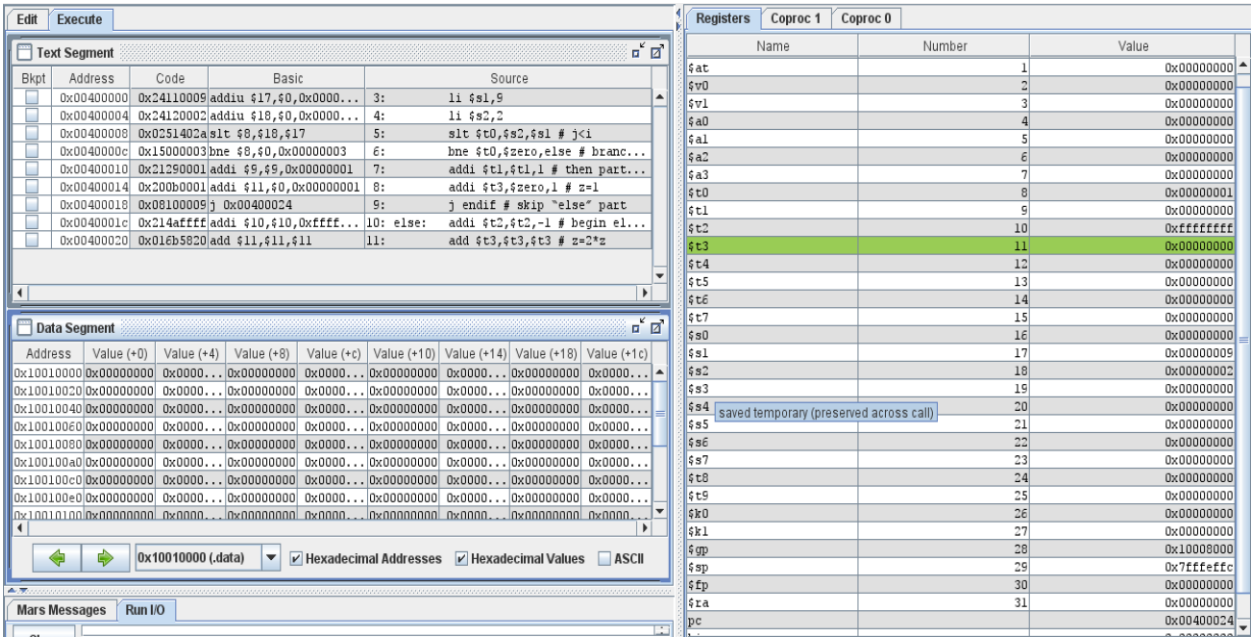
Phạm Duy Hưng – 20225850

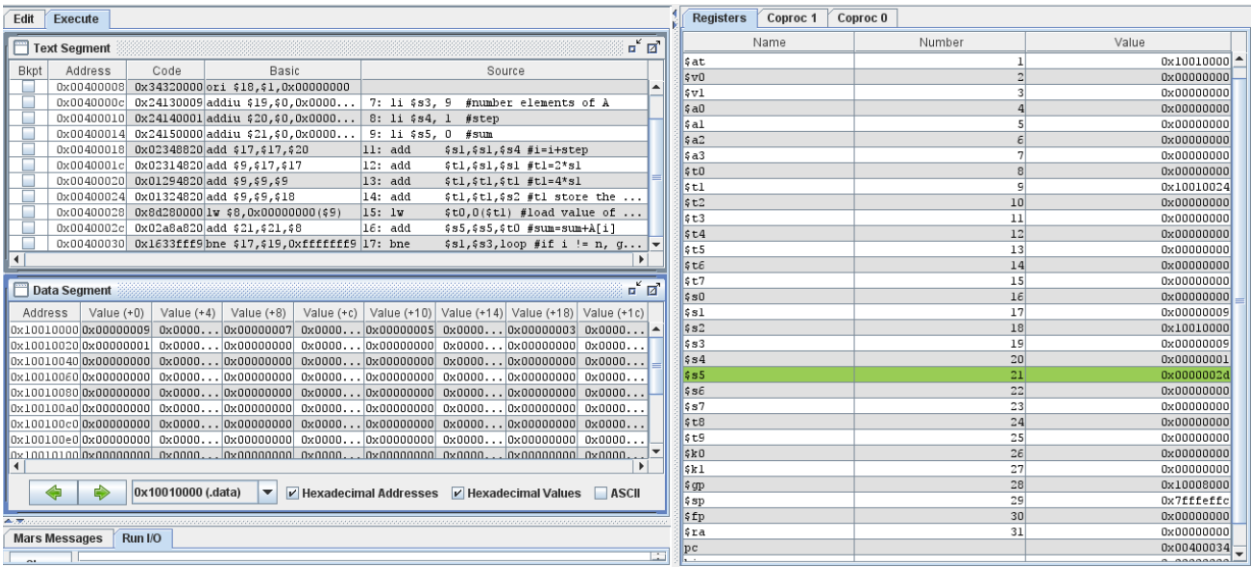
Assignment 1:

```
start:
        li $s1,9 #i=9
        li $s2,2 #j=2
        slt $t0,$s2,$s1 #j<i
        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:
```



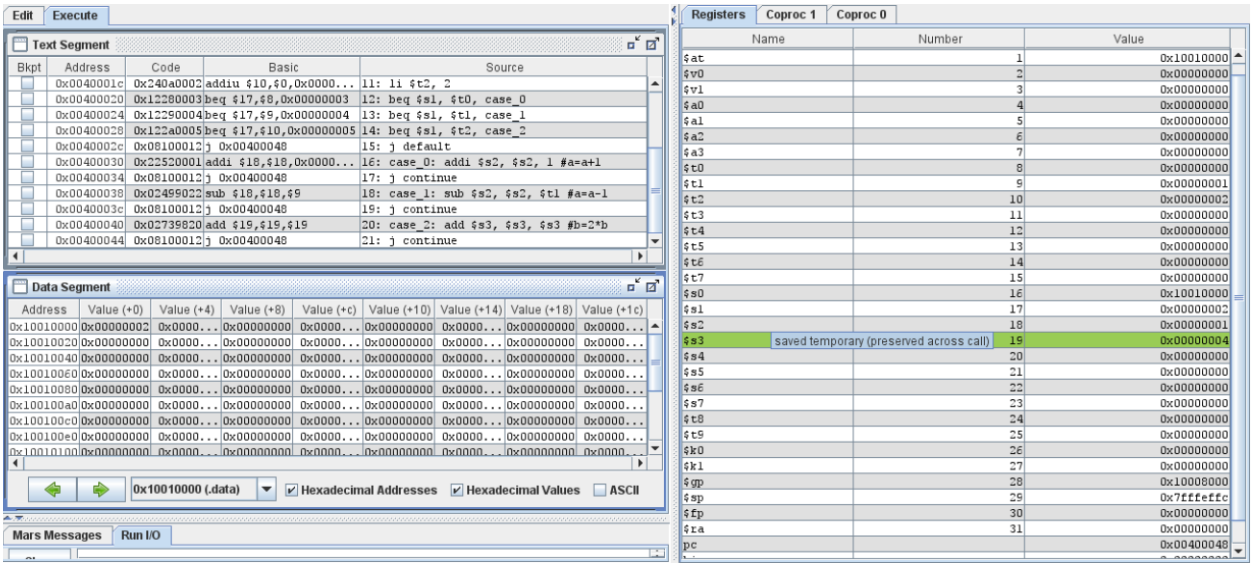
Assignment 2:

```
.data

A: .word 9,8,7,6,5,4,3,2,1

.text

li $s1, -1 #i=-1

la $s2, A  #s2 stores the address of array

li $s3, 9  #number elements of A

li $s4, 1  #step

li $s5, 0  #sum

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 the address of A[i]

lw      $t0,0($t1) #load value of A[i] in $t0

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

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



Assignment 3:

