

Exercício Prático 06 - MIPS

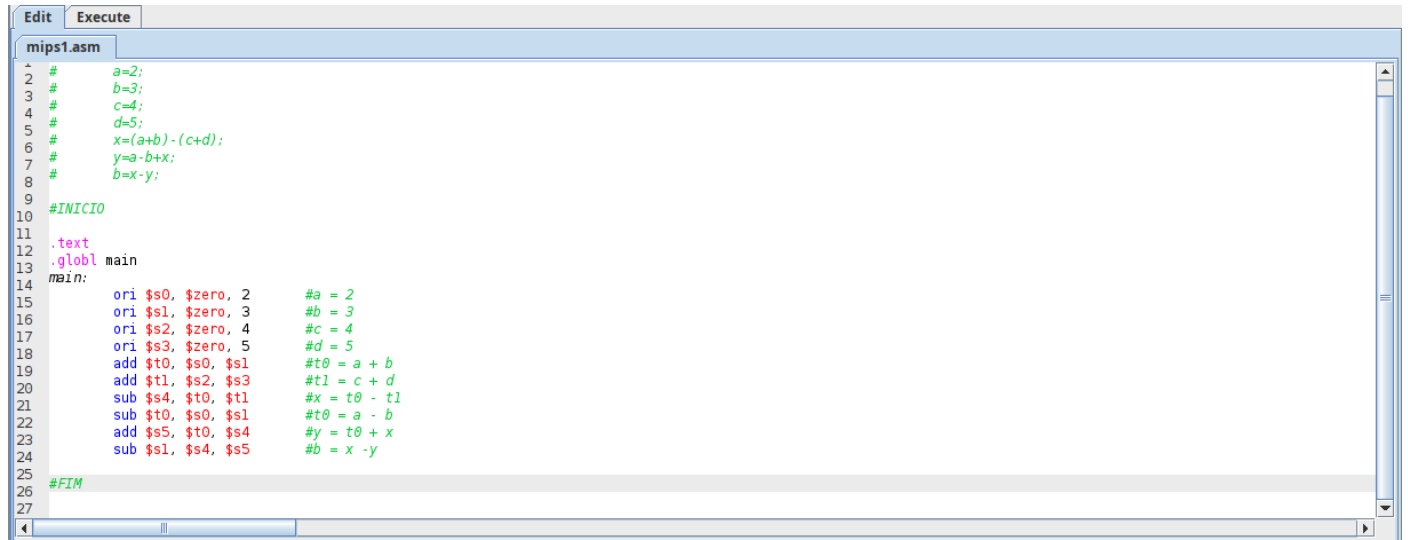
806454 - Yago Almeida Melo

Parte 1 - Questões

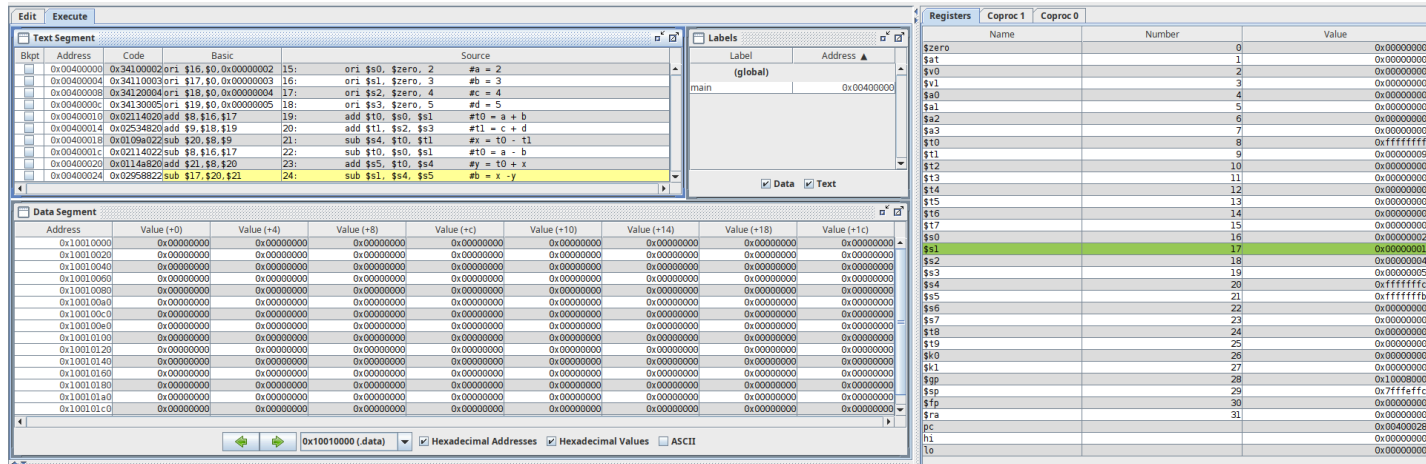
- 1) A. um arquivo de texto que contém instruções de linguagem de programação.
- 2) B. uma parte do processador que possui um padrão de bits.
- 3) A. #
- 4) C. 32
- 5) D. parte do processador que contém o endereço da próxima instrução de máquina para ser obtida.
- 6) C. 4
- 7) D. uma declaração que diz o montador algo sobre o que o programador quer, mas não corresponde diretamente a uma instrução de máquina.
- 8) D. um nome usado no código-fonte em linguagem assembly para um local na memória.
- 9) B. 0x00400000
- 10) A. operando imediato.
- 11) B. operação bitwise.
- 12) D. Cada um dos registradores deve possuir 32 bits.
- 13) B. Os dados são estendidos em zero à esquerda por 16 bits.
- 14) C. ori \$5, \$0, 48
- 15) B. Sim
- 16) D. andi \$8, \$8, 0xFF
- 17) A. Todos os bits em zero
- 18) A. Não. Diferentes instruções de máquina possuem campos diferentes.

Parte 2 - Implementação no MARS

1)



```
mips1.asm
1 #      a=2;
2 #      b=3;
3 #      c=4;
4 #      d=5;
5 #      x=(a+b) - (c+d);
6 #      y=a-b+x;
7 #      b=x-y;
8
9
10 #INICIO
11
12 .text
13 .globl main
14 main:
15     ori $s0, $zero, 2      #a = 2
16     ori $s1, $zero, 3      #b = 3
17     ori $s2, $zero, 4      #c = 4
18     ori $s3, $zero, 5      #d = 5
19     add $t0, $s0, $s1      #t0 = a + b
20     add $t1, $s2, $s3      #t1 = c + d
21     sub $s4, $t0, $t1      #x = t0 - t1
22     sub $t0, $s0, $s1      #t0 = a - b
23     add $s5, $t0, $s4      #y = t0 + x
24     sub $s1, $s4, $s5      #b = x - y
25
26 #FIM
27
```



The screenshot displays the MARS MIPS simulator interface during execution. The **Text Segment** panel shows the assembly code with addresses and hex values. The **Labels** panel shows the global label `main` at address `0x00400000`. The **Registers** panel shows the state of the MIPS registers, with `$s1` highlighted at address `0x00000017`.

| Register | Value |
|----------|------------|
| \$zero | 0x00000000 |
| \$at | 0x00000001 |
| \$v0 | 0x00000002 |
| \$v1 | 0x00000003 |
| \$a0 | 0x00000004 |
| \$a1 | 0x00000005 |
| \$a2 | 0x00000006 |
| \$a3 | 0x00000007 |
| \$t0 | 0xffffffff |
| \$t1 | 0x00000009 |
| \$t2 | 0x0000000a |
| \$t3 | 0x0000000b |
| \$t4 | 0x0000000c |
| \$t5 | 0x0000000d |
| \$t6 | 0x0000000e |
| \$t7 | 0x0000000f |
| \$s0 | 0x00000010 |
| \$s1 | 0x00000017 |
| \$s2 | 0x00000018 |
| \$s3 | 0x00000019 |
| \$s4 | 0xffffffff |
| \$s5 | 0x00000020 |
| \$s6 | 0x00000021 |
| \$s7 | 0x00000022 |
| \$s8 | 0x00000023 |
| \$s9 | 0x00000024 |
| \$s10 | 0x00000025 |
| \$s11 | 0x00000026 |
| \$s12 | 0x00000027 |
| \$s13 | 0x00000028 |
| \$s14 | 0x00000029 |
| \$s15 | 0x0000002a |
| \$s16 | 0x0000002b |
| \$s17 | 0x0000002c |
| \$s18 | 0x0000002d |
| \$s19 | 0x0000002e |
| \$s20 | 0x0000002f |
| \$s21 | 0x00000030 |
| \$s22 | 0x00000031 |
| \$s23 | 0x00000032 |
| \$s24 | 0x00000033 |
| \$s25 | 0x00000034 |
| \$s26 | 0x00000035 |
| \$s27 | 0x00000036 |
| \$s28 | 0x00000037 |
| \$s29 | 0x00000038 |
| \$s30 | 0x00000039 |
| \$s31 | 0x0000003a |
| \$pc | 0x00400028 |
| \$hi | 0x00000000 |
| \$lo | 0x00000000 |

2)

```

1  # Programa 2
2  # x=1;;
3  # y = 5*x + 15;;
4
5  #INICIO
6
7  .text
8  .globl main
9  main:
10     ori $s0, $zero, 1      #x = 1
11     add $t0, $s0, $s0      #t0 = 2x
12     add $t1, $t0, $t0      #t1 = 4x
13     add $s1, $t1, $s0      #y = 4x + x
14     addi $s1, $s1, 15      #y = 5x + 15
15 #FIM
16

