TRABALHO PRÁTICO 6

Aluno: João Madeira Carneiro Braga de Freitas
Matrícula: 800854

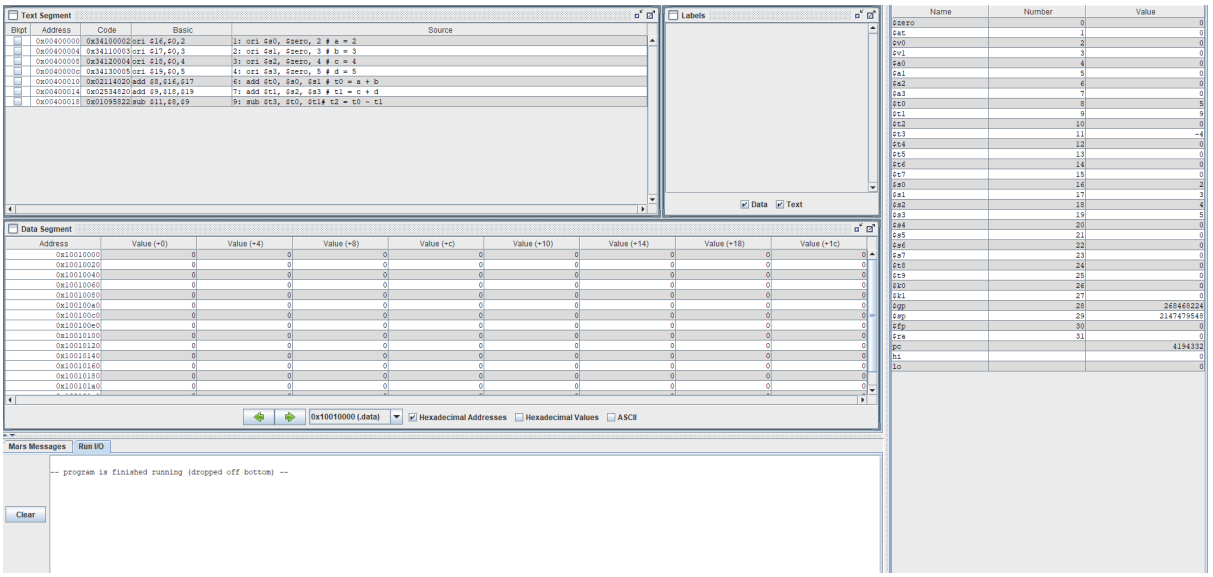
Parte 1 - Responda:

1) **A**
2) **C**
3) **A**
4) **C**
5) **D**
6) **C**
7) **A**
8) **D**
9) **A**
10) **A**
11) **B**
12) **D**
13) **C**
14) **C**
15) **B**
16) **C**
17) **A**
18) **A**

Programas:

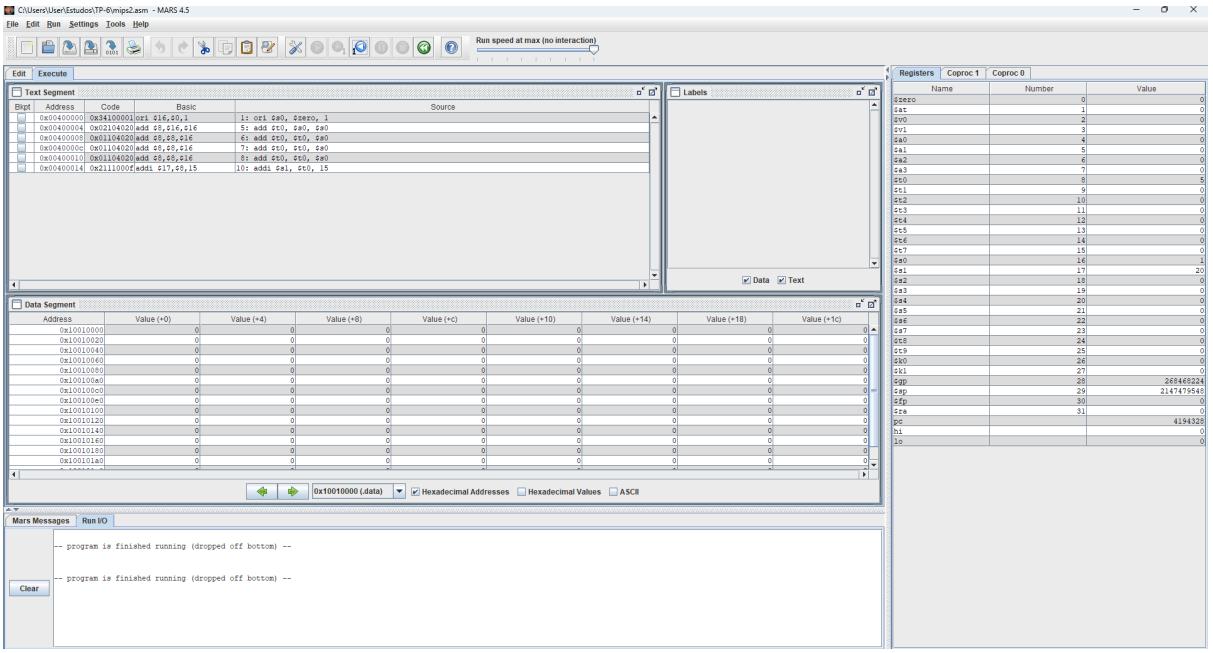
1)

```
ori $s0, $zero, 2 # a = 2
ori $s1, $zero, 3 # b = 3
ori $s2, $zero, 4 # c = 4
ori $s3, $zero, 5 # d = 5

add $t0, $s0, $s1 # t0 = a + b
add $t1, $s2, $s3 # t1 = c + d

sub $t3, $t0, $t1# t2 = t0 - t1
```



2)

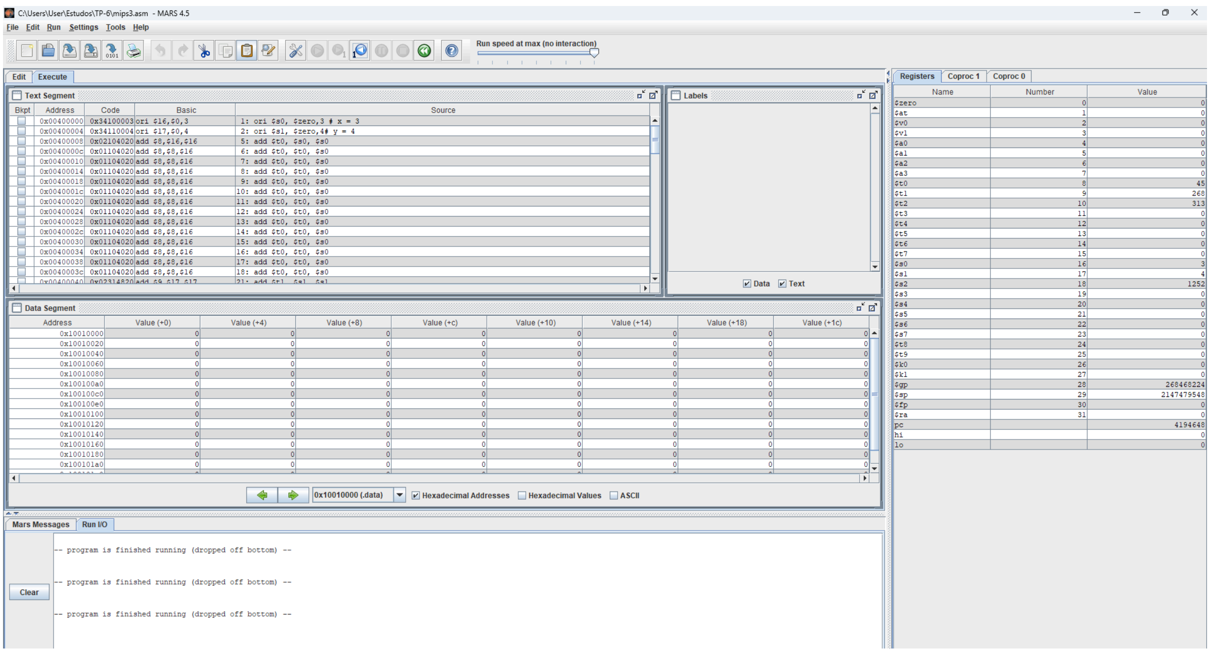
```
1  ori $s0, $zero, 1
2
3  # 5x = x + x + x + x + x
4
5  add $t0, $s0, $s0
6  add $t0, $t0, $s0
7  add $t0, $t0, $s0
8  add $t0, $t0, $s0
9
10 addi $s1, $t0, 15
11
```



3)