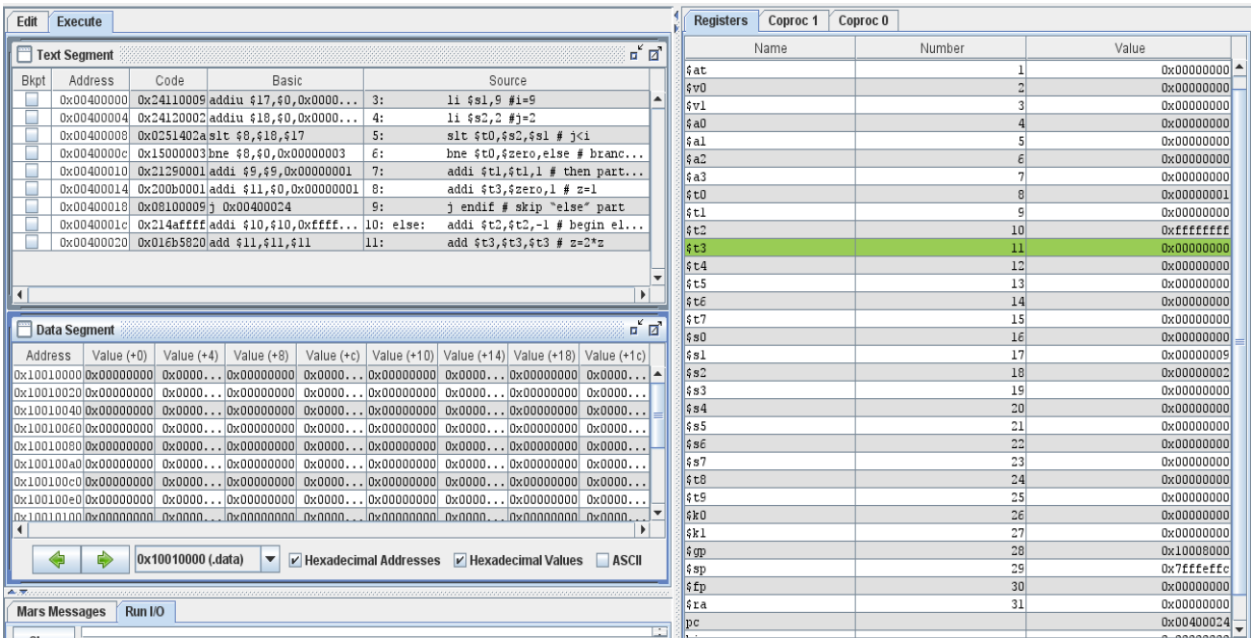
```
.data
test: .word 2
.text
li $s2,1 #a=1
li $s3,2 #b=2
la $s0, test #load the address of test variable
lw $s1, 0($s0) #load the value of test to register $t1
li $t0, 0 #load value for test case
li $t1, 1
li $t2, 2
beq $s1, $t0, case_0
beq $s1, $t1, case_1
beq $s1, $t2, case_2
j default
case_0: addi $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:
```


Edit
Execute
Text Segment
Bkpt Address Code Basic Source
0x0040001c 0x240a0002 addiu $10,$0,0x0000... 11: li $t2, 2
0x00400020 0x12280003 beq $17,$8,0x00000003 12: beq $s1, $t0, case_0
0x00400024 0x12290004 beq $17,$9,0x00000004 13: beq $s1, $t1, case_1
0x00400028 0x122a0005 beq $17,$10,0x00000005 14: beq $s1, $t2, case_2
0x0040002c 0x08100012 j 0x00400048 15: j default
0x00400030 0x22520001 addi $18,$18,0x0000... 16: case_0: addi $s2, $s2, 1 #a=a+1
0x00400034 0x08100012 j 0x00400048 17: j continue
0x00400038 0x02499022 sub $18,$18,$9 18: case_1: sub $s2, $s2, $t1 #a=a-1
0x0040003c 0x08100012 j 0x00400048 19: j continue
0x00400040 0x02739820 add $19,$19,$19 20: case_2: add $s3, $s3, $s3 #b=2*b
0x00400044 0x08100012 j 0x00400048 21: j continue
Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+c) Value (+10) Value (+14) Value (+18) Value (+1c)
0x10010000 0x00000002 0x0000... 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000...
0x10010020 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000...
0x10010040 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000...
0x10010060 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000...