```

Line: 16 Column: 1 Show Line Numbers

| Text Segment | | | | | | | | Labels | |
|--------------|------------|------------|--------------------------|--------|----------------------|--------------|--|--------|----------|
| Block | Address | Code | Basic | Source | | | | Label | Address |
| | 0x00400000 | 0x34100001 | ori \$t6,\$0,0x00000001 | 10: | ori \$s0, \$zero, 1 | #x = 1 | | | (global) |
| | 0x00400004 | 0x02104020 | add \$t6,\$t6,\$t6 | 11: | add \$t0, \$s0, \$s0 | #t0 = 2x | | | |
| | 0x00400008 | 0x01084820 | add \$t6,\$t6,\$t6 | 12: | add \$t1, \$t0, \$t0 | #t1 = 4x | | | |
| | 0x0040000c | 0x01308820 | add \$t7,\$t7,\$t6 | 13: | add \$s1, \$t1, \$s0 | #y = 4x + x | | | |
| | 0x00400010 | 0x2231000f | addi \$t7,\$t7,0x0000... | 14: | addi \$s1, \$s1, 15 | #y = 5x + 15 | | | |

| Data Segment | | | | | | | |
|--------------|------------|------------|------------|------------|-------------|-------------|-------------|
| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) |
| 0x10010000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010004 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010008 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001000c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010010 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010014 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010018 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001001c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

| | | |
|--------|----|------------|
| \$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 | 0x00000002 |
| \$t1 | 9 | 0x00000004 |
| \$t2 | 10 | 0x00000000 |
| \$t3 | 11 | 0x00000000 |
| \$t4 | 12 | 0x00000000 |
| \$t5 | 13 | 0x00000000 |
| \$t6 | 14 | 0x00000000 |
| \$t7 | 15 | 0x00000000 |
| \$s0 | 16 | 0x00000001 |
| \$s1 | 17 | 0x00000014 |
| \$s2 | 18 | 0x00000000 |
| \$s3 | 19 | 0x00000000 |
| \$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 | 0x10000000 |
| \$fp | 29 | 0x7fffffff |
| \$tp | 30 | 0x00000000 |
| \$ra | 31 | 0x00000000 |
| pc | | 0x00400014 |
| hi | | 0x00000000 |
| lo | | 0x00000000 |

3)

```

2  #INICIO
3
4  #      Programa 3
5  #      x=3;
6  #      y=4;
7  #      z=(15*x + 67*y)*4
8
9  .text
10 .globl main
11 main:
12     ori $s0, $zero, 3      #x = 3
13     ori $s1, $zero, 4      #y = 4
14     add $t0, $s0, $s0      #t0 = 2x
15     add $t1, $t0, $t0      #t1 = 4x
16     add $t2, $t1, $t1      #t2 = 8x
17     add $t3, $t2, $t2      #t3 = 16x
18     sub $s2, $t3, $s0      #z = 15x
19     add $t0, $s1, $s1      #t0 = 2y
20     add $t1, $t0, $t0      #t1 = 4y
21     add $t2, $t1, $t1      #t2 = 8y
22     add $t3, $t2, $t2      #t3 = 16y
23     add $t4, $t3, $t3      #t4 = 32y
24     add $t5, $t4, $t4      #t5 = 64y
25     add $t6, $t0, $s1      #t6 = 3y
26     add $t7, $t5, $t6      #t7 = 67y
27     add $s2, $s2, $t7      #z = 15x + 67y
28     add $t0, $s2, $s2      #t0 = 2*z
29     add $s2, $t0, $t0      #z = 4*t0
30
31 #FIM
32
33
34
35

```

Line: 35 Column: 5 ☒ Show Line Numbers

| Text Segment | | | | | | | | | |
|--------------------------|------------|------------|--------------------|--------|----------------------|----------------|--|--|--|
| Bkpt | Address | Code | Basic | Source | | | | | |
| <input type="checkbox"/> | 0x00400020 | 0x01084820 | add \$9,\$8,\$8 | 24: | add \$t1, \$t0, \$t0 | #t1 = 4y | | | |
| <input type="checkbox"/> | 0x00400024 | 0x01295020 | add \$10,\$9,\$9 | 25: | add \$t2, \$t1, \$t1 | #t2 = 8y | | | |
| <input type="checkbox"/> | 0x00400028 | 0x014a6020 | add \$11,\$10,\$10 | 26: | add \$t3, \$t2, \$t2 | #t3 = 16y | | | |
| <input type="checkbox"/> | 0x0040002c | 0x016b6020 | add \$12,\$11,\$11 | 27: | add \$t4, \$t3, \$t3 | #t4 = 32y | | | |
| <input type="checkbox"/> | 0x00400030 | 0x018c6020 | add \$13,\$12,\$12 | 28: | add \$t5, \$t4, \$t4 | #t5 = 64y | | | |
| <input type="checkbox"/> | 0x00400034 | 0x01117020 | add \$14,\$9,\$17 | 29: | add \$t5, \$t0, \$s1 | #t5 = 3y | | | |
| <input type="checkbox"/> | 0x00400038 | 0x01ae7820 | add \$15,\$13,\$14 | 30: | add \$t7, \$t5, \$t6 | #t7 = 67y | | | |
| <input type="checkbox"/> | 0x0040003c | 0x024f9020 | add \$18,\$18,\$15 | 31: | add \$s2, \$s2, \$t7 | #z = 15x + 67y | | | |
| <input type="checkbox"/> | 0x00400040 | 0x025a4020 | add \$9,\$18,\$18 | 32: | add \$t0, \$s2, \$s2 | #t0 = 2*z | | | |
| <input type="checkbox"/> | 0x00400044 | 0x01089020 | add \$18,\$8,\$8 | 33: | add \$s2, \$t0, \$t0 | #z = 4*t0 | | | |

| Data Segment | | | | | | | | | |
|--------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|--------|
| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) | |
| 0x10010000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | \$zero |
| 0x10010020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | \$t1 |
| 0x10010040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | \$t0 |
| 0x10010060 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | \$t1 |
| 0x10010080 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | \$t2 |
| 0x100100a0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | \$t3 |
| 0x100100c0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | \$t4 |
| 0x100100e0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | \$t5 |
| 0x10010100 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | \$t6 |
| 0x10010120 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | \$t7 |
| 0x10010140 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | \$t8 |
| 0x10010160 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | \$t9 |
| 0x10010180 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | \$t10 |
| 0x100101a0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | \$t11 |
| 0x100101c0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | \$t12 |

0x10010000 (data) ☒ Hexadecimal Addresses ☒ Hexadecimal Values ☐ ASCII

4)

```

1  #      Programa 4
2  #      x=3;
3  #      y=4;
4  #      z=(15*x + 67*y)*4
5
6  #INICIO
7
8  .text
9  .globl main
10 main:
11     ori $s0, $zero, 3      #x = 3
12     ori $s1, $zero, 4      #y = 4
13     sll $t0, $s0, 4         #t0 = 16x
14     sub $t1, $t0, $s0       #t1 = t0 - x (15x)
15     sll $t2, $s1, 6         #t2 = 64y
16     sll $t3, $s1, 1         #t3 = 2y
17     add $t3, $t3, $s1       #t3 = 3y
18     add $t3, $t3, $t2       #t3 = 67y
19     add $t4, $t1, $t3       #t4 = 15x + 67y
20     sll $s2, $t4, 2         #z = (15x + 67y)*4
21
22 #FIM
23

```

Line: 20 Column: 37 ☒ Show Line Numbers

[illegible]

5)

mipso.asm mips32.asm

```
1  #      Programa 5
2  #      x=100000;
3  #      y=200000;
4  #      z=x+y
5
6  #INICIO
7
8  .text
9  .globl main
10 main:
11     ori $t0, $zero, 0x186A    #t0 = 0x186A
12     sll $s0, $t0, 4           #x = 0x186A0 (100000)
13     ori $t1, $zero, 0x30D4    #t1 = 0x30D4
14     sll $s1, $t1, 4           #y = 0x30D40 (200000)
15     add $s2, $s0, $s1         #z = x+y
16
17 #FIM
18
```

Line: 18 Column: 1 ☒ Show Line Numbers

Text Segment

| Bkpt | Address | Code | Basic | Source |
|--------------------------|------------|------------|--------------------------|---|
| <input type="checkbox"/> | 0x00400000 | 0x3406186a | ori \$t0, \$zero, 0x186A | 11: ori \$t0, \$zero, 0x186A #t0 = 0x186A |
| <input type="checkbox"/> | 0x00400004 | 0x00080100 | sll \$s0, \$t0, 4 | 12: sll \$s0, \$t0, 4 #x = 0x186A0 (100000) |
| <input type="checkbox"/> | 0x00400008 | 0x340930d4 | ori \$t1, \$zero, 0x30D4 | 13: ori \$t1, \$zero, 0x30D4 #t1 = 0x30D4 |
| <input type="checkbox"/> | 0x0040000c | 0x00090900 | sll \$s1, \$t1, 4 | 14: sll \$s1, \$t1, 4 #y = 0x30D40 (200000) |
| <input type="checkbox"/> | 0x00400010 | 0x02119020 | add \$s2, \$s0, \$s1 | 15: add \$s2, \$s0, \$s1 #z = x+y |

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|
| 0x10010000 | 0x00000000 | 0x00000000 | 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 |

0x10010000 (.data) ☒ Hexadecimal Addresses ☒ Hexadecimal Values ☐ ASCII

Labels

| Label | Address |
|----------|------------|
| (global) | |
| main | 0x00400000 |

Registers

| Register | Value |
|----------|-------------|
| \$zero | 0x00000000 |
| \$at | 0x00000000 |
| \$v0 | 0x00000000 |
| \$v1 | 0x00000000 |
| \$a0 | 0x00000000 |
| \$a1 | 0x00000000 |
| \$a2 | 0x00000000 |
| \$a3 | 0x00000000 |
| \$t0 | 0x0000186A |
| \$t1 | 0x000030D4 |
| \$t2 | 0x00000000 |
| \$t3 | 0x00000000 |
| \$t4 | 0x00000000 |
| \$t5 | 0x00000000 |
| \$t6 | 0x00000000 |
| \$t7 | 0x00000000 |
| \$s0 | 0x0000186A0 |
| \$s1 | 0x000030D40 |
| \$s2 | 0x0000495A0 |
| \$s3 | 0x00000000 |
| \$s4 | 0x00000000 |
| \$s5 | 0x00000000 |
| \$s6 | 0x00000000 |
| \$s7 | 0x00000000 |
| \$t8 | 0x00000000 |
| \$t9 | 0x00000000 |
| \$k0 | 0x00000000 |
| \$k1 | 0x00000000 |
| \$gp | 0x10008000 |
| \$sp | 0x7FFFFFFC |
| \$fp | 0x00000000 |
| \$ra | 0x00000000 |
| pc | 0x00400014 |
| hi | 0x00000000 |
| lo | 0x00000000 |