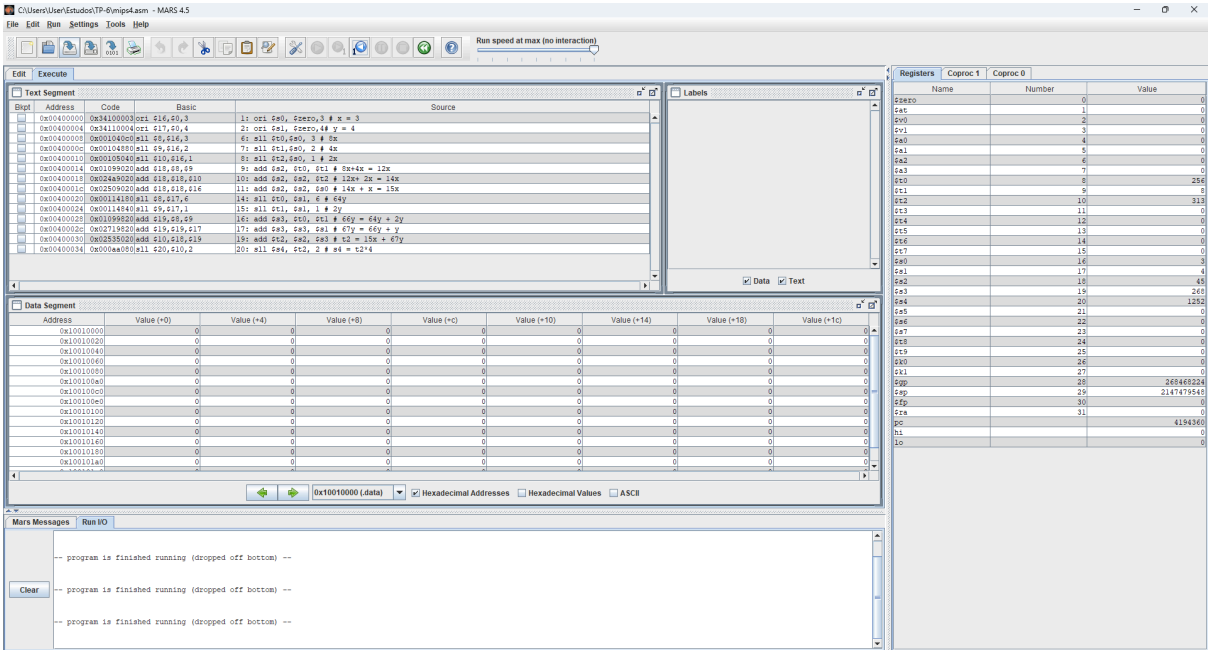
```
ori $s0, $zero,3 # x = 3
ori $s1, $zero,4# y = 4

# 15x
add $t0, $s0, $s0
add $t0, $t0, $s0
add $t0, $t0, $s0
add $t0, $t0, $s0
add $t0, $t0, $s0
add $t0, $t0, $s0
add $t0, $t0, $s0
add $t0, $t0, $s0
add $t0, $t0, $s0
add $t0, $t0, $s0
add $t0, $t0, $s0
add $t0, $t0, $s0
add $t0, $t0, $s0
add $t0, $t0, $s0

# 67y
add $t1, $s1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
add $t1, $t1, $s1
```

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


# 15x + 67y
add $t2, $t0, $t1

# (15x + 67y)*4
add $s2, $t2, $t2
add $s2, $s2, $t2
add $s2, $s2, $t2
```



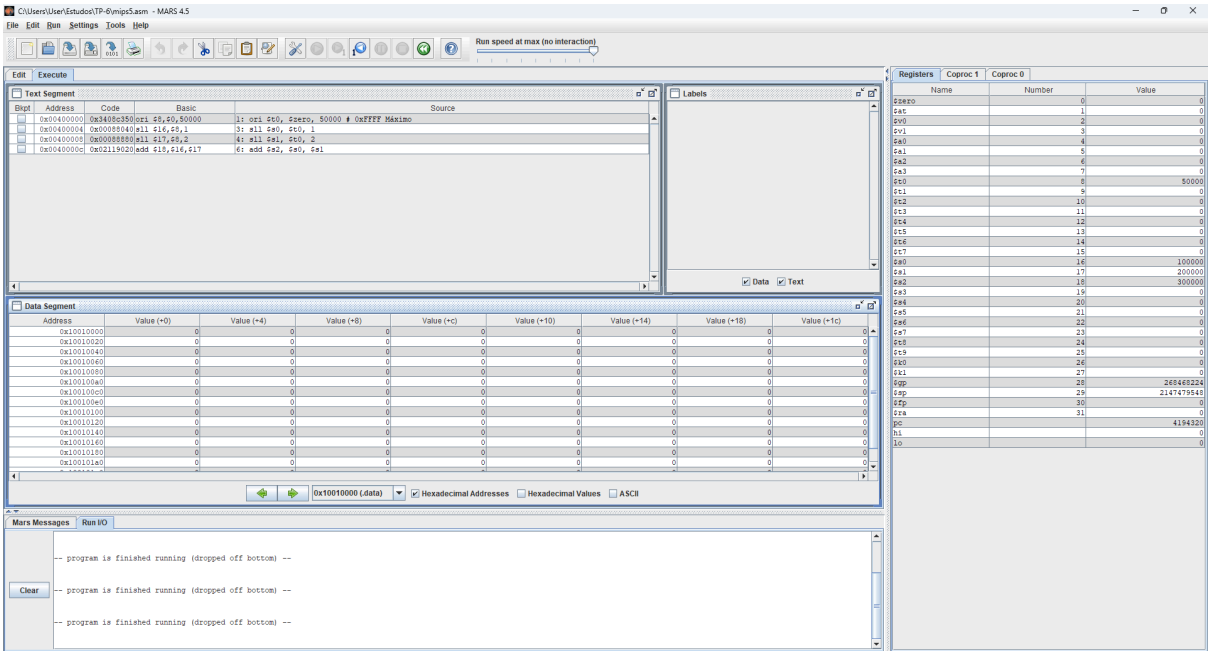
4)

```
1  ori $s0, $zero,3 # x = 3
2  ori $s1, $zero,4# y = 4
3
4  # 15x = 8x + 4x + 2x + x
5
6  sll $t0,$s0, 3 # 8x
7  sll $t1,$s0, 2 # 4x
8  sll $t2,$s0, 1 # 2x
9  add $s2, $t0, $t1 # 8x+4x = 12x
10 add $s2, $s2, $t2 # 12x+ 2x = 14x
11 add $s2, $s2, $s0 # 14x + x = 15x
12
13 # 67y = 64y + 2y + y
14 sll $t0, $s1, 6 # 64y
15 sll $t1, $s1, 1 # 2y
16 add $s3, $t0, $t1 # 66y = 64y + 2y
17 add $s3, $s3, $s1 # 67y = 66y + y
18
19 add $t2, $s2, $s3 # t2 = 15x + 67y
20 sll $s4, $t2, 2 # s4 = t2*4
21
22
```



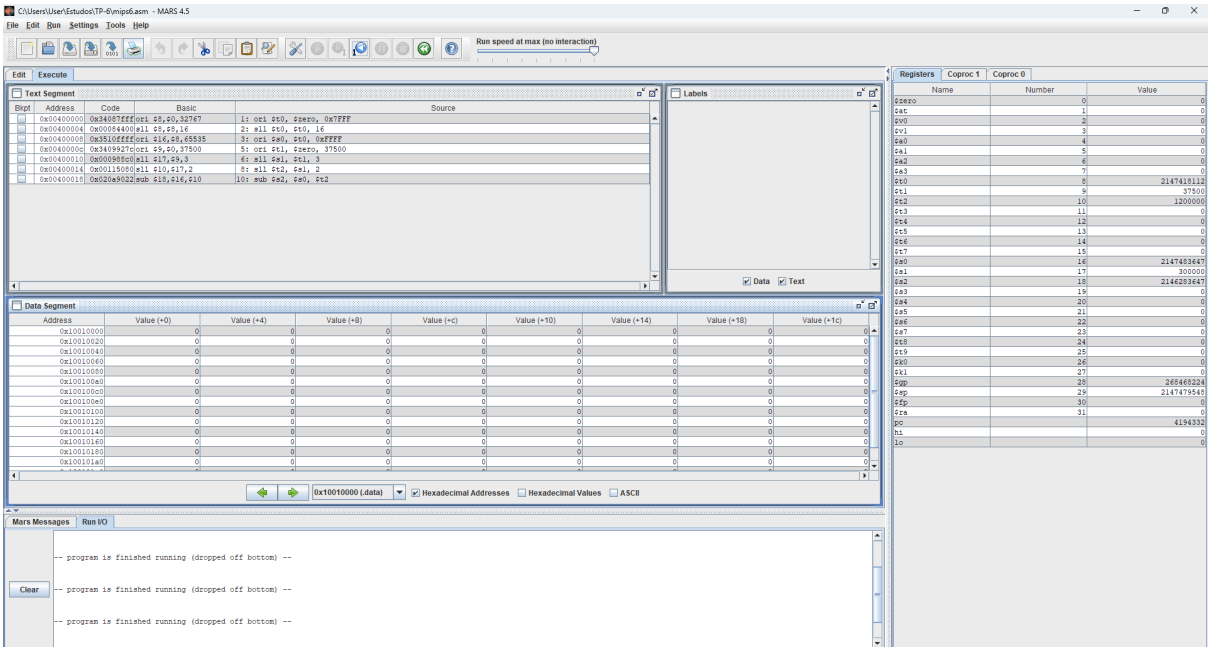
5)