0x10010080 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000...
0x100100a0 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000...
0x100100c0 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000...
0x100100e0 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000...
0x10010100 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000... 0x00000000 0x0000...
0x10010000 (.data)
Hexadecimal Addresses
Hexadecimal Values
ASCII
Mars Messages
Run I/O
Registers
Coproc 1
Coproc 0
Name Number Value
$at 1 0x10010000
$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 0x00000001
$s3 saved temporary (preserved across call) 19 0x00000004
$s4 20 0x00000000
$s5 21 0x00000000
$s6 22 0x00000000
$s7 23 0x00000000
$t8 24 0x00000000
$t9 25 0x00000000
$k0 26 0x00000000
$k1 27 0x00000000
$gp 28 0x10008000
$sp 29 0x7fffeffc
$fp 30 0x00000000
$ra 31 0x00000000
pc 0x00400048

Assignment 4:

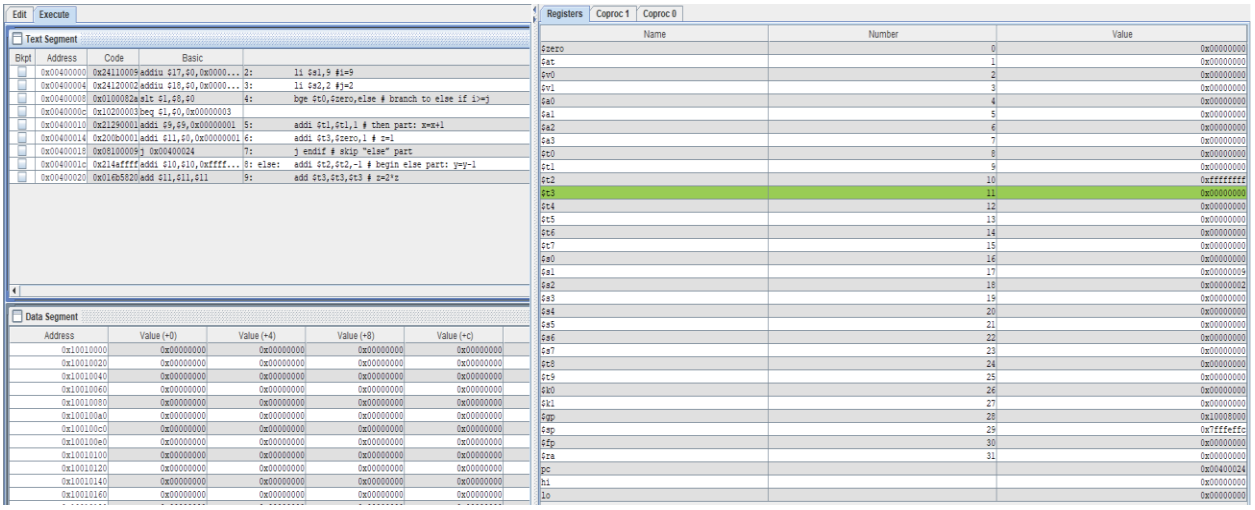
a.