Mars Messages Run I/O

-- program is finished running (dropped off bottom) --

6)

```
mips6.asm
1 # Programa 6
2 # x=maior inteiro possivel (0x7FFFFFFF);
3 # y=300000;
4 # z=x-4y
5
6 #INICIO
7
8 .text
9 .globl main
10 main:
11 ori $t0, $zero, 0x7FFF #t0 = 0x7FFF
12 sll $t0, $t0, 16 #t0 = 0x7FFF0000
13 ori $s0, $t0, 0xFFFF #x = 0x7FFFFFFF
14
15 ori $t1, $zero, 0x493E #t1 = 0x493E
16 sll $s1, $t1, 4 #y = 0x493E0
17 sll $t2, $t1, 2 #t2 = 0x124f80 (4y)
18
19 sub $s2, $s0, $t2 #z = x-t2(x-4y)
20
21 #FIM
22
23
```

[illegible]

```
mips7.asm
1  #      Programa 7
2  #      ori $8, $0, 0x01
3  #      $8 = 0xFFFFFFFF
4
5  #INICIO
6
7  .text
8  .globl main
9  main:
10     ori $8, $0, 0x01      # $8 = 1
11     addi $8, $8, -2       # $8 = 0xFFFFFFFF
12
13  #FIM
14
```

[illegible]

8)

```
mips8.asm
1 # Programa 8
2 # ori $8, $0, 0x12345678
3 # $9 = 0x12
4 # $10 = 0x34
5 # $11 = 0x56
6 # $12 = 0x78
7
8 #INICIO
9
10 .text
11 .globl main
12 main:
13     ori $t0, $0, 0x1234      #t0 = 0x00001234
14     sll $t0, $t0, 16        #t0 = 0x12340000
15     ori $8, $t0, 0x5678     #8 = 0x12345678
16     srl $9, $8, 24          #9 = 0x00000012
17     sll $t5, $8, 8          #t5 = 0x34567800
18     srl $10, $t5, 24        #10 = 0x00000034
19     sll $t6, $8, 16         #t6 = 0x56780000
20     srl $11, $t6, 24        #11 = 0x00000056
21     sll $t7, $8, 24         #t7 = 0x78000000
22     srl $12, $t7, 24        #12 = 0x00000078
23
24 #FIM
25
```

Text Segment

Program Arguments:

| Bkpt | Address | Code | Basic | Source |
|------|------------|------------|--------------------------|--|
| | 0x00400000 | 0x34081234 | ori \$8, \$0, 0x00001234 | 13: ori \$t0, \$0, 0x1234 #t0 = 0x00001234 |
| | 0x00400004 | 0x00084400 | sll \$8, \$0, 0x00000016 | 14: sll \$t0, \$t0, 16 #t0 = 0x12340000 |
| | 0x00400008 | 0x35085678 | ori \$9, \$8, 0x00000018 | 15: ori \$8, \$t0, 0x5678 #8 = 0x12345678 |
| | 0x0040000c | 0x00084e02 | srl \$9, \$8, 0x00000016 | 16: srl \$9, \$8, 24 #9 = 0x00000012 |
| | 0x00400010 | 0x00086a00 | sll \$t5, \$8, 8 | 17: sll \$t5, \$8, 8 #t5 = 0x34567800 |
| | 0x00400014 | 0x00085602 | srl \$t0, \$t5, 24 | 18: srl \$t0, \$t5, 24 #t0 = 0x00000034 |
| | 0x00400018 | 0x00087400 | sll \$t6, \$8, 16 | 19: sll \$t6, \$8, 16 #t6 = 0x56780000 |
| | 0x0040001c | 0x00085e02 | srl \$t1, \$t6, 24 | 20: srl \$t1, \$t6, 24 #t1 = 0x00000056 |
| | 0x00400020 | 0x00087e00 | sll \$t7, \$8, 24 | 21: sll \$t7, \$8, 24 #t7 = 0x78000000 |
| | 0x00400024 | 0x000f6602 | srl \$12, \$t7, 24 | 22: srl \$12, \$t7, 24 #12 = 0x00000078 |

Labels

| Label | Address |
|----------|------------|
| (global) | |
| main | 0x00400000 |

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|
| 0x10010000 | 0x00000000 | 0x00000000 | 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

| 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 | 0x12345678 |
| \$t1 | 9 | 0x00000012 |
| \$t2 | 10 | 0x00000034 |
| \$t3 | 11 | 0x00000056 |
| \$t4 | 12 | 0x00000078 |
| \$t5 | 13 | 0x34567800 |
| \$t6 | 14 | 0x56780000 |
| \$t7 | 15 | 0x78000000 |
| \$s0 | 16 | 0x00000000 |
| \$s1 | 17 | 0x00000000 |
| \$s2 | 18 | 0x00000000 |
| \$s3 | 19 | 0x00000000 |
| \$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 | 0x10000000 |
| \$sp | 29 | 0x7fffffc0 |
| \$fp | 30 | 0x00000000 |
| \$ra | 31 | 0x00000000 |
| pc | | 0x00400028 |
| hi | | 0x00000000 |
| lo | | 0x00000000 |

```
mips9.asm
1 # Programa 9
2 # Escrever um programa que leia todos os números,
3 # calcule e substitua o valor da variável soma por este valor.
4
5 #INICIO
6
7 .data
8 x1: .word 15
9 x2: .word 25
10 x3: .word 13
11 x4: .word 17
12 soma: .word -1
13
14 .text
15 .globl main
16 main:
17     #t0 -> first address
18     #t1 -> offset
19
20     ori $t0, $0, 0x1001    #t0 = 0x1001
21     sll $t0, $t0, 16       #t0 = 0x10010000
22
23     lw $s0, ($t0)          #s0 = x1
24     lw $s1, 4($t0)         #s1 = x2
25     lw $s2, 8($t0)         #s2 = x3
26     lw $s3, 12($t0)        #s3 = x4
27
28     add $s4, $s0, $s1       #s4 = x1 + x2
29     add $s4, $s4, $s2       #s4 = s4 + x3
30     add $s4, $s4, $s3       #s4 = s4 + x4
31
32     sw $s4, 16($t0)         #MEM[10010016] = soma (s4)
33
34 #FIM
```

[illegible]

10)

```

1  #      Programa 10
2  #      Calcule o valor de y conhecendo os valores de x e z, estes armazenados na memória
3  #      Substitua o valor de y pelo seguinte programa y = 127x - 65z + 1.
4
5  #INICIO
6
7  .data
8  x: .word 5
9  z: .word 7
10 y: .word 0      #será sobrescrito após execução
11
12 .text
13 .globl main
14 main:
15     #t0 -> first address
16     #t1 -> offset
17
18     #x -> $s0
19     #z -> $s1
20     #y -> $s2
21
22     ori $t0, $0, 0x1001      #t0 = 0x1001
23     sll $t0, $t0, 16         #t0 = 0x10010000
24
25     lw $s0, 0($t0)          #s0 = x
26     lw $s1, 4($t0)          #s1 = z
27     lw $s2, 8($t0)          #s2 = y
28
29     sll $t1, $s0, 7          #t1 = 128x
30     sub $t1, $t1, $s0        #t1 = 127x
31
32     sll $t2, $s1, 6          #t2 = 64z
33     add $t2, $t2, $s1        #t2 = 65z
34
35     sub $t3, $t1, $t2        #t3 = 127x - 65z
36     addi $s2, $t3, 1         #s2 = 127x - 65z + 1
37
38     sw $s2, 8($t0)          #MEM[10010016] = soma (s2)
39
40 #FIM

```

Line: 26 Column: 24 ☒ Show Line Numbers

Program Arguments:

| Bkpt | Address | Code | Basic | Source |
|--------------------------|------------|------|-------------------|----------------------------|
| <input type="checkbox"/> | 0x00400000 | ori | \$t0, \$0, 0x1001 | #t0 = 0x1001 |
| <input type="checkbox"/> | 0x00400004 | sll | \$t0, \$t0, 16 | #t0 = 0x10010000 |
| <input type="checkbox"/> | 0x00400008 | lw | \$s0, 0(\$t0) | #s0 = x |
| <input type="checkbox"/> | 0x0040000c | lw | \$s1, 4(\$t0) | #s1 = z |
| <input type="checkbox"/> | 0x00400010 | lw | \$s2, 8(\$t0) | #s2 = y |
| <input type="checkbox"/> | 0x00400014 | sll | \$t1, \$s0, 7 | #t1 = 128x |
| <input type="checkbox"/> | 0x00400018 | sub | \$t1, \$t1, \$s0 | #t1 = 127x |
| <input type="checkbox"/> | 0x0040001c | sll | \$t2, \$s1, 6 | #t2 = 64z |
| <input type="checkbox"/> | 0x00400020 | add | \$t2, \$t2, \$s1 | #t2 = 65z |
| <input type="checkbox"/> | 0x00400024 | sub | \$t3, \$t1, \$t2 | #t3 = 127x - 65z |
| <input type="checkbox"/> | 0x00400028 | addi | \$s2, \$t3, 1 | #s2 = 127x - 65z + 1 |
| <input type="checkbox"/> | 0x0040002c | sw | \$s2, 8(\$t0) | #MEM[10010016] = soma (s2) |

Labels

| Label | Address |
|----------|------------|
| (global) | |
| main | 0x00400000 |
| x | 0x10010000 |
| z | 0x10010004 |
| y | 0x10010008 |

