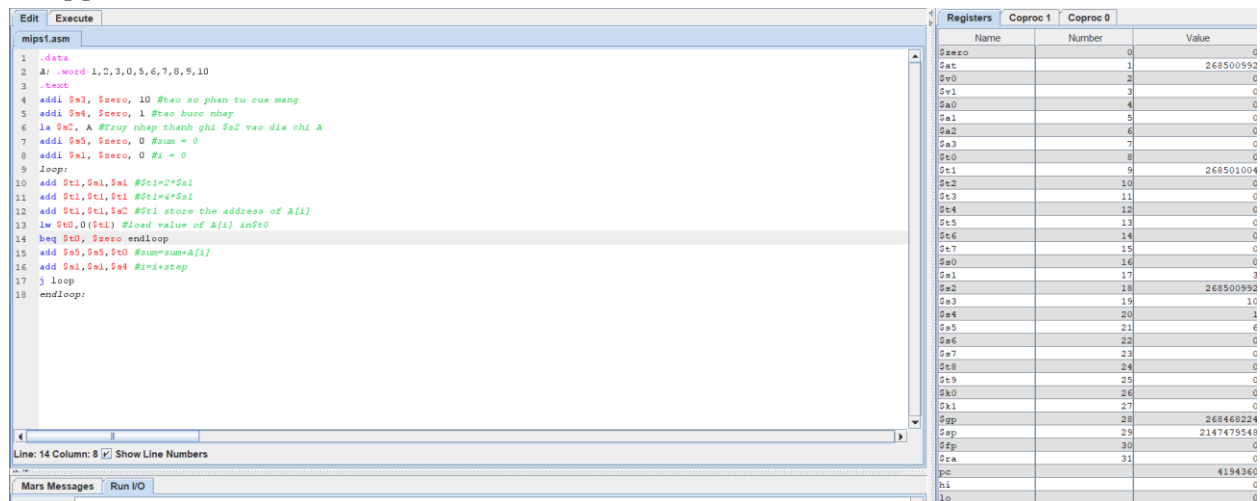
```

The register window shows the following values:

Register	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	-7
\$t1	9	268501004
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	4
\$s1	17	0
\$s2	18	268500992
\$s3	19	10
\$s4	20	1
\$s5	21	-1
\$s6	22	0
\$s7	23	0
\$s8	24	0
\$s9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$ep	29	2147479548
\$fp	30	0
\$ra	31	0
\$pc		4194360
\$hi		0
\$lo		0

Sau khi chương trình chạy đến phần tử thứ 5 là 5 lúc này giá trị lưu ở thanh ghi là \$s5 = -1 < 0 không thỏa mãn điều kiện nên không đi vào loop và kết thúc vòng lặp.

d. $A[i] \neq 0$



The screenshot shows the MIPS simulator interface. The assembly code in the main window is as follows:

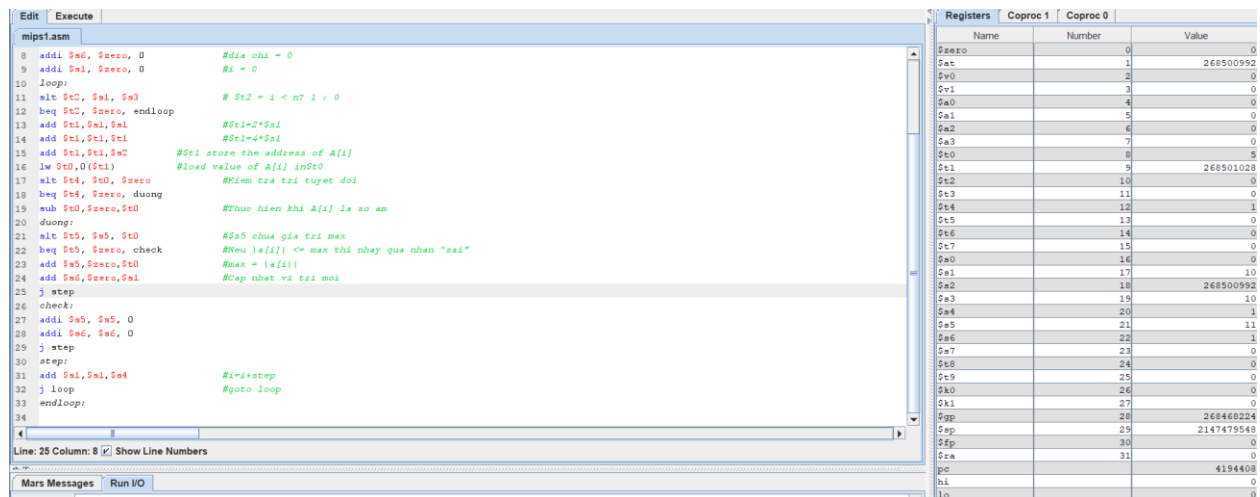
```
1 .data
2 A: .word 1,2,3,0,5,6,7,8,9,10
3 .text
4 addi $s3, $zero, 10 #tạo số phần tử của mảng
5 addi $s4, $zero, 1 #tạo bước nhảy
6 la $s2, A #tạo thanh ghi $s2 vào địa chỉ A
7 addi $s5, $zero, 0 #sum = 0
8 addi $s1, $zero, 0 #i = 0
9 loop:
10 add $t1, $s1, $s4 #t1 = i + s4
11 add $t1, $t1, $t1 #t1 = 2*i
12 add $t1, $t1, $s2 #t1 store the address of A[i]
13 lw $t0, 0($t1) #load value of A[i] into $t0
14 beq $t0, $zero, endloop
15 add $s5, $s5, $t0 #sum = sum + A[i]
16 add $s1, $s1, $s4 #i = i + step
17 j loop
18 endloop:
```

The register window on the right shows the following values:

Register	Value
\$zero	0
\$at	268500992
\$v0	0
\$v1	0
\$a0	0
\$a1	0
\$a2	0
\$a3	0
\$t0	0
\$t1	268501004
\$t2	0
\$t3	0
\$t4	0
\$t5	0
\$t6	0
\$t7	0
\$s0	0
\$s1	17
\$s2	268500992