```
start:
        li $s1,9 #i=9
        li $s2,2 #j=2
        slt $t0,$s2,$s1 # j<i
        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:
```



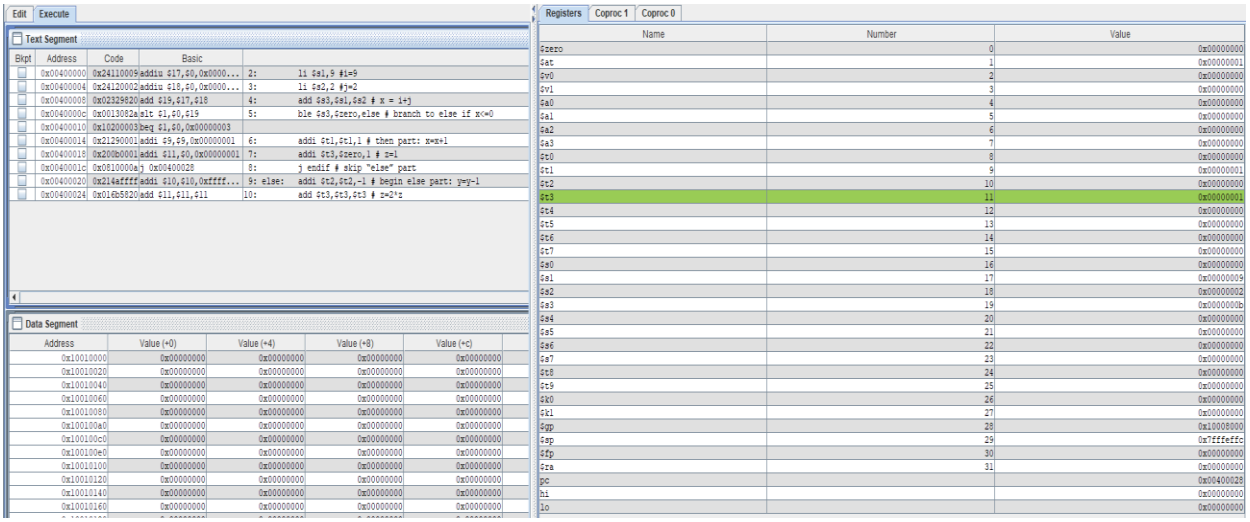
b.

```
start:
        li $s1,9 #i=9
        li $s2,2 #j=2
        bge $t0,$zero,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:   addi $t2,$t2,-1 # begin else part: y=y-1
        add $t3,$t3,$t3 # z=2*z
endif:
```



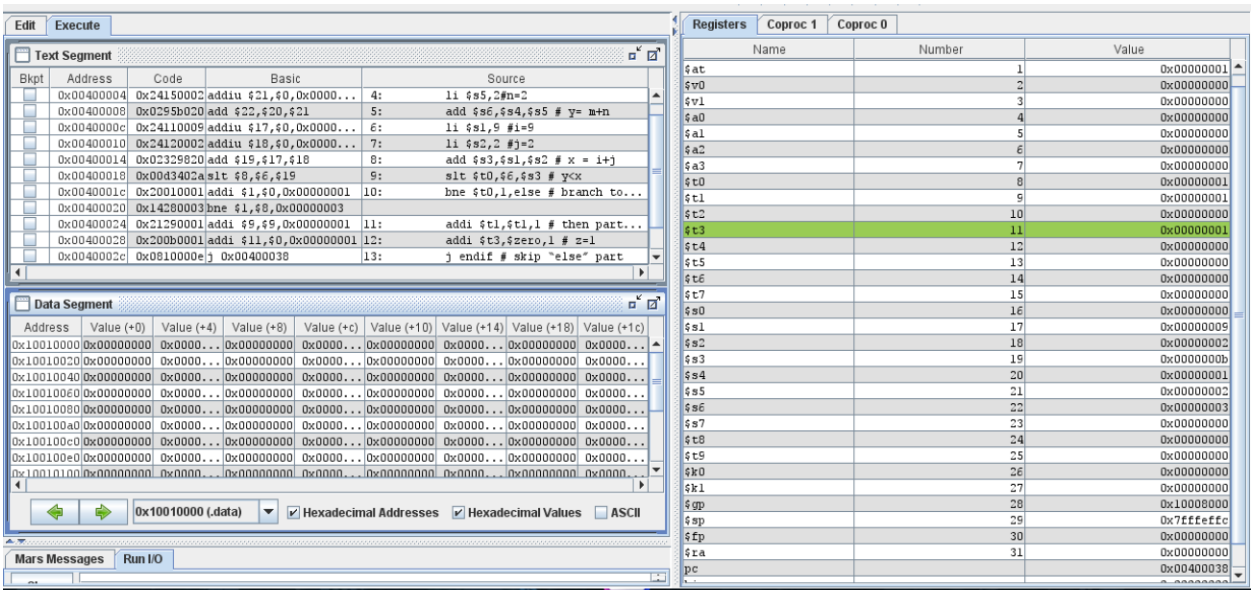
c.

```
start:
        li $s1,9 #i=9
        li $s2,2 #j=2
        add $s3,$s1,$s2 # x = i+j
        ble $s3,$zero,else # branch to else if x<=0
        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:
```



d.