Text Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|
| 0x10010000 | 0x00000005 | 0x00000007 | 0x0000000b | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010004 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010008 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001000c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010010 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010014 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010018 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001001c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010024 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010028 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001002c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010030 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010034 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010038 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001003c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|
| 0x10010000 | 0x00000005 | 0x00000007 | 0x0000000b | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010004 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010008 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001000c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010010 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010014 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010018 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001001c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010024 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010028 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001002c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010030 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010034 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010038 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001003c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

Registers

| Name | Number | Value |
|--------|--------|------------|
| \$zero | 0 | 0x00000000 |
| \$at | 1 | 0x00000000 |
| \$v0 | 2 | 0x00000000 |
| \$v1 | 3 | 0x00000000 |
| \$a0 | 4 | 0x00000001 |
| \$a1 | 5 | 0x7ffff1f6 |
| \$a2 | 6 | 0x00000000 |
| \$a3 | 7 | 0x00000000 |
| \$t0 | 8 | 0x10010000 |
| \$t1 | 9 | 0x0000002b |
| \$t2 | 10 | 0x000000c7 |
| \$t3 | 11 | 0x000000b4 |
| \$t4 | 12 | 0x00000000 |
| \$t5 | 13 | 0x00000000 |
| \$t6 | 14 | 0x00000000 |
| \$t7 | 15 | 0x00000000 |
| \$s0 | 16 | 0x00000005 |
| \$s1 | 17 | 0x00000007 |
| \$s2 | 18 | 0x0000000b |
| \$s3 | 19 | 0x00000000 |
| \$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 |
| \$fp | 29 | 0x00000000 |
| \$ra | 31 | 0x00000000 |
| \$pc | | 0x00400038 |
| \$hi | | 0x00000000 |
| \$lo | | 0x00000000 |

11)

```

1  # Programa 11
2  # Calcule o valor de y conhecendo os valores de x e z, estes armazenados na memória
3  # Substitua o valor de y pelo seguinte programa  $y = x - z + 300000$ 
4
5  #INICIO
6
7  .data
8  x: .word 100000
9  z: .word 200000
10 y: .word 0      #será sobrescrito após execução
11
12 .text
13 .globl main
14 main:
15     #t0 -> first address
16     #t1 -> offset
17
18     #x -> $s0
19     #z -> $s1
20     #y -> $s2
21
22     ori $t0, $0, 0x1001      #t0 = 0x1001
23     sll $t0, $t0, 16         #t0 = 0x10010000
24
25     lw $s0, ($t0)      #s0 = x
26     lw $s1, 4($t0)     #s1 = z
27     lw $s2, 8($t0)     #s2 = y
28
29     sub $t1, $s0, $s1   #t1 = x-z
30     ori $t2, $zero 0x493E      #t2 = 0x493E
31     sll $t2, $t2, 4      #t2 = 0x493E0 (300000)
32
33     add $s2, $t1, $t2    #y = x - z + 300000
34
35     sw $s2, 8($t0)      #MEM[8($t0)] = y
36
37 #FIM
38

```

Text Segment

Program Arguments:

| Bkpt | Address | Code | Basic | Source |
|------|------------|------------|-------------------------|--|
| 0 | 0x00400000 | 0x34081001 | ori \$8,\$0,0x0001001 | 22: ori \$t0,\$0,0x1001 #t0 = 0x1001 |
| 1 | 0x00400004 | 0x00084400 | ell \$8,\$8,0x0000010 | 23: ell \$t0,\$t0,16 #t0 = 0x10010000 |
| 2 | 0x00400008 | 0x00100004 | lw \$16,0x0000000(\$8) | 25: lw \$80,\$t0 #s0 = x |
| 3 | 0x0040000c | 0x00110004 | lw \$17,0x0000004(\$8) | 26: lw \$s1,4(\$t0) #s1 = z |
| 4 | 0x00400010 | 0x00120004 | lw \$18,0x0000008(\$8) | 27: lw \$s2,8(\$t0) #s2 = y |
| 5 | 0x00400014 | 0x00214822 | sub \$9,\$16,\$17 | 29: sub \$t1,\$s0,\$s1 #t1 = x-z |
| 6 | 0x00400018 | 0x0040493e | ori \$10,\$0,0x0000493e | 30: ori \$t2,\$zero,0x493e #t2 = 0x493e |
| 7 | 0x0040001c | 0x00045100 | ell \$10,\$10,0x0000004 | 31: ell \$t2,\$t2,4 #t2 = 0x493e0 (300000) |
| 8 | 0x00400020 | 0x012a9028 | add \$18,\$9,\$10 | 33: add \$s2,\$t1,\$t2 #y = x - z + 300000 |
| 9 | 0x00400024 | 0x0d120008 | sw \$18,0x0000008(\$8) | 35: sw \$s2,8(\$t0) #MEM[8(\$t0)] = y |

Labels

| Label | Address |
|------------|------------|
| (global) | |
| main | 0x00400000 |
| mips11.asm | |
| x | 0x10010000 |
| y | 0x10010004 |
| z | 0x10010008 |

☒ Data ☒ Text

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|
| 0x10010000 | 0x00000000 | 0x00000040 | 0x00000040 | 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 |

| Register | Value |
|----------|------------|
| \$zero | 0x00000000 |
| \$at | 0x00000000 |
| \$v0 | 0x00000000 |
| \$v1 | 0x00000000 |
| \$a0 | 0x00000000 |
| \$a1 | 0x00000000 |
| \$a2 | 0x00000000 |
| \$a3 | 0x00000004 |
| \$t0 | 0x10010000 |
| \$t1 | 0x10010004 |
| \$t2 | 0x10010008 |
| \$f16 | 0x00000000 |
| \$f17 | 0x00000000 |
| \$f18 | 0x00000000 |
| \$f19 | 0x00000000 |
| \$f20 | 0x00000000 |
| \$f21 | 0x00000000 |
| \$f22 | 0x00000000 |
| \$f23 | 0x00000000 |
| \$f24 | 0x00000000 |
| \$f25 | 0x00000000 |
| \$f26 | 0x00000000 |
| \$f27 | 0x00000000 |
| \$f28 | 0x00000000 |
| \$f2 | |

```

1 # Programa 12
2 # Considere int ***x;
3 # A primeira posição da memória é do int, coloque os outros valores em reg's e use endereços de memória
4 # O programa deve ler o valor k, multiplicar por 2 e reescrever no local da memória
5
6 #INICIO
7
8 .data
9 k: .word 5
10 a: .word 0x10010000
11 b: .word 0x10010004
12 c: .word 0x10010008
13
14 .text
15 .globl main
16 main:
17     #t0 -> first address
18
19     #k -> $s0
20     #a -> $s1
21     #b -> $s2
22     #c -> $s3
23
24     ori $t0, $0, 0x1001    #t0 = 0x1001
25     sll $t0, $t0, 16       #t0 = 0x10010000
26
27     lw $s0, 12($t0) #s0 = c
28     lw $s1, 0($s0)  #s1 = b
29     lw $s2, 0($s1)  #s2 = a
30     lw $s3, 0($s2)  #s3 = k
31
32     sll $s3, $s3, 1 #s3 = 2*k
33
34     sw $s3, 0($s2)  # ***c = (**c) * 2
35
36 #FIM
37

```

Line: 37 Column: 1 ☒ Show Line Numbers

[illegible]

```
mips13.asm
1 # Programa 13
2 # Escreva um programa que leia um valor A da memória, identifique se é negativo ou não e encontre seu módulo
3 # 0 valor deve ser reescrito sobre A
4
5 #INICIO
6
7 .data
8 A: .word -9
9
10 .text
11 .globl main
12 main:
13     #t0 -> first address
14
15     #A -> $s0
16
17     ori $t0, $0, 0x1001    #t0 = 0x1001
18     sll $t0, $t0, 16       #t0 = 0x10010000
19
20     lw $s0, 0($t0)         #s0 = A
21     sra $t1, $s0, 31       #t1 = s0 >> 31 (mantem o bit de sinal)
22     beq $t1, $0, positivo  #if(t1==0){ goto positivo; }
23     sub $s0, $0, $s0       #s0 = 0 - s0
24 positivo:
25     sw $s0, 0($t0)         #MEM[t0] = s0
26
27
28 #FIM
29
```

Test Segment

Program Arguments:

| Bkpt | Address | Code | Basic | Source |
|--------------------------|------------|----------------|----------------------|--|
| <input type="checkbox"/> | 0x00400000 | 0x3d081001 ori | \$0,\$0,0x00001001 | 17: ori \$10,\$0,0x1001 #10 = 0x1001 |
| <input type="checkbox"/> | 0x00400004 | 0x3d084001 lri | \$0,\$0,0x00001001 | 18: slt \$10,\$0,0x1001 #10 = 0x10010000 |
| <input type="checkbox"/> | 0x00400008 | 0x3d010000 lw | \$16,0x00000000(\$0) | 20: lw \$0,0(\$t0) #0 = A |
| <input type="checkbox"/> | 0x0040000c | 0x00104fc3 sra | \$0,\$16,0x0000001f | 21: sra \$t1,\$0,\$1 #t1 = \$0 >> 31 (santem o bit de sinal) |
| <input type="checkbox"/> | 0x00400010 | 0x11200001 beq | \$0,\$0,0x00000001 | 22: beq \$t1,\$0,positivo #if(t1==0) goto positivo:; |
| <input type="checkbox"/> | 0x00400014 | 0x00100022 sub | \$16,\$0,\$16 | 23: sub \$0,\$0,\$0 #0 = 0 - 0 |
| <input type="checkbox"/> | 0x00400018 | 0x3d010000 sw | \$16,0x00000000(\$0) | 25: sw \$0,0(\$t0) #MEM[t0] = \$0 |

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+C) | Value (+10) | Value (+14) | Value (+18) | Value (+1C) |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|
| 0x10010000 | 0x00000009 | 0x00000000 | 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 |
| 0x100101C0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x100101E0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