```
1  ori $t0, $zero, 50000 # 0xFFFF Máximo
2
3  sll $s0, $t0, 1
4  sll $s1, $t0, 2
5
6  add $s2, $s0, $s1
7
```



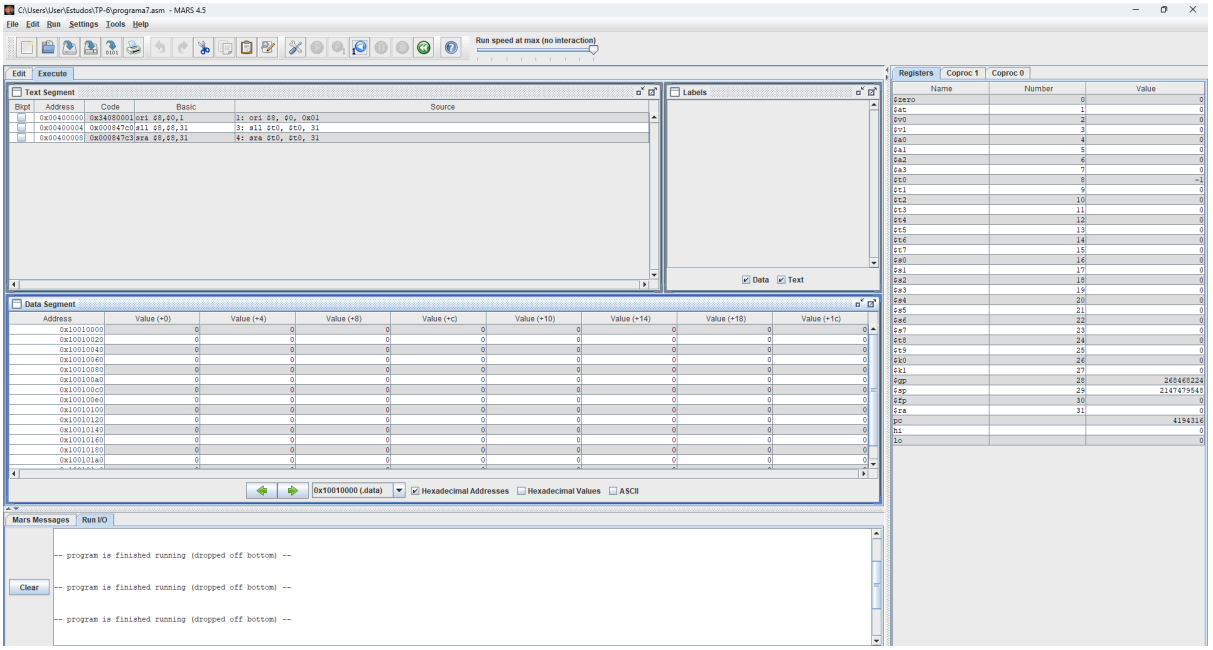
6)

```
1  ori $t0, $zero, 0x7FFF
2  sll $t0, $t0, 16
3  ori $s0, $t0, 0xFFFF
4
5  ori $t1, $zero, 37500
6  sll $s1, $t1, 3
7
8  sll $t2, $s1, 2
9
10 sub $s2, $s0, $t2
11
```



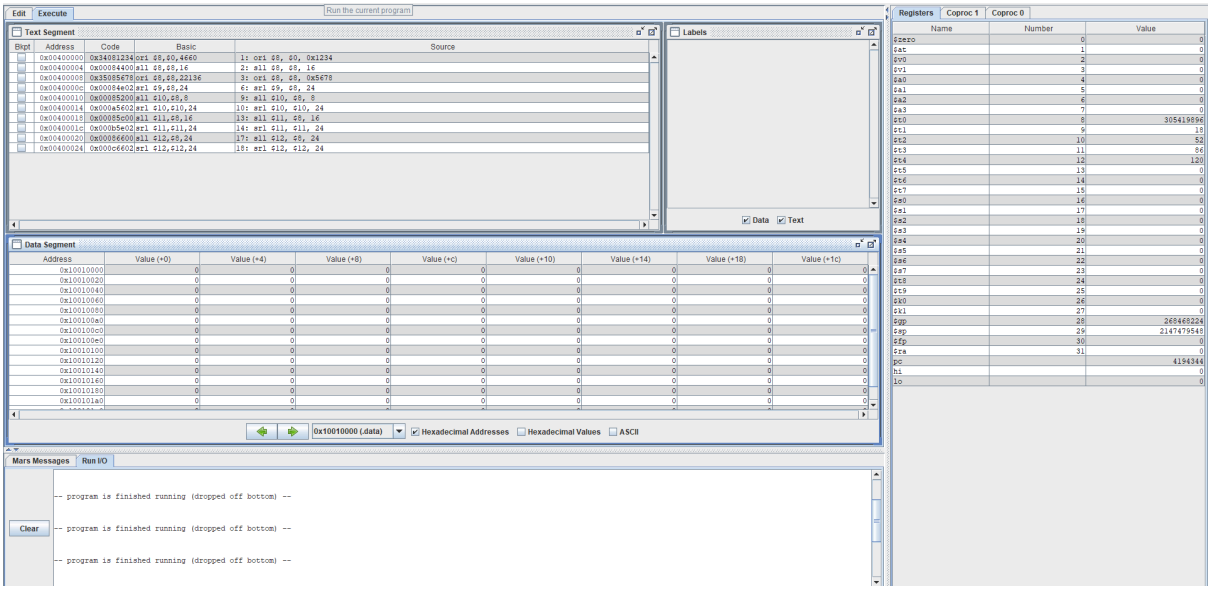
7)

```
1  ori $8, $0, 0x01
2
3  sll $t0, $t0, 31
4  sra $t0, $t0, 31
5
6
```



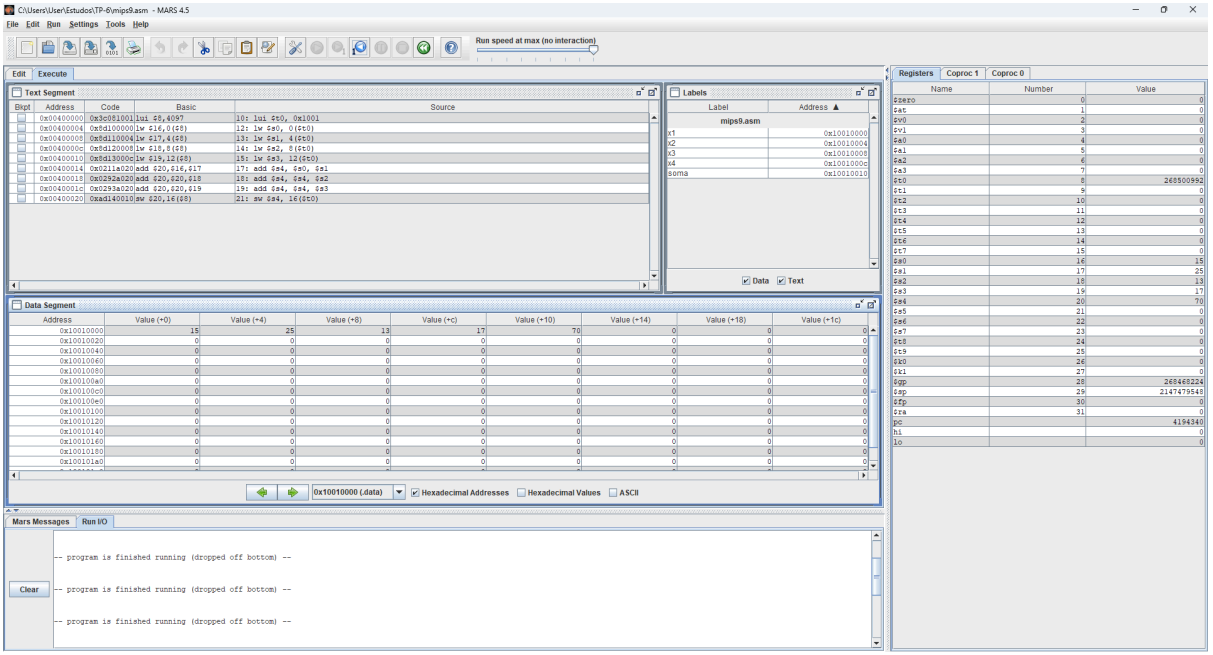
8)