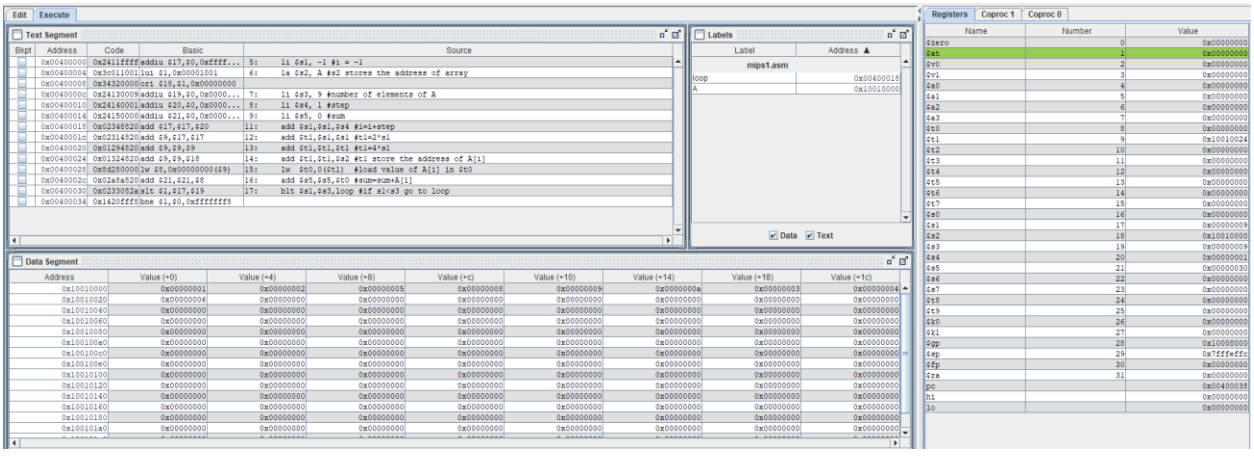
```
start:
        li $s4,1#m=1
        li $s5,2#n=2
        add $s6,$s4,$s5 # y= m+n
        li $s1,9 #i=9
        li $s2,2 #j=2
        add $s3,$s1,$s2 # x = i+j
        slt $t0,$6,$s3 # y<x
        bne $t0,1,else # branch to else if t0 not equal to 1
        addi $t1,$t1,1 # then part: x=x+1
        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:
```



Assignment 5:

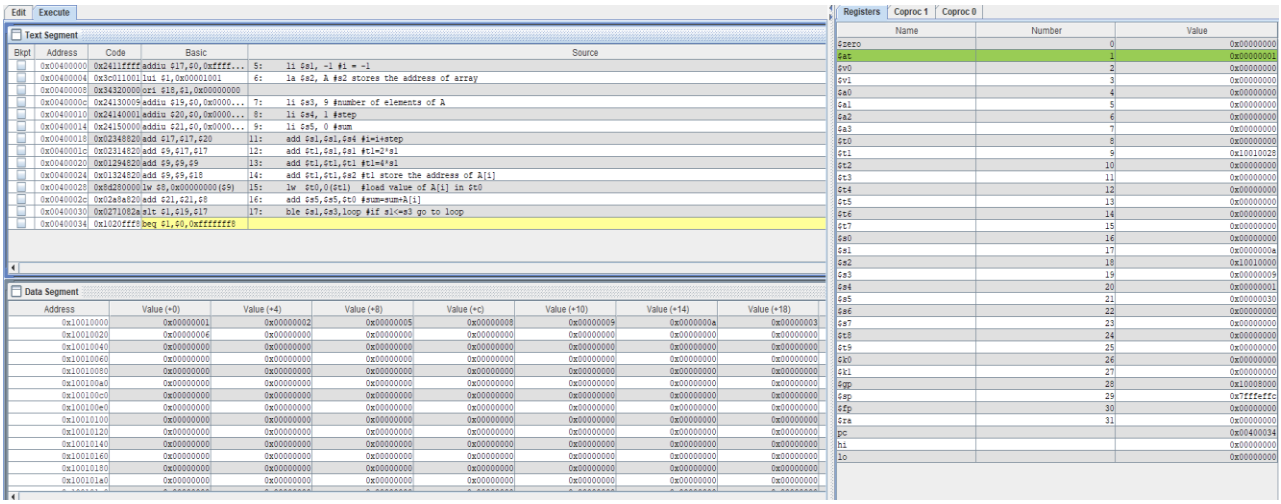
a.