Labels

| Label | Address |
|-----------|------------|
| (global) | 0x00400000 |
| main | 0x00400000 |
| mips1.asm | |
| positivo | 0x00400018 |
| A | 0x10010000 |

| 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 |
| \$a4 | 8 | 0x01000000 |
| \$t1 | 9 | 0xffffffff |
| \$t2 | 10 | 0x00000000 |
| \$t3 | 11 | 0x00000000 |
| \$t4 | 12 | 0x00000000 |
| \$t5 | 13 | 0x00000000 |
| \$t6 | 14 | 0x00000000 |
| \$t7 | 15 | 0x00000000 |
| \$s1 | 17 | 0x00000000 |
| \$s2 | 18 | 0x00000000 |
| \$s3 | 19 | 0x00000000 |
| \$s4 | 20 | 0x00000000 |
| \$s5 | 21 | 0x00000000 |
| \$s6 | 22 | 0x00000000 |
| \$s7 | 23 | 0x00000000 |
| \$t8 | 24 | 0x00000000 |
| \$t9 | 25 | 0x00000000 |
| \$0 | 26 | 0x00000000 |
| \$1 | 27 | 0x00000000 |
| \$gp | 28 | 0x10000000 |
| \$fp | 29 | 0x7fffffff |
| \$ra | 30 | 0x00000000 |
| \$ra | 31 | 0x00000000 |
| pc | | 0x00400001 |
| hi | | 0x00000000 |

Mars Messages

Run I/O

```

mips13.asm      mips14.asm
1  #           Programa 14
2  #           Escreva um programa que leia um valor da memoria e identifique se é par ou não
3  #           Devera ser escrito na segunda posição da memoria:
4  #           0 para par ou 1 para impar
5
6  #INICIO
7
8  .data
9  A: .word 9
10
11 .text
12 .globl main
13 main:
14     #t0 -> first address
15
16     #A -> $s0
17     #isEven -> $s1
18
19     ori $t0, $0, 0x1001    #t0 = 0x1001
20     sll $t0, $t0, 16       #t0 = 0x10010000
21     lw $s0, 0($t0)        #s0 = A
22     andi $t1, $s0, 1      #t1 = s0 & 0x00000001
23     bne $t1, $0, else     #if(t1!=0){ goto else; }
24 if:   and $s1, $0, $0     #s1 = 0 & 0
25     j fim                #goto fim
26 else: or $s1, $0, 1      #s1 = 0 | 1
27 fim:  sw $s1, 4($t0)     #MEM[4+t0] = s1
28
29 #FIM
30

```

Line: 30 Column: 1 ☒ Show Line Numbers

Text Segment

Program Arguments:

| Bkpt | Address | Code | Basic | Source |
|------|------------|-----------|----------------------------|------------------------------|
| | 0x00400000 | 0x3481001 | ori \$0, \$0, 0x00000001 | 19: ori \$t0, \$0, 0x1001 |
| | 0x00400004 | 0x0084400 | ll \$t8, 0x00000010 | 20: slt \$t0, \$t0, \$t6 |
| | 0x00400008 | 0x8d10000 | lw \$t6, 0x00000020(\$t8) | 21: lw \$t0, 0(\$t0) |
| | 0x0040000c | 0x3209001 | and \$t9, \$t6, 0x00000012 | 22: and \$t1, \$t0, \$1 |
| | 0x00400010 | 0x1520002 | lne \$t9, \$0, 0x00000002 | 23: bne \$t1, \$0, else |
| | 0x00400014 | 0x0000824 | and \$t7, \$0, \$0 | 24: if: and \$t1, \$t0, \$0 |
| | 0x00400018 | 0x0810008 | 0x00400020 | 25: j file |
| | 0x0040001c | 0x2410001 | ori \$t7, \$0, 0x00000001 | 26: else: or \$t1, \$t0, \$1 |
| | 0x00400020 | 0x8d10004 | lw \$t7, 0x0000004(\$t8) | 27: fin: lw \$t1, 4(\$t0) |

Labels

| Label | Address |
|------------|------------|
| (global) | 0x00400000 |
| main | 0x00400000 |
| mips14.asm | 0x00400014 |
| if | 0x0040001c |
| else | 0x00400020 |
| fin | 0x00100000 |

| Name | Number | Value |
|--------|--------|------------|
| \$zero | 0 | 0x00000000 |
| \$t0 | 1 | 0x00000000 |
| \$t1 | 2 | 0x00000000 |
| \$t2 | 3 | 0x00000000 |
| \$t3 | 4 | 0x00000000 |
| \$t4 | 5 | 0x00000000 |
| \$t5 | 6 | 0x00000000 |
| \$t6 | 7 | 0x00000000 |
| \$t7 | 8 | 0x18000000 |
| \$t8 | 9 | 0x00000000 |
| \$t9 | 10 | 0x00000000 |
| \$t10 | 11 | 0x00000000 |
| \$t11 | 12 | 0x00000000 |
| \$t12 | 13 | 0x00000000 |
| \$t13 | 14 | 0x00000000 |
| \$t14 | 15 | 0x00000000 |
| \$t15 | 16 | 0x00000000 |
| \$t16 | 17 | 0x00000000 |
| \$t17 | 18 | 0x00000000 |
| \$t18 | 19 | 0x00000000 |
| \$t19 | 20 | 0x00000000 |
| \$t20 | 21 | 0x00000000 |
| \$t21 | 22 | 0x00000000 |
| \$t22 | 23 | 0x00000000 |
| \$t23 | 24 | 0x00000000 |
| \$t24 | 25 | 0x00000000 |
| \$t25 | 26 | 0x00000000 |
| \$t26 | 27 | 0x00000000 |
| \$t27 | 28 | 0x00000000 |
| \$t28 | 29 | 0x00000000 |
| \$t29 | 30 | 0x00000000 |
| \$t30 | 31 | 0x00000000 |
| \$t31 | 32 | 0x00000000 |
| \$t32 | 33 | 0x00000000 |
| \$t33 | 34 | 0x00000000 |
| \$t34 | 35 | 0x00000000 |
| \$t35 | 36 | 0x00000000 |
| \$t36 | 37 | 0x00000000 |
| \$t37 | 38 | 0x00000000 |
| \$t38 | 39 | 0x00000000 |
| \$t39 | 40 | 0x00000000 |
| \$t40 | 41 | 0x00000000 |
| \$t41 | 42 | 0x00000000 |
| \$t42 | 43 | 0x00000000 |
| \$t43 | 44 | 0x00000000 |
| \$t44 | 45 | 0x00000000 |
| \$t45 | 46 | 0x00000000 |
| \$t46 | 47 | 0x00000000 |
| \$t47 | 48 | 0x00000000 |
| \$t48 | 49 | 0x00000000 |
| \$t49 | 50 | 0x00000000 |
| \$t50 | 51 | 0x00000000 |
| \$t51 | 52 | 0x00000000 |
| \$t52 | 53 | 0x00000000 |
| \$t53 | 54 | 0x00000000 |
| \$t54 | 55 | 0x00000000 |
| \$t55 | 56 | 0x00000000 |
| \$t56 | 57 | 0x00000000 |
| \$t57 | 58 | 0x00000000 |
| \$t58 | 59 | 0x00000000 |
| \$t59 | 60 | 0x00000000 |
| \$t60 | 61 | 0x00000000 |
| \$t61 | 62 | 0x00000000 |
| \$t62 | 63 | 0x00000000 |
| \$t63 | 64 | 0x00000000 |
| \$t64 | 65 | 0x00000000 |
| \$t65 | 66 | 0x00000000 |
| \$t66 | 67 | 0x00000000 |
| \$t67 | 68 | 0x00000000 |
| \$t68 | 69 | 0x00000000 |
| \$t69 | 70 | 0x00000000 |
| \$t70 | 71 | 0x00000000 |
| \$t71 | 72 | 0x00000000 |
| \$t72 | 73 | 0x00000000 |
| \$t73 | 74 | 0x00000000 |
| \$t74 | 75 | 0x00000000 |
| \$t75 | 76 | 0x00000000 |
| \$t76 | 77 | 0x00000000 |
| \$t77 | 78 | 0x00000000 |
| \$t78 | 79 | 0x00000000 |
| \$t79 | 80 | 0x00000000 |
| \$t80 | 81 | 0x00000000 |
| \$t81 | 82 | 0x00000000 |
| \$t82 | 83 | 0x00000 |

```

1 # Programa 15
2 # Crie um vetor de 100 elementos, onde v[i] = 2*i+1
3 # Apos a ultima posicao, escrever a soma de todos os valores armazenados no vetor
4
5 #INICIO
6
7 .data
8 .text
9 .globl main
10 main:
11     #t0 -> first address
12     #t1 -> offset
13
14     #v[i] -> $t4
15     #soma -> $s1
16     #i -> $t2 (0)
17     #tam -> $t3 (100)
18
19     ori $t0, $0, 0x1001    #t0 = 0x1001
20     sll $t0, $t0, 16       #t0 = 0x10010000
21     or $t1, $t0, $0        #t1 = t0
22     or $t2, $0, $0         #i = 0
23     ori $t3, $0, 100       #tam = 100
24
25 if:
26     beq $t2, $t3, end      #if(t2==t3){ goto end; }
27     add $t4, $t2, $t2      #t4 = 2*i
28     addi $t4, $t4, 1       #t4 = 2*i + 1
29     add $s0, $s0, $t4      #soma = soma + t4
30     sw $t4, 0($t1)         #MEM[$t1] = t4
31     addi $t1, $t1, 4       #t1 = t1 + 4
32     addi $t2, $t2, 1       #i++
33     j if                   #goto if
34 end:
35     sw $s0, 0($t1)         #MEM[$t1] = s0
36
37 #FIM