```
1  ori $8, $0, 0x1234
2  sll $8, $8, 16
3  ori $8, $8, 0x5678
4
5  # $9 = 0X12
6  srl $9, $8, 24
7
8  # $10 = 0X34
9  sll $10, $8, 8
10 srl $10, $10, 24
11
12 # $11 = 0X56
13 sll $11, $8, 16
14 srl $11, $11, 24
15
16 # $12 = 0X78
17 sll $12, $8, 24
18 srl $12, $12, 24
19
```



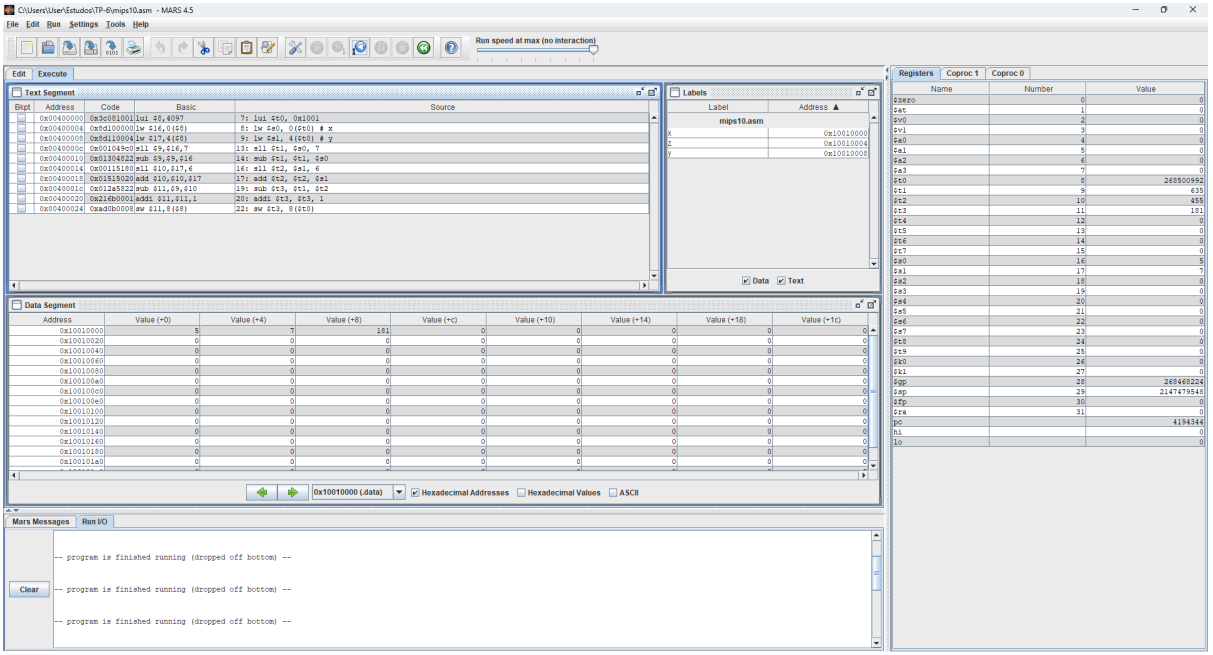
9)

```
1  .data
2  x1: .word 15
3  x2: .word 25
4  x3: .word 13
5  x4: .word 17
6  soma: .word -1
7
8  .text
9
10 lui $t0, 0x1001
11
12 lw $s0, 0($t0)
13 lw $s1, 4($t0)
14 lw $s2, 8($t0)
15 lw $s3, 12($t0)
16
17 add $s4, $s0, $s1
18 add $s4, $s4, $s2
19 add $s4, $s4, $s3
20
21 sw $s4, 16($t0)
22
```



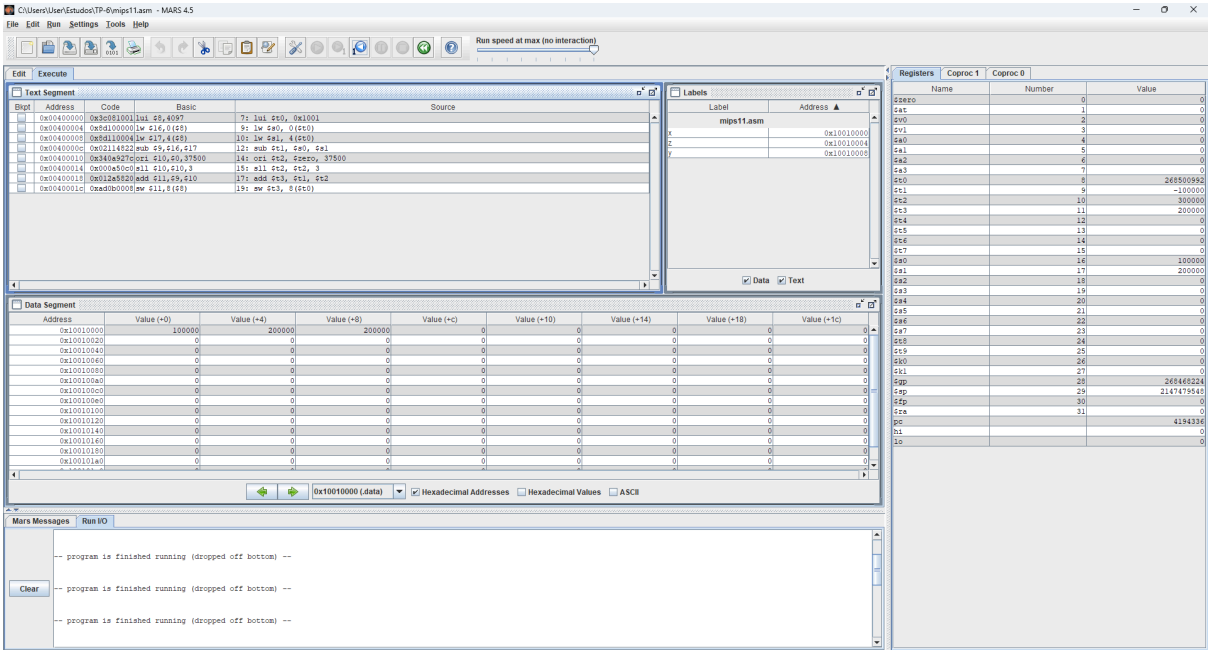
10)

```
.data
x: .word 5
z: .word 7
y: .word 0 # esse valor deverá ser sobrescrito após a execução do programa.

.text
lui $t0, 0x1001
lw $s0, 0($t0) # x
lw $s1, 4($t0) # y

#127x, 65z

sll $t1, $s0, 7
sub $t1, $t1, $s0

sll $t2, $s1, 6
add $t2, $t2, $s1

sub $t3, $t1, $t2
addi $t3, $t3, 1

sw $t3, 8($t0)
```



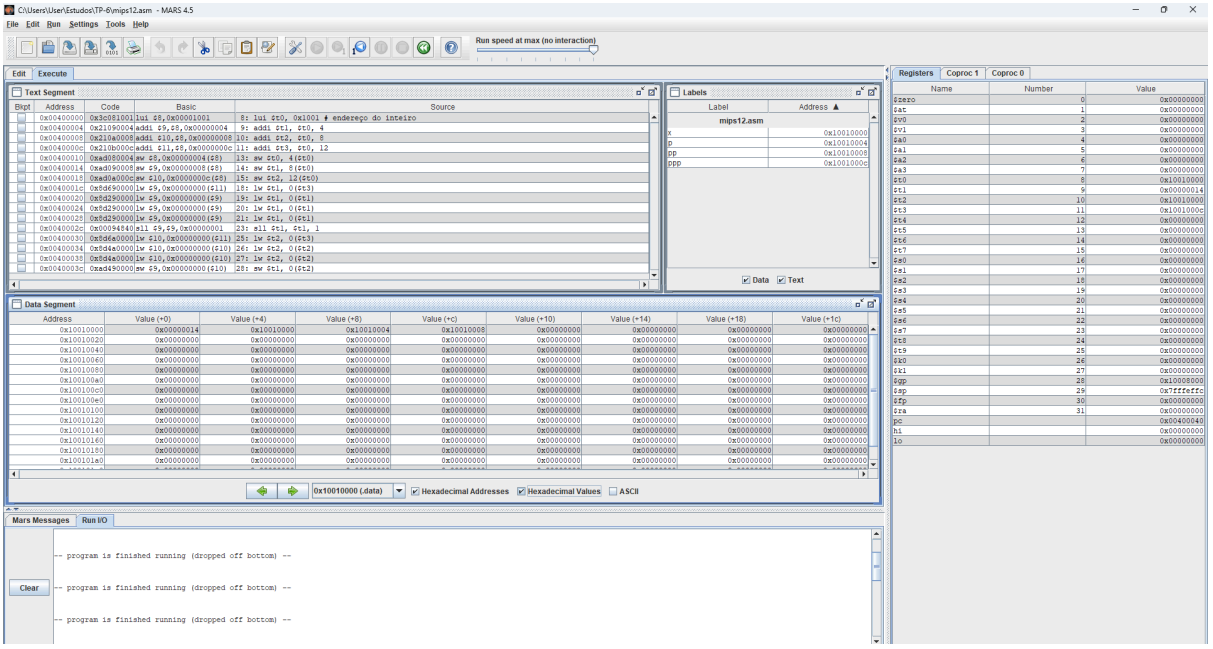
11)

```
.data
x: .word 100000
z: .word 200000
y: .word 0 # esse valor deverá ser sobrescrito após a execução do programa.