```
.data
A: .word 1, 2, 5, 8, 9, 10, 3, 4, 6
.text
   li $s1, -1 #i = -1
   la $s2, A #s2 stores the address of array
   li $s3, 9 #number of elements of A
   li $s4, 1 #step
   li $s5, 0 #sum
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 the address of A[i]
   lw  $t0,0($t1)  #load value of A[i] in $t0
   add $s5,$s5,$t0 #sum=sum+A[i]
   blt $s1,$s3,loop #if s1<s3 go to loop
```



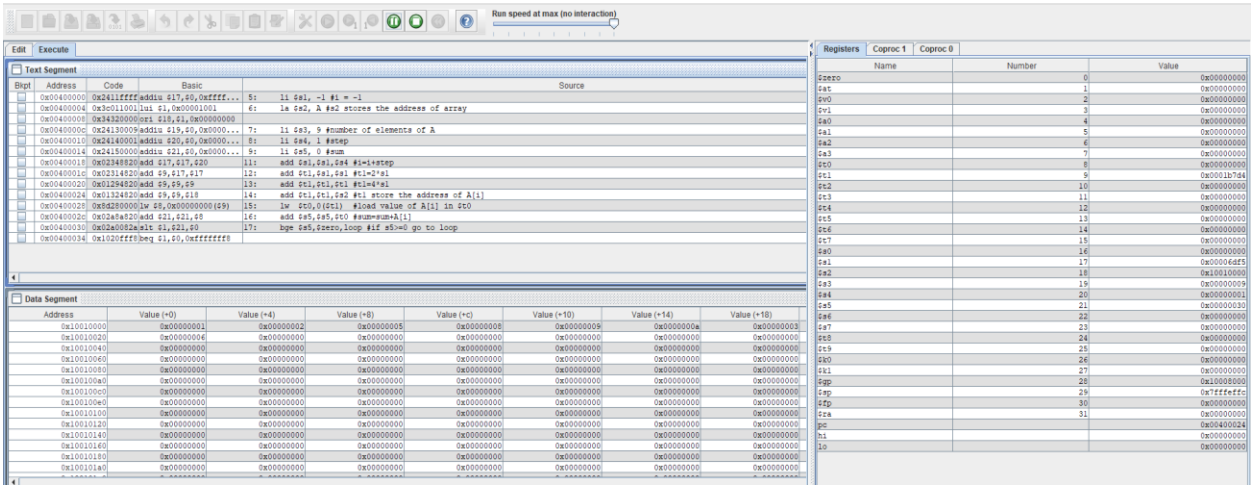
b.