```

Line: 35 Column: 32 ☒ Show Line Numbers

Mars Messages

Run I/O

Text Segment

Program Arguments:

| Bkpt | Address | Code | Basic | Source |
|--------------------------|------------|------|-------------------------------------|--|
| <input type="checkbox"/> | 0x00400000 | ori | 0x34081001,ori \$8,\$0,0x00001001 | 19: ori \$t0,\$0,0x1001 #t0 = 0x1001 |
| <input type="checkbox"/> | 0x00400004 | ori | 0x00084400,sll \$8,\$8,0x00000100 | 20: sll \$t0,\$t0,16 #t0 = 0x10010000 |
| <input type="checkbox"/> | 0x00400008 | ori | 0x01004825,ori \$9,\$8,\$0 | 21: or \$t1,\$t0,\$0 #t1 = t0 |
| <input type="checkbox"/> | 0x0040000c | ori | 0x000552c0,ori \$10,\$10,\$0 | 22: or \$t2,\$t0,\$0 #t2 = 0 |
| <input type="checkbox"/> | 0x00400010 | ori | 0x340c0064,ori \$11,\$10,0x00000064 | 23: ori \$t3,\$10,100 #t3 = 100 |
| <input type="checkbox"/> | 0x00400014 | beq | 0x114c0007,beq \$10,\$11,0x00000007 | 26: beq \$t2,\$t3,end #if(t2==t3){ goto end; } |
| <input type="checkbox"/> | 0x00400018 | ori | 0x0144c020,add \$12,\$10,\$10 | 27: add \$t4,\$t2,\$t2 #t4 = 2*1 |
| <input type="checkbox"/> | 0x0040001c | ori | 0x218c0001,add \$12,\$12,0x0000... | 28: add \$t4,\$t4,1 #t4 = 2*1 + 1 |
| <input type="checkbox"/> | 0x00400020 | ori | 0x020c8020,add \$10,\$16,\$12 | 29: add \$t0,\$t0,\$t4 #t0 = t0 + t4 |
| <input type="checkbox"/> | 0x00400024 | ori | 0xa2c00000,sw \$12,0x00000000(\$9) | 30: sw \$t4,0(\$t1) #MEM[t1] = t4 |
| <input type="checkbox"/> | 0x00400028 | ori | 0x21390004,add \$9,\$9,0x00000004 | 31: add \$t1,\$t1,4 #t1 = t1 + 4 |
| <input type="checkbox"/> | 0x0040002c | ori | 0x214c0001,add \$10,\$10,0x0000... | 32: add \$t2,\$t2,1 #t2++ |
| <input type="checkbox"/> | 0x00400030 | ori | 0x08100005,ori \$0,0x400014 | 33: j lf #goto lf |
| <input type="checkbox"/> | 0x00400034 | ori | 0xad300000,sw \$16,0x00000000(\$9) | 35: sw \$t0,0(\$t1) #MEM[t1] = t0 |

Labels

| Label | Address |
|----------|------------|
| (global) | 0x00400000 |
| main | 0x00400014 |
| end | 0x00400034 |

☒ Data ☒ Text

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|
| 0x10100020 | 0x00000011 | 0x00000013 | 0x00000015 | 0x00000017 | 0x00000019 | 0x0000001b | 0x0000001d | 0x0000001f |
| 0x10100040 | 0x00000021 | 0x00000023 | 0x00000025 | 0x00000027 | 0x00000029 | 0x0000002b | 0x0000002d | 0x0000002f |
| 0x10100060 | 0x00000031 | 0x00000033 | 0x00000035 | 0x00000037 | 0x00000039 | 0x0000003b | 0x0000003d | 0x0000003f |
| 0x10100080 | 0x00000041 | 0x00000043 | 0x00000045 | 0x00000047 | 0x00000049 | 0x0000004b | 0x0000004d | 0x0000004f |
| 0x101000a0 | 0x00000051 | 0x00000053 | 0x00000055 | 0x00000057 | 0x00000059 | 0x0000005b | 0x0000005d | 0x0000005f |
| 0x101000c0 | 0x00000061 | 0x00000063 | 0x00000065 | 0x00000067 | 0x00000069 | 0x0000006b | 0x0000006d | 0x0000006f |
| 0x101000e0 | 0x00000071 | 0x00000073 | 0x00000075 | 0x00000077 | 0x00000079 | 0x0000007b | 0x0000007d | 0x0000007f |
| 0x10101000 | 0x00000081 | 0x00000083 | 0x00000085 | 0x00000087 | 0x00000089 | 0x0000008b | 0x0000008d | 0x0000008f |
| 0x10101020 | 0x00000091 | 0x00000093 | 0x00000095 | 0x00000097 | 0x00000099 | 0x0000009b | 0x0000009d | 0x0000009f |
| 0x10101040 | 0x000000a1 | 0x000000a3 | 0x000000a5 | 0x000000a7 | 0x000000a9 | 0x000000ab | 0x000000ad | 0x000000af |
| 0x10101060 | 0x000000b1 | 0x000000b3 | 0x000000b5 | 0x000000b7 | 0x000000b9 | 0x000000bb | 0x000000bd | 0x000000bf |
| 0x10101080 | 0x000000c1 | 0x000000c3 | 0x000000c5 | 0x000000c7 | 0x000000c9 | 0x000000cb | 0x000000cd | 0x000000cf |
| 0x101010a0 | 0x000000d1 | 0x000000d3 | 0x000000d5 | 0x000000d7 | 0x000000d9 | 0x000000db | 0x000000dd | 0x000000df |
| 0x101010c0 | 0x000000e1 | 0x000000e3 | 0x000000e5 | 0x000000e7 | 0x000000e9 | 0x000000eb | 0x000000ed | 0x000000ef |
| 0x101010e0 | 0x000000f1 | 0x000000f3 | 0x000000f5 | 0x000000f7 | 0x000000f9 | 0x000000fb | 0x000000fd | 0x000000ff |

☒ Hexadecimal Addresses ☒ Hexadecimal Values ☐ ASCII

16)

```

1  # Programa 16
2  # Avalie a expressao (x*y)/z
3  # Use x=1600000(0x186A00), y=80000(0x13880) e z = 400000(0x61A80), inicialize nos regs
4
5  #INICIO
6
7  .data
8  .text
9  .globl main
10 main:
11     #x -> s0
12     #y -> s1
13     #z -> s2
14
15     ori $s0, $0, 0x186A    #s0 = 0x186A
16     ori $s1, $0, 0x1388    #s1 = 0x1388
17     ori $s2, $0, 0x61A8    #s2 = 0x61A8
18
19     mult $s0,$s1           #s0 * s1
20     mflo $t0               #t0 = s0 * s1
21     div $s3,$t0, $s2       #s3 = t0 / s2
22     sll $s3, $s3, 8        #s3 << 8
23
24 #FIM
25

```

Text Segment

Program Arguments:

| Bkpt | Address | Code | Basic | Source |
|------------|------------|--------------------------|-------|------------------------------------|
| 0x00400000 | 0x34101869 | ori \$16,\$0,0x0000186a | 15: | ori \$s0, \$0, 0x186A #s0 = 0x186A |
| 0x00400004 | 0x34111388 | ori \$17,\$0,0x00001388 | 16: | ori \$s1, \$0, 0x1388 #s1 = 0x1388 |
| 0x00400008 | 0x341261a8 | ori \$18,\$0,0x000061a8 | 17: | ori \$s2, \$0, 0x61A8 #s2 = 0x61A8 |
| 0x0040000c | 0x02110018 | mult \$16,\$17 | 19: | mult \$s0,\$s1 #s0 * s1 |
| 0x00400010 | 0x00004012 | mflo \$8 | 20: | mflo \$t0 #t0 = s0 * s1 |
| 0x00400014 | 0x15400001 | bne \$18,\$0,0x00000001 | 21: | div \$s3,\$t0, \$s2 #s3 = t0 / s2 |
| 0x00400018 | 0x0000000d | break | | |
| 0x0040001c | 0x0112001a | div \$8,\$18 | | |
| 0x00400020 | 0x00009812 | mflo \$19 | | |
| 0x00400024 | 0x00139a00 | sll \$19,\$19,0x00000008 | 22: | sll \$s3, \$s3, 8 #s3 << 8 |

Labels

(global)

main 0x00400000

☒ Data ☒ Text

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|-------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|
| 0x1001.0000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001.0020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001.0040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001.0060 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001.0080 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001.00a0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001.00c0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001.00e0 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001.0100 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001.0120 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001.0140 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001.0160 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

| Label | Address | 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 | 0x01dcdf50 |
| \$t1 | 9 | 0x00000000 |
| \$t2 | 10 | 0x00000000 |
| \$t3 | 11 | 0x00000000 |
| \$t4 | 12 | 0x00000000 |
| \$t5 | 13 | 0x00000000 |
| \$t6 | 14 | 0x00000000 |
| \$t7 | 15 | 0x00000000 |
| \$s0 | 16 | 0x0000186a |
| \$s1 | 17 | 0x00001388 |
| \$s2 | 18 | 0x000061a8 |
| \$s3 | 19 | 0x00004e20 |
| \$s4 | 20 | 0x00000000 |
| \$s5 | 21 | 0x00000000 |
| \$s6 | 22 | 0x00000000 |
| \$s7 | 23 | 0x00000000 |
| \$t8 | 24 | 0x00000000 |
| \$t9 | 25 | 0x00000000 |
| \$t0 | 26 | 0x00000000 |
| \$t1 | 27 | 0x00000000 |
| \$sp | 28 | 0x10008000 |
| \$fp | 29 | 0x7ffffffc |
| \$ra | 30 | 0x00000000 |
| \$pc | 31 | 0x00000026 |
| hi | | 0x00000000 |
| lo | | 0x000000e2 |