.text
lui $t0, 0x1001

lw $s0, 0($t0)
lw $s1, 4($t0)

sub $t1, $s0, $s1

ori $t2, $zero, 37500
sll $t2, $t2, 3

add $t3, $t1, $t2

sw $t3, 8($t0)
```



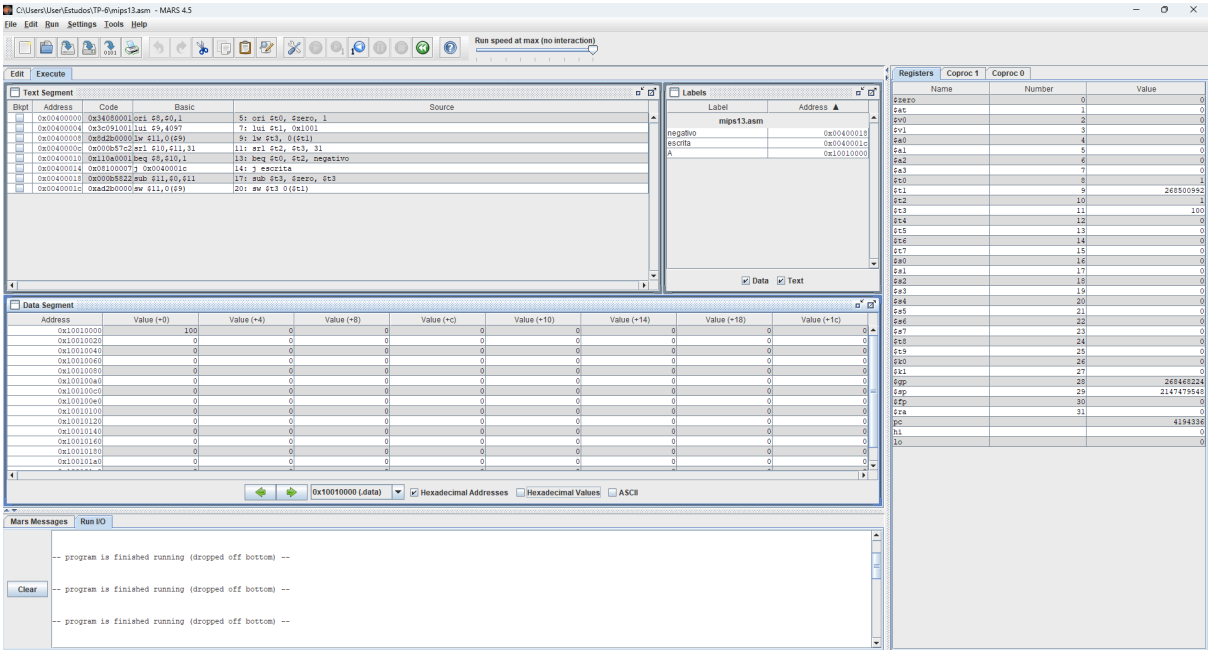
12)

```
.data
x: .word 10
p: .word 0
pp: .word 0
ppp: .word 0

.text
lui $t0, 0x1001 # endereço do inteiro
addi $t1, $t0, 4
addi $t2, $t0, 8
addi $t3, $t0, 12

sw $t0, 4($t0)
sw $t1, 8($t0)
sw $t2, 12($t0)


lw $t1, 0($t3)
lw $t1, 0($t1)
lw $t1, 0($t1)
lw $t1, 0($t1)

sll $t1, $t1, 1

lw $t2, 0($t3)
lw $t2, 0($t2)
lw $t2, 0($t2)
sw $t1, 0($t2)
```



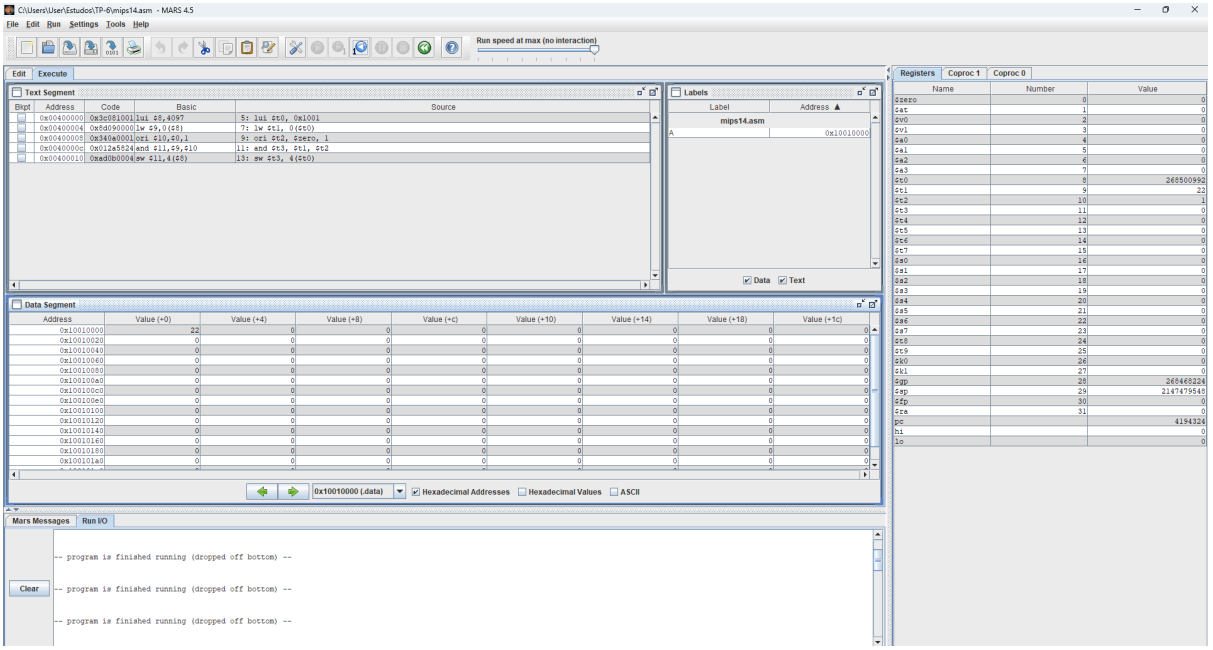
13)

```
.data
A: .word -100

.text
ori $t0, $zero, 1

lui $t1, 0x1001

lw $t3, 0($t1)