\$s3	19
\$s4	1
\$s5	6
\$s6	0
\$s7	0
\$s8	0
\$s9	0
\$k0	0
\$k1	0
\$gp	268468224
\$fp	2147479548
\$ra	0
\$pc	4194360
\$lo	0

sau khi chương trình chạy tới phần tử thứ 4 là 0 thỏa mãn điều kiện so sánh nên chương trình nhảy tới endloop và kết thúc.

• Assignment 6



The screenshot shows the MIPS simulator interface for Assignment 6. The assembly code in the main window is as follows:

```
8 addi $s6, $zero, 0 #địa chỉ = 0
9 addi $s1, $zero, 0 #i = 0
10 loop:
11 slt $t2, $s1, $s3 # $t2 = 1 < n? 1 : 0
12 beq $t2, $zero, endloop
13 add $t1, $s1, $s1 #t1 = 2*i
14 add $t1, $t1, $s1 #t1 = 4*i
15 add $t1, $t1, $s2 #t1 store the address of A[i]
16 lw $t0, 0($t1) #load value of A[i] into $t0
17 slt $t4, $t0, $s5 #Kiểm tra trị tuyệt đối
18 beq $t4, $zero, duong #Thực hiện khi A[i] là số âm
19 mul $t0, $zero, $t0
20 duong:
21 slt $t5, $s5, $t0 #s5 chứa giá trị max
22 beq $t5, $zero, check #Nếu |a[i]| <= max thì nhảy qua nhãn "sai"
23 add $s5, $zero, $t0 #max = |a[i]|
24 add $s6, $zero, $s1 #Cập nhật vị trí mới
25 j step
26 check:
27 addi $s5, $s5, 0
28 addi $s6, $s6, 0
29 j step
30 step:
31 add $s1, $s1, $s4 #i = i + step
32 j loop
33 endloop:
```

The register window on the right shows the following values:

Register	Value
\$zero	0
\$at	268500992
\$v0	0
\$v1	0
\$a0	0
\$a1	0
\$a2	0
\$a3	0
\$t0	5
\$t1	268501028
\$t2	0
\$t3	0
\$t4	1
\$t5	0
\$t6	0
\$t7	0
\$s0	0
\$s1	17
\$s2	268500992
\$s3	19
\$s4	1
\$s5	11
\$s6	1
\$s7	0
\$s8	0
\$s9	0
\$k0	0
\$k1	0
\$gp	268468224
\$fp	2147479548
\$ra	0
\$pc	4194408
\$lo	0

Thanh ghi \$s5 lưu giá trị max