17)

```

mips17.asm
1  #          Programa 17
2  #          k = x * y
3  #          x sera lido da primeira posicao, o y da segunda, o k sera resscrito na terceira
4
5  #INICIO
6
7  .data
8  x: .word 9
9  y: .word 5
10 .text
11 .globl main
12 main:
13     #t0 -> 0x10010000 (first position)
14     #x -> s0
15     #y -> s1
16     #k -> s2
17
18     ori $t0, $0, 0x1001    #t0 = 0x1001
19     sll $t0, $t0, 16       #t0 = 0x10010000
20
21     lw $s0, 0($t0)         #s0 = MEM[$t0]
22     lw $s1, 4($t0)         #s1 = MEM[$t0+4]
23
24     and $t1, $0, $0        #t1 = 0
25 if:    beq $t1, $s1, fim    #if(t1==s1){ goto fim }
26     add $s2, $s2, $s0      #t2 = t2 + s0
27     addi $t1, $t1, 1       #t1 = t1++
28     j if                  #goto if
29 fim:
30     sw $s2, 8($t0)         #MEM[$t0+8] = s2
31
32 #FIM
33

```

Line: 33 Column: 1 ☒ Show Line Numbers

Program Arguments:

| Bkpt | Address | Code | Basic | Source |
|------|------------|------------|-----------------------|-------------------------|
| | 0x00400000 | 0x34081001 | ori \$t0, \$0, 0x1001 | #t0 = 0x1001 |
| | 0x00400004 | 0x00840001 | sll \$t0, \$t0, 16 | #t0 = 0x10010000 |
| | 0x00400008 | 0x8d100000 | lw \$s0, 0(\$t0) | #s0 = MEM[\$t0] |
| | 0x0040000c | 0x8d110004 | lw \$s1, 4(\$t0) | #s1 = MEM[\$t0+4] |
| | 0x00400010 | 0x00004824 | and \$t1, \$0, \$0 | #t1 = 0 |
| | 0x00400014 | 0x11100003 | beq \$t1, \$s1, fim | #if(t1==s1){ goto fim } |
| | 0x00400018 | 0x02509020 | add \$s2, \$s2, \$s0 | #t2 = t2 + s0 |
| | 0x0040001c | 0x21290001 | addi \$t1, \$t1, 1 | #t1 = t1++ |
| | 0x00400020 | 0x08100005 | j if | #goto if |
| | 0x00400024 | 0xad120008 | sw \$s2, 8(\$t0) | #MEM[\$t0+8] = s2 |

Labels

| Label | Address |
|------------|------------|
| (global) | |
| main | 0x00400000 |
| mips17.asm | |
| if | 0x00400014 |
| fin | 0x00400024 |
| x | 0x10010000 |
| y | 0x10010004 |

| Name | Number | Value |
|--------|--------|------------|
| \$zero | 0 | 0x00000000 |
| \$at | 1 | 0x00000000 |
| \$t0 | 2 | 0x00000000 |
| \$t1 | 3 | 0x00000000 |
| \$t2 | 4 | 0x00000000 |
| \$t3 | 5 | 0x00000000 |
| \$t4 | 6 | 0x00000000 |
| \$t5 | 7 | 0x00000000 |
| \$t6 | 8 | 0x00000000 |
| \$t7 | 9 | 0x00000000 |
| \$t8 | 10 | 0x00000000 |
| \$t9 | 11 | 0x00000000 |
| \$t10 | 12 | 0x00000000 |
| \$t11 | 13 | 0x00000000 |
| \$t12 | 14 | 0x00000000 |
| \$t13 | 15 | 0x00000000 |
| \$t14 | 16 | 0x00000000 |
| \$t15 | 17 | 0x00000000 |
| \$t16 | 18 | 0x00000000 |
| \$t17 | 19 | 0x00000000 |
| \$t18 | 20 | 0x00000000 |
| \$t19 | 21 | 0x00000000 |
| \$t20 | 22 | 0x00000000 |
| \$t21 | 23 | 0x00000000 |
| \$t22 | 24 | 0x00000000 |
| \$t23 | 25 | 0x00000000 |
| \$t24 | 26 | 0x00000000 |
| \$t25 | 27 | 0x00000000 |
| \$t26 | 28 | 0x00000000 |
| \$t27 | 29 | 0x00000000 |
| \$t28 | 30 | 0x00000000 |
| \$t29 | 31 | 0x00000000 |
| \$t30 | 32 | 0x00000000 |
| \$t31 | 33 | 0x00000000 |

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|
| 0x10010000 | 0x00000009 | 0x00000005 | 0x00000002 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010004 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010008 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001000c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010010 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010014 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010018 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001001c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010024 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010028 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001002c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010030 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010034 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010038 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001003c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

0x10010000 (.data) ☒ Hexadecimal Addresses ☒ Hexadecimal Values ☐ ASCII

```

1 # Programa 18
2 # k = x^y
3 # x sera lido da primeira posicao, o y da segunda, o k sera resscrito na terceira
4
5 #INICIO
6
7 .data
8 x: .word 2
9 y: .word 3
10 .text
11 .globl main
12 main:
13     #t0 -> 0x10010000 (first position)
14     #x -> s0
15     #y -> s1
16     #k -> s2
17
18     ori $t0, $0, 0x1001    #t0 = 0x1001
19     sll $t0, $t0, 16       #t0 = 0x10010000
20
21     lw $s0, 0($t0)         #s0 = MEM[$t0]
22     lw $s1, 4($t0)         #s1 = MEM[$t0+4]
23
24     ori $s2, $0, 1         #s2 = 1
25     or $t2, $0, $s1        #t2 = y
26
27 while: beq $t2, $0, fim     #if(t2==0){ goto fim }
28         mul $s2, $s2, $s0    #s2 = s2 * s0
29         addi $t2, $t2, -1    #t2 = t2-1
30         j while              #goto while
31 fim:
32     sw $s2, 8($t0)         #MEM[$t0+8] = s2
33
34 #FIM

```

[illegible]

Responda

- 1) C. 64
- 2) B. hi e lo
- 3) A. mult
- 4) C. mflo \$8
- 5) B. 32
- 6) A. lo
- 7) D. div
- 8) B. 0010 0110
- 9) A. Se o inteiro for unsigned, o shift o divide por 2. Se o inteiro for signed, o shift o divide por 2.
- 10) A.
 - ori \$3, \$0, 3
 - mult \$8, \$3
 - mflo \$9
 - addi \$9, \$9, 7

19)

```

1  # Programa 19
2  # Ler dois numeros da memoria, determinar qtd de bits
3  # significantes de cada um, multiplicar ambos,
4  #INICIO
5  .data
6  x: .word 2
7  y: .word 3
8  .text
9  .globl main
10 main:
11     #t0 -> 0x10010000 (first position)
12     #x -> s0
13     #y -> s1
14     #k -> s2
15     ori $t2, $0, 0x1001      #t2 = 0x1001
16     sll $t2, $t2, 16         #t2 = 0x10010000
17     lw $s0, 0($t2)          #s0 = MEM[$t0]
18     lw $s1, 4($t2)          #s1 = MEM[$t0+4]
19     or $t3, $s0, $0         #t3 = x
20     or $t4, $s1, $0         #t4 = y
21     or $t0, $0, $0          #t0 = 0 (contador de x)
22     or $t1, $0, $0          #t1 = 0 (contador de y)
23 if: beq $t3, $0, if2        #if(t3==0){ goto fim1}
24     addi $t0, $t0, 1         #t0 = t0 + 1
25     srl $t3, $t3, 1         #t3 >> 1
26     j if                    #goto if
27 if2: beq $t4, $0, fim       #if(t4==0){ goto fim2 }
28     addi $t1, $t1, 1         #t1 = t1 + 1
29     srl $t4, $t4, 1         #t4 >> 1
30     j if2                   #goto if
31 fim: mult $t0, $t1          #t0 * t1
32     mflo $t5                #t5 = lo
33     slti $t6, $t5, 32       #if(t5<32){ t6=1 } else { t6 = 0 }
34     bne $t6, 1, maior       #if(t6!=1){ goto maior }
35     ori $s2, $0, $t5        #s2 = t5
36     mflo $s2                #s2 = lo
37     j fim                   #goto fim
38 maior: mfhi $s2             #s2 = hi
39     mflo $s3                #s3 = lo
40 fim:
41
42 #FIM

```

Program Arguments:

| Bkpt | Address | Code | Basic | Source |
|------------|------------|---------------------------|-----------------------------|------------------------------------|
| 0x00400024 | 0x21080001 | addi \$9,\$9,0x00000001 | 24: addi \$t0, \$t0, 1 | r0 = t0 + 1 |
| 0x00400028 | 0x00005842 | srl \$11,\$11,0x00000001 | 25: srl \$t3, \$t3, 1 | #t3 >> 1 |
| 0x0040002c | 0x00100008 | j 0x00400020 | 26: j if | #goto if |
| 0x00400030 | 0x11800003 | beq \$12,\$0,0x00000003 | 27: if2: beq \$t4, \$0, fim | #if(t4==0){ goto fim2 } |
| 0x00400034 | 0x21200011 | addi \$9,\$9,0x00000001 | 28: addi \$t1, \$t1, 1 | #t1 = t1 + 1 |
| 0x00400038 | 0x000c6042 | srl \$12,\$12,0x00000001 | 29: srl \$t4, \$t4, 1 | #t4 >> 1 |
| 0x0040003c | 0x0010000c | j 0x00400030 | 30: j if2 | #goto if |
| 0x00400040 | 0x01090018 | mult \$8,\$9 | 31: fim: mult \$t0, \$t1 | #t0 * t1 |
| 0x00400044 | 0x00000021 | mflo \$13 | 32: mflo \$t5 | #t5 = lo |
| 0x00400048 | 0x20a00020 | slti \$14,\$13,0x00000000 | 33: slti \$t6, \$t5, 32 | #if(t5<32){ t6=1 } else { t6 = 0 } |
| 0x0040004c | 0x20010001 | addi \$1,\$0,0x00000001 | 34: bne \$t6, 1, maior | #if(t6!=1){ goto maior } |
| 0x00400050 | 0x14200003 | bne \$1,\$14,0x00000003 | | |
| 0x00400054 | 0x00009025 | or \$18,\$0,\$13 | 35: or \$s2, \$0, \$t5 | #s2 = t5 |
| 0x00400058 | 0x00009012 | mflo \$18 | 36: mflo \$s2 | #s2 = lo |
| 0x0040005c | 0x0010001a | j 0x00400068 | 37: j fim2 | #goto fim |