srl $t2, $t3, 31

beq $t0, $t2, negativo
j escrita

negativo:
sub $t3, $zero, $t3

escrita:
sw $t3 0($t1)
```



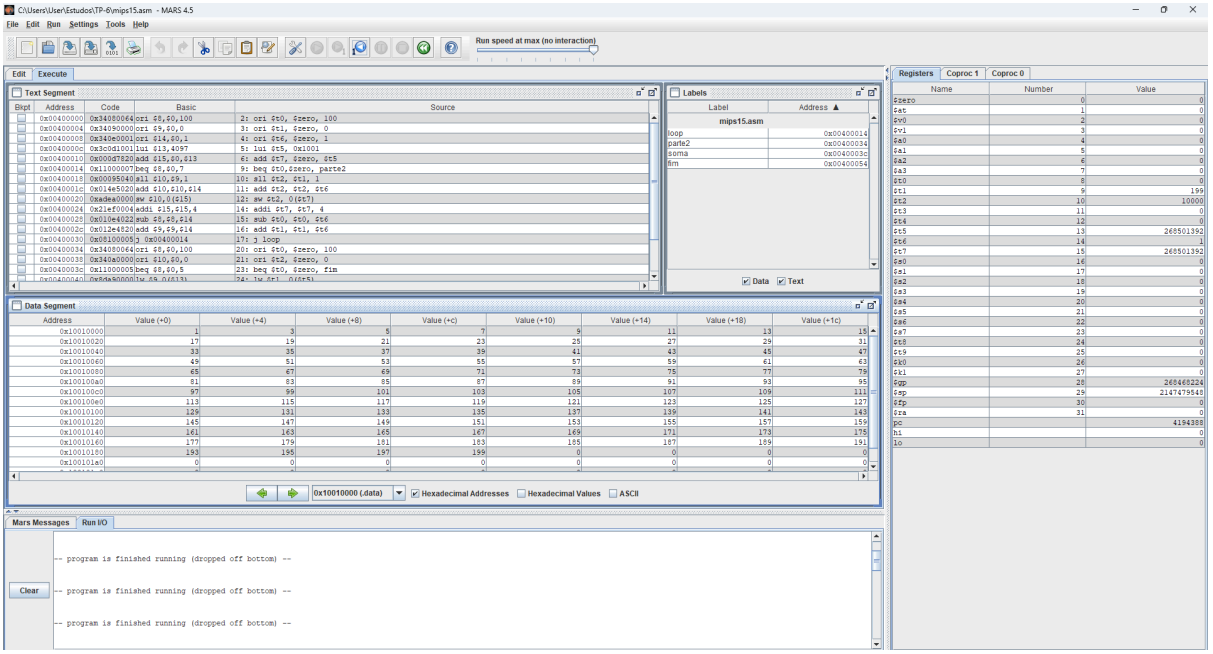
14)

```
1  .data
2  A: .word 22
3
4  .text
5  lui $t0, 0x1001
6
7  lw $t1, 0($t0)
8
9  ori $t2, $zero, 1
10
11 and $t3, $t1, $t2
12
13 sw $t3, 4($t0)
14
15
16
17
```



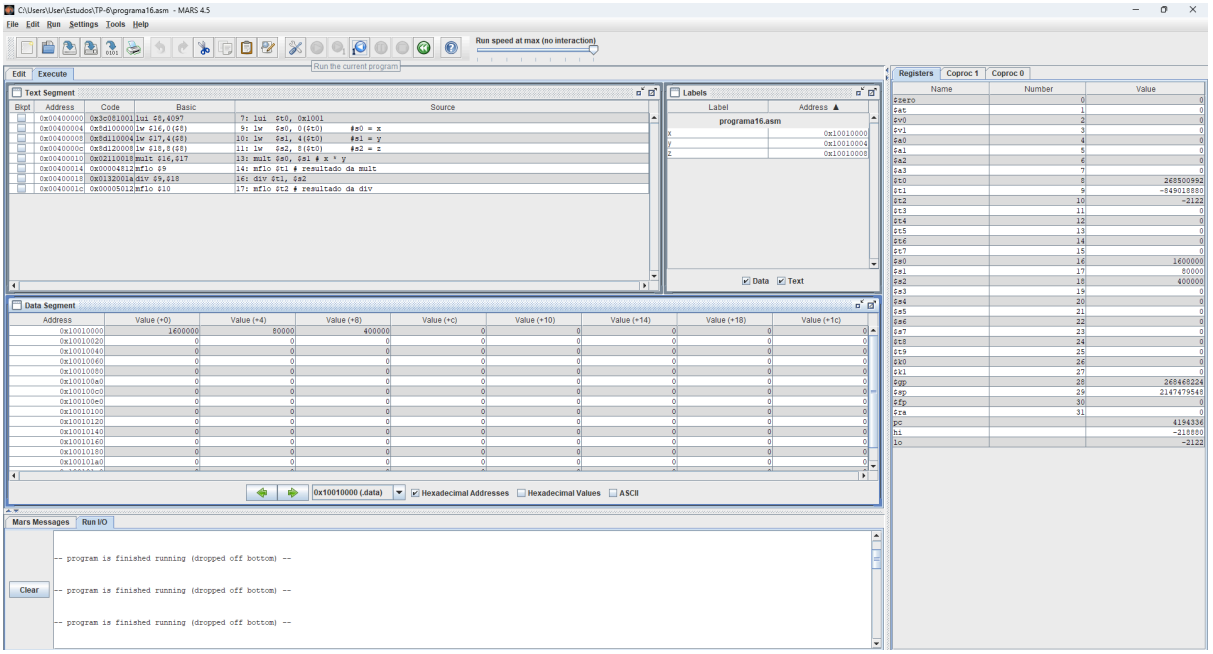
15)

```
1  .text
2  ori $t0, $zero, 100
3  ori $t1, $zero, 0
4  ori $t6, $zero, 1
5  lui $t5, 0x1001
6  add $t7, $zero, $t5
7
8  loop:
9  beq $t0,$zero, parte2
10 sll $t2, $t1, 1
11 add $t2, $t2, $t6
12 sw $t2, 0($t7)
13
14 addi $t7, $t7, 4
15 sub $t0, $t0, $t6
16 add $t1, $t1, $t6
17 j loop
18
19 parte2:
20 ori $t0, $zero, 100
21 ori $t2, $zero, 0
22 soma:
23 beq $t0, $zero, fim
24 lw $t1, 0($t5)
25 add $t2, $t1, $t2
26
27 addi $t5, $t5, 4
28 sub $t0, $t0, $t6
29 j soma
30
31 fim:
32
```



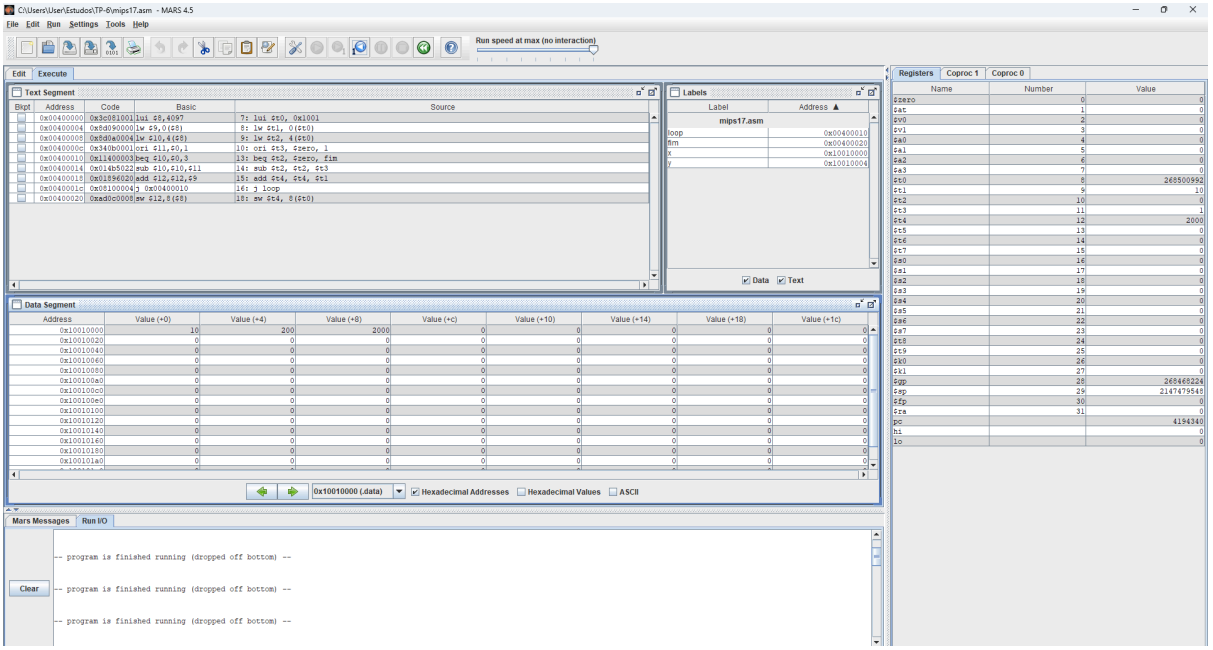
16)