```
.data
A: .word 1, 2, 5, 8, 9, 10, 3, 4, 6
.text
   li $s1, -1 #i = -1
   la $s2, A #s2 stores the address of array
   li $s3, 9 #number of elements of A
   li $s4, 1 #step
   li $s5, 0 #sum
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 the address of A[i]
   lw  $t0,0($t1)  #load value of A[i] in $t0
   add $s5,$s5,$t0 #sum=sum+A[i]
   ble $s1,$s3,loop #if s1<=s3 go to loop
```



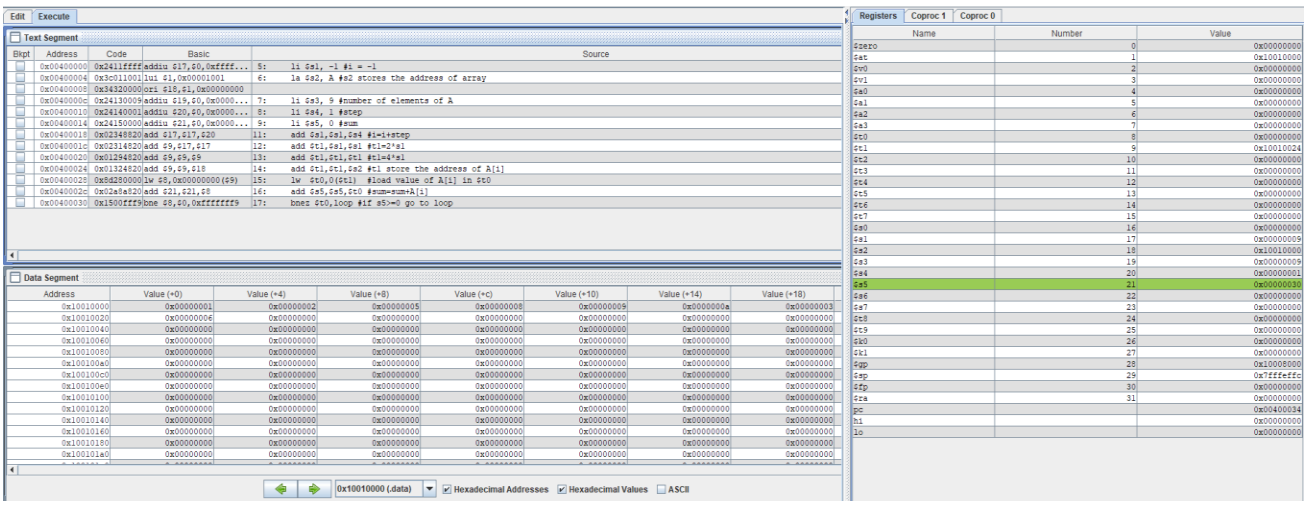
c.

```
.data
A: .word 1, 2, 5, 8, 9, 10, 3, 4, 6
.text
   li $s1, -1 #i = -1
   la $s2, A #s2 stores the address of array
   li $s3, 9 #number of elements of A
   li $s4, 1 #step
   li $s5, 0 #sum
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 the address of A[i]
   lw  $t0,0($t1)  #load value of A[i] in $t0
   add $s5,$s5,$t0 #sum=sum+A[i]
   bge $s5,$zero,loop #if s5>=0 go to loop
```



d.

```
.data
A: .word 1, 2, 5, 8, 9, 10, 3, 4, 6
.text
   li $s1, -1 #i = -1
   la $s2, A #s2 stores the address of array
   li $s3, 9 #number of elements of A
   li $s4, 1 #step
   li $s5, 0 #sum
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 the address of A[i]
   lw  $t0,0($t1)  #load value of A[i] in $t0
   add $s5,$s5,$t0 #sum=sum+A[i]
   bnez $t0,loop #if s5>=0 go to loop
```



Assignment 6:

```
.data
i: .word -1
A: .word -5, -10, 8, 9, 4
max: .word -1
.text
la $t0, i # load the address of i variable
lw $s0, 0($t0) # load the value of i to $s0
la $t1, A # load the address of array A
la $t6, max # load the address og max variable
lw $s6, 0($t6) # load the value of max to $s6
addi $s7, $s0, 5  # the number of elements in array = 5
loop:
addi $s0, $s0, 1 # i = i + 1
sw $s0, 0($t0)
sll $s1, $s0, 2 # s1 = i * 4
add $t2, $s1, $t1 # load the address of A[i]
lw $t2, 0($t2) # load the value of A[i] to $t2
slt $t3, $t2, $0 # t3 = 1 if (t2 < 0) else t3 = 0
beq $t3, $0, else
sub $t2, $0, $t2 # exp(A[i]) = -A[i]
j endif
else:
endif:
slt $t5, $s6, $t2 # t5 = 1 if (max < A[i]) else t5 = 0
beq $t5, $0, end # branch to end if (max >= A[i])
```

```
add $s6, $0, $t2 # max = A[i]
end:
slt $t4, $s0, $s7 # t4 = 1 if (i < 5) else t4 = 0
beq $t4, $0, endcmd # branch to endcmd if (i >= 5)
j loop # jump to loop
endcmd: # max = 10 (the value of register $s6 = a)
```