```
1  .data
2  x:.word 0x186A00
3  y:.word 0x13880
4  z:.word 0x61A80
5
6  .text
7  lui  $t0, 0x1001
8
9  lw   $s0, 0($t0)       #s0 = x
10 lw   $s1, 4($t0)       #s1 = y
11 lw   $s2, 8($t0)       #s2 = z
12
13 mult $s0, $s1 # x * y
14 mflo $t1 # resultado da mult
15
16 div $t1, $s2
17 mflo $t2 # resultado da div
18
```



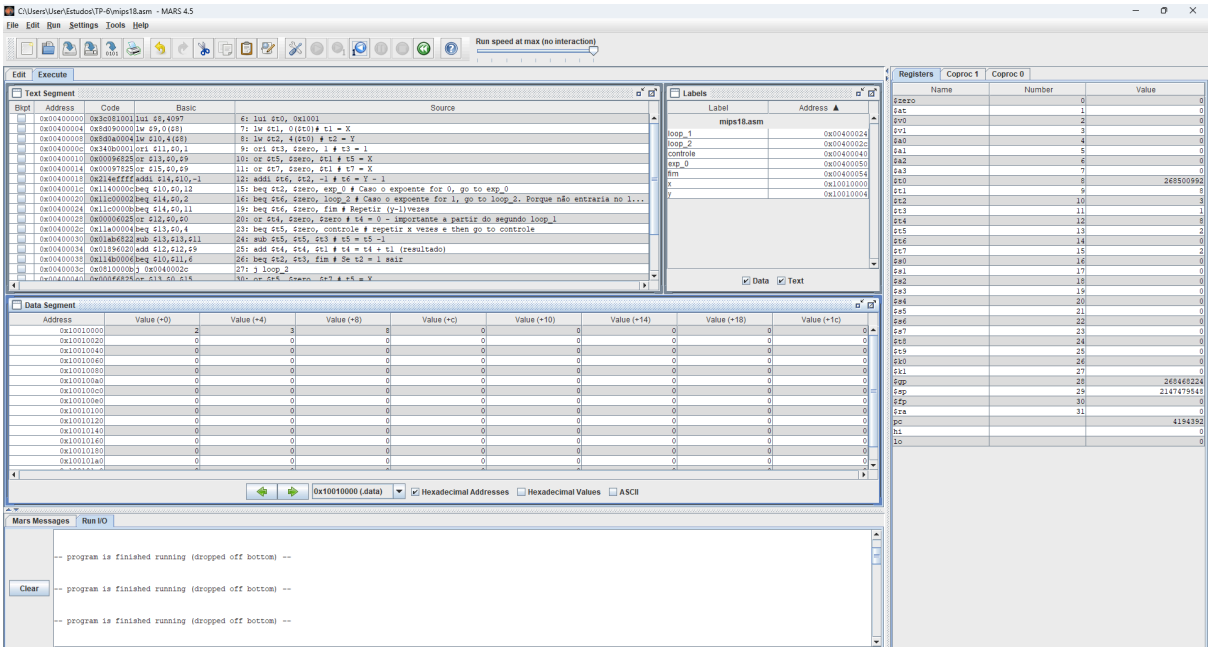
17)

```
1  .data
2  x: .word 10
3  y: .word 200
4
5  .text
6
7  lui $t0, 0x1001
8  lw $t1, 0($t0)
9  lw $t2, 4($t0)
10 ori $t3, $zero, 1
11
12 loop:
13 beq $t2, $zero, fim
14 sub $t2, $t2, $t3
15 add $t4, $t4, $t1
16 j loop
17 fim:
18 sw $t4, 8($t0)
19
```



18)

```
.data
x: .word 2
y: .word 3

.text
lui $t0, 0x1001
lw $t1, 0($t0)# t1 = X
lw $t2, 4($t0) # t2 = Y
ori $t3, $zero, 1 # t3 = 1
or $t5, $zero, $t1 # t5 = X
or $t7, $zero, $t1 # t7 = X
addi $t6, $t2, -1 # t6 = Y - 1


beq $t2, $zero, exp_0 # Caso o expoente for 0, go to exp_0
beq $t6, $zero, loop_2 # Caso o expoente for 1, go to loop_2. Porque não entraria no loop_1, porque t6 = 0

loop_1:
beq $t6, $zero, fim # Repetir (y-1)vezes
or $t4, $zero, $zero # t4 = 0 - importante a partir do segundo loop_1

loop_2:
beq $t5, $zero, controle # repetir x vezes e then go to controle
sub $t5, $t5, $t3 # t5 = t5 -1
add $t4, $t4, $t1 # t4 = t4 + t1 (resultado)

beq $t2, $t3, fim # Se t2 = 1 sair
j loop_2

controle:
or $t5, $zero, $t7 # t5 = X
or $t1, $zero, $t4 # t1 = t4
sub $t6, $t6, $t3 # t6 = t6 - 1
j loop_1


exp_0:
ori $t4, $zero, 1 # resultado = 1

fim:
sw $t4, 8($t0) # Escrever o resultado

```



Responda:

1) **C**
2) **B**
3) **A**
4) **C**
5) **B**
6) **A**
7) **A**
8) **C**
9) **A**
10) **A**

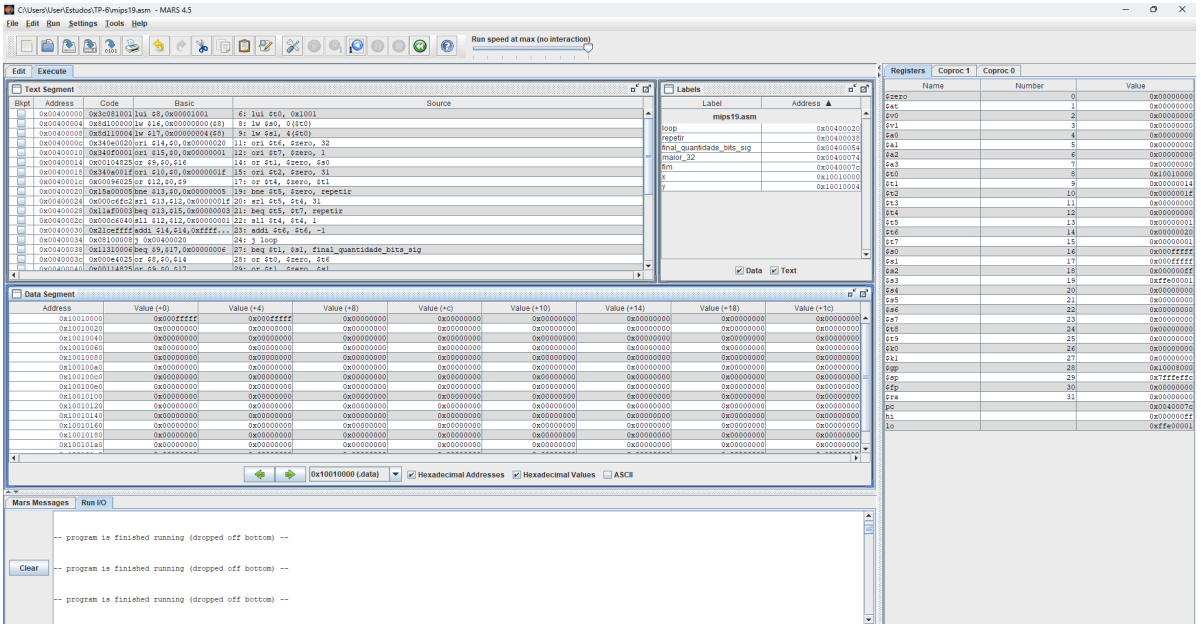
Continuação Programas

19)

```
.data
x: .word 0xFFFFF
y: .word 0xFFFFF

.text
lui $t0, 0x1001

lw $s0, 0($t0)
lw $s1, 4($t0)

ori $t6, $zero, 32
ori $t7, $zero, 1

or $t1, $zero, $s0
ori $t2, $zero, 31

or $t4, $zero, $t1
loop:
bne $t5, $zero, repetir
srl $t5, $t4, 31
beq $t5, $t7, repetir
sll $t4, $t4, 1
addi $t6, $t6, -1
j loop
```

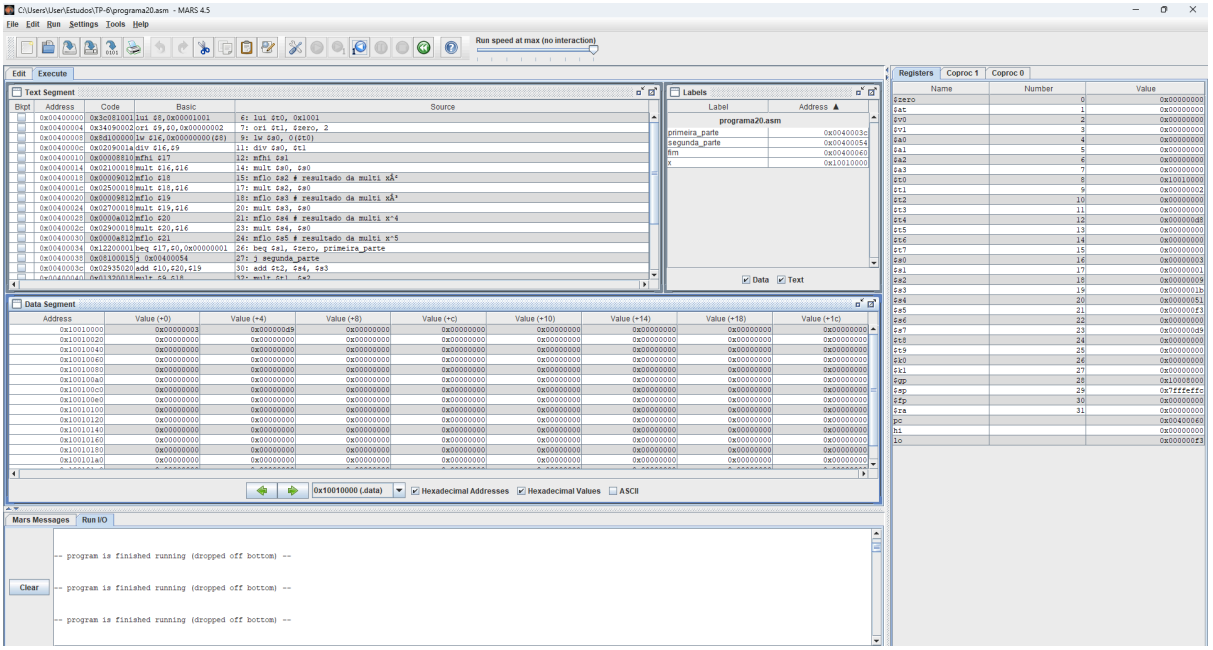
```
repetir:
beq $t1, $s1, final_quantidade_bits_sig
or $t0, $zero, $t6
or $t1, $zero, $s1
or $t4, $zero, $t1
or $t5, $zero, $zero
ori $t6, $zero, 32

j loop

final_quantidade_bits_sig:
or $t1, $zero, $t6
ori $t6, $zero, 32


mult $s0, $s1

add $t4, $t0, $t1

slt $t4, $t4, $t6

beq $t4, $zero, maior_32
mflo $s2
j fim


```

```
maior_32:
mfhi $s2
mflo $s3

fim:

```



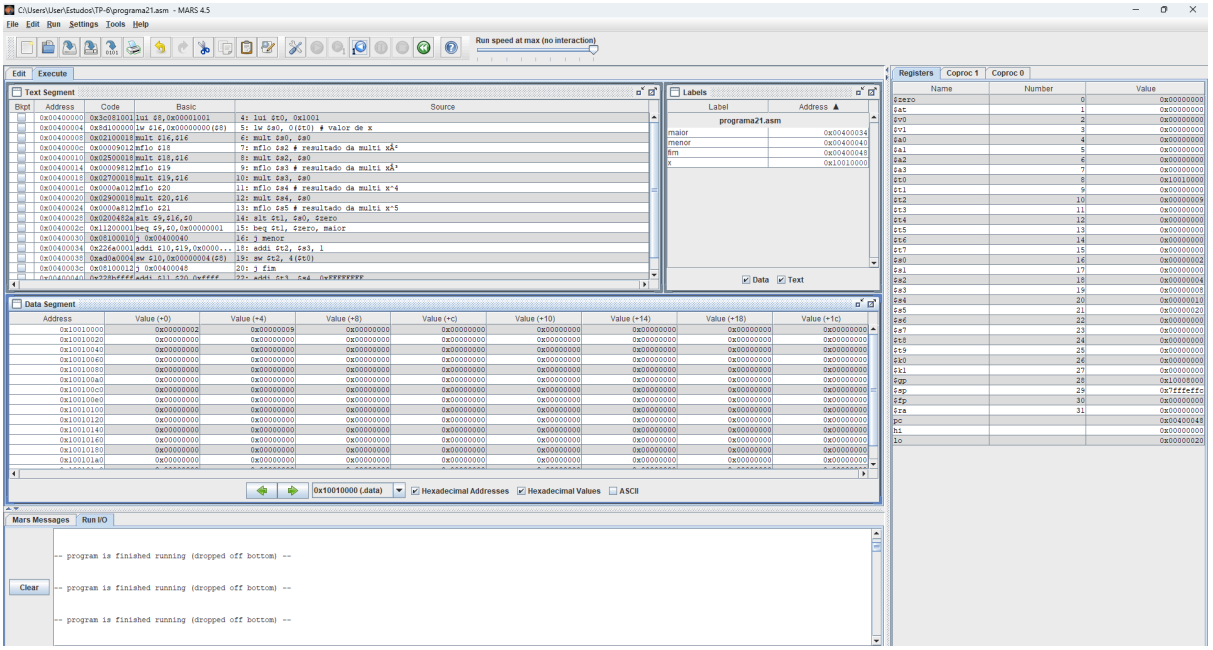
20)

```
.data
x: .word 3

.text

lui $t0, 0x1001
ori $t1, $zero, 2

lw $s0, 0($t0)

div $s0, $t1
mfhi $s1

mult $s0, $s0
mflo $s2 # resultado da multi xÂ²

mult $s2, $s0
mflo $s3 # resultado da multi xÂ³

mult $s3, $s0
mflo $s4 # resultado da multi x^4

mult $s4, $s0
mflo $s5 # resultado da multi x^5

beq $s1, $zero, primeira_parte
j segunda_parte

primeira_parte:
add $t2, $s4, $s3

mult $t1, $s2
mflo $s6 # 2*xÂ²

sub $t3, $t2, $s6
sw $t3, 4($t0)
j fim

segunda_parte:
sub $t4, $s5, $s3
addi $s7, $t4, 1
sw $s7, 4($t0)

fim:

```



21)

```
.data
x: .word 2
.text
lui $t0, 0x1001
lw $s0, 0($t0) # valor de x
mult $s0, $s0
mflo $s2 # resultado da multi xÂ²
mult $s2, $s0
mflo $s3 # resultado da multi xÂ³
mult $s3, $s0
mflo $s4 # resultado da multi x^4
mult $s4, $s0
mflo $s5 # resultado da multi x^5
slt $t1, $s0, $zero
beq $t1, $zero, maior
j menor
maior:
addi $t2, $s3, 1
sw $t2, 4($t0)
j fim
menor:
addi $t3, $s4, 0xFFFFFFFF
sw $t3, 4($t0)
fim:
```


C:\Users\User\Estudos\TP-6\programa21.asm - MARS 4.5
File Edit Run Settings Tools Help
Run speed at max (no interaction)
Edit Execute
Text Segment
Bkpt Address Code Basic Source
0x00400000 0x3c081001 lui $8,0x00001001 4: lui $t0, 0x1001
0x00400004 0x8d100000 lw $16,0x00000000($8) 5: lw $a0, 0($t0) # valor de x
0x00400008 0x02100018 mult $16,$16 6: mult $a0, $a0
0x0040000c 0x00009012 mflo $18 7: mflo $a2 # resultado da multi xÂ²
0x00400010 0x02500018 mult $18,$16 8: mult $a2, $a0
0x00400014 0x00009812 mflo $19 9: mflo $a3 # resultado da multi xÂ³
0x00400018 0x02700018 mult $19,$16 10: mult $a3, $a0
0x0040001c 0x0000a012 mflo $20 11: mflo $a4 # resultado da multi x^4
0x00400020 0x02900018 mult $20,$16 12: mult $a4, $a0
0x00400024 0x0000a812 mflo $21 13: mflo $a5 # resultado da multi x^5
0x00400028 0x0200482a slt $9,$16,$0 14: slt $t1, $a0, $zero
0x0040002c 0x11200001 beq $9,$0,0x00000001 15: beq $t1, $zero, maior
0x00400030 0x08100010 j 0x00400040 16: j menor
0x00400034 0x226a0001 addi $10,$19,0x0000... 18: addi $t2, $a3, 1
0x00400038 0xadda0004 sw $10,0x00000004($8) 19: sw $t2, 4($t0)
0x0040003c 0x08100012 j 0x00400048 20: j fim
Labels
Label Address
programa21.asm
maior 0x00400034
menor 0x00400040
fim 0x00400048
x 0x10010000
Data Text
Registers Coproc 1 Coproc 0
Name Number Value
$zero 0 0x00000000
$at 1 0x00000000
$v0 2 0x00000000
$v1 3 0x00000000
$a0 4 0x00000000
$a1 5 0x00000000
$a2 6 0x00000000
$a3 7 0x00000000
$t0 8 0x10010000
$t1 9 0x00000000
$t2 10 0x00000009
$t3 11 0x00000000
$t4 12 0x00000000
$t5 13 0x00000000
$t6 14 0x00000000
$t7 15 0x00000000
$s0 16 0x00000002
$s1 17 0x00000000
$s2 18 0x00000004
$s3 19 0x00000008
$s4 20 0x00000010
$s5 21 0x00000020
$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
hi 0x00000000
lo 0x00000020
Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+c) Value (+10) Value (+14) Value (+18) Value (+1c)
0x10010000 0x00000002 0x00000009 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
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
0x10010160 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010180 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100101a0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII
Mars Messages Run I/O
-- program is finished running (dropped off bottom) --
Clear
-- program is finished running (dropped off bottom) --
-- program is finished running (dropped off bottom) --