Labels

| Label | Address |
|------------|------------|
| (global) | 0x00400000 |
| main | 0x00400000 |
| mips19.asm | 0x00400020 |
| if | 0x00400030 |
| fim | 0x00400030 |
| f2 | 0x00400040 |
| maior | 0x00400060 |
| fim2 | 0x00400068 |
| x | 0x10010000 |
| y | 0x10010004 |

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|
| 0x10010000 | 0x00000003 | 0x00000004 | 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 |

Registers

| Register | Value |
|----------|------------|
| \$zero | 0x00000000 |
| \$at | 0x00000001 |
| \$v0 | 0x00000000 |
| \$v1 | 0x00000000 |
| \$a0 | 0x00000000 |
| \$a1 | 0x00000000 |
| \$a2 | 0x00000000 |
| \$a3 | 0x00000000 |
| \$t0 | 0x00000000 |
| \$t1 | 0x00000000 |
| \$t2 | 0x00000000 |
| \$t3 | 0x00000000 |
| \$t4 | 0x00000000 |
| \$t5 | 0x00000000 |
| \$t6 | 0x00000000 |
| \$t7 | 0x00000000 |
| \$t8 | 0x00000000 |
| \$t9 | 0x00000000 |
| \$s0 | 0x00000000 |
| \$s1 | 0x00000000 |
| \$s2 | 0x00000000 |
| \$s3 | 0x00000000 |
| \$s4 | 0x00000000 |
| \$s5 | 0x00000000 |
| \$s6 | 0x00000000 |
| \$s7 | 0x00000000 |
| \$s8 | 0x00000000 |
| \$s9 | 0x00000000 |
| \$k0 | 0x00000000 |
| \$k1 | 0x00000000 |
| \$gp | 0x00000000 |
| \$fp | 0x00000000 |
| \$ra | 0x00000000 |
| pc | 0x00000000 |
| hi | 0x00000000 |
| lo | 0x00000000 |

20)

```

1  # Programa 20
2  # Ler x da primeira pos da memoria, se x par y = x^4 + x^3 - 2x^2
3  # se x impar y = x^5 - x^3 + 1, escrever y na segunda pos da memoria
4  #INICIO
5  .data
6  x: .word 3
7  .text
8  .globl main
9  main:
10     ori $t0, $0, 0x1001      #t0 -> 0x10010000 (first position)
11     sll $t0, $t0, 16        #t0 << 16
12
13     lw $s0, 0($t0)          #s0 = MEM[$t0]
14     andi $t1, $s0, 1        #t1 = s0 & 1
15     mult $s0, $s0           #s0 * s0
16     mflo $t2                #t2 = x^2
17     mult $t2, $s0           #t2 * s0
18     mflo $t3                #t3 = x^3
19     mult $t3, $s0           #t3 * s0
20     mflo $t4                #t4 = x^4
21     mult $t4, $s0           #t5 * s0
22     mflo $t5                #t5 = x^5
23     bne $t1, $0, impar      #if(t1!=0){ goto impar }
24     add $t2, $t2, $t2        #t2 = t2 + t2
25     add $s1, $t4, $t3        #s1 = t4 + t3
26     sub $s1, $s1, $t2        #s1 = s1 - t2
27     j fim                   #goto fim
28 impar: sub $s1, $t5, $t3      #s1 = t5 - t3
29     addi $s1, $s1, 1         #s1 = s1 + 1
30 fim:   sw $s1, 4($t0)        #MEM[4+$t0] = s1
31 #FIM
32

```

| Program Arguments: | | | | | Labels | | | |
|--------------------|------------|------------|-----------------|-----------|----------|---------|--|---------------------|
| Bkpt | Address | Code | Basic | Source | Label | Address | | |
| | 0x00400018 | 0x00000012 | mult \$t0, \$t0 | #t2 = x^2 | (global) | | | \$zero 0 0x00000000 |
| | 0x00400019 | 0x00000012 | mult \$t0, \$t0 | #t2 = x^2 | | | | \$t0 1 0x00000000 |
| | 0x0040001c | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$v0 2 0x00000000 |
| | 0x00400023 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$v1 3 0x00000000 |
| | 0x00400024 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$a0 4 0x00000000 |
| | 0x00400028 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$a1 5 0x00000000 |
| | 0x00400029 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$a2 6 0x00000000 |
| | 0x0040002a | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$a3 7 0x00000000 |
| | 0x0040002b | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t0 8 0x10010000 |
| | 0x0040002c | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t1 9 0x00000001 |
| | 0x0040002d | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t2 10 0x00000009 |
| | 0x0040002e | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t3 11 0x0000001b |
| | 0x0040002f | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t4 12 0x00000051 |
| | 0x00400030 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t5 13 0x000000f3 |
| | 0x00400031 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t6 14 0x00000000 |
| | 0x00400032 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t7 15 0x00000000 |
| | 0x00400033 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t8 16 0x00000003 |
| | 0x00400034 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$s1 17 0x000000d9 |
| | 0x00400035 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$s2 18 0x00000000 |
| | 0x00400036 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$s3 19 0x00000000 |
| | 0x00400037 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$s4 20 0x00000000 |
| | 0x00400038 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$s5 21 0x00000000 |
| | 0x00400039 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$s6 22 0x00000000 |
| | 0x0040003a | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$s7 23 0x00000000 |
| | 0x0040003b | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t8 24 0x00000000 |
| | 0x0040003c | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t9 25 0x00000000 |
| | 0x0040003d | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t0 26 0x00000000 |
| | 0x0040003e | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t1 27 0x00000000 |
| | 0x0040003f | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t2 28 0x10000000 |
| | 0x00400040 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t3 29 0x7ffffeffc |
| | 0x00400041 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t4 30 0x00000000 |
| | 0x00400042 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t5 31 0x00000000 |
| | 0x00400043 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t6 0x00400050 |
| | 0x00400044 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t7 0x00000000 |
| | 0x00400045 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t8 0x00000000 |
| | 0x00400046 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t9 0x00000000 |
| | 0x00400047 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t0 0x00000000 |
| | 0x00400048 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t1 0x00000000 |
| | 0x00400049 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t2 0x00000000 |
| | 0x0040004a | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t3 0x00000000 |
| | 0x0040004b | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t4 0x00000000 |
| | 0x0040004c | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t5 0x00000000 |
| | 0x0040004d | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t6 0x00000000 |
| | 0x0040004e | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t7 0x00000000 |
| | 0x0040004f | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t8 0x00000000 |
| | 0x00400050 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t9 0x00000000 |
| | 0x00400051 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t0 0x00000000 |
| | 0x00400052 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t1 0x00000000 |
| | 0x00400053 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t2 0x00000000 |
| | 0x00400054 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t3 0x00000000 |
| | 0x00400055 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t4 0x00000000 |
| | 0x00400056 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t5 0x00000000 |
| | 0x00400057 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t6 0x00000000 |
| | 0x00400058 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t7 0x00000000 |
| | 0x00400059 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t8 0x00000000 |
| | 0x0040005a | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t9 0x00000000 |
| | 0x0040005b | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t0 0x00000000 |
| | 0x0040005c | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t1 0x00000000 |
| | 0x0040005d | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t2 0x00000000 |
| | 0x0040005e | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t3 0x00000000 |
| | 0x0040005f | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t4 0x00000000 |
| | 0x00400060 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t5 0x00000000 |
| | 0x00400061 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t6 0x00000000 |
| | 0x00400062 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t7 0x00000000 |
| | 0x00400063 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t8 0x00000000 |
| | 0x00400064 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t9 0x00000000 |
| | 0x00400065 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t0 0x00000000 |
| | 0x00400066 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t1 0x00000000 |
| | 0x00400067 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t2 0x00000000 |
| | 0x00400068 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t3 0x00000000 |
| | 0x00400069 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t4 0x00000000 |
| | 0x0040006a | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t5 0x00000000 |
| | 0x0040006b | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t6 0x00000000 |
| | 0x0040006c | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t7 0x00000000 |
| | 0x0040006d | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t8 0x00000000 |
| | 0x0040006e | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t9 0x00000000 |
| | 0x0040006f | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t0 0x00000000 |
| | 0x00400070 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t1 0x00000000 |
| | 0x00400071 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t2 0x00000000 |
| | 0x00400072 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t3 0x00000000 |
| | 0x00400073 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t4 0x00000000 |
| | 0x00400074 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t5 0x00000000 |
| | 0x00400075 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t6 0x00000000 |
| | 0x00400076 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t7 0x00000000 |
| | 0x00400077 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t8 0x00000000 |
| | 0x00400078 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t9 0x00000000 |
| | 0x00400079 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t0 0x00000000 |
| | 0x0040007a | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t1 0x00000000 |
| | 0x0040007b | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t2 0x00000000 |
| | 0x0040007c | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t3 0x00000000 |
| | 0x0040007d | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t4 0x00000000 |
| | 0x0040007e | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t5 0x00000000 |
| | 0x0040007f | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t6 0x00000000 |
| | 0x00400080 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t7 0x00000000 |
| | 0x00400081 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t8 0x00000000 |
| | 0x00400082 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t9 0x00000000 |
| | 0x00400083 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t0 0x00000000 |
| | 0x00400084 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t1 0x00000000 |
| | 0x00400085 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t2 0x00000000 |
| | 0x00400086 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t3 0x00000000 |
| | 0x00400087 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t4 0x00000000 |
| | 0x00400088 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t5 0x00000000 |
| | 0x00400089 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t6 0x00000000 |
| | 0x0040008a | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t7 0x00000000 |
| | 0x0040008b | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t8 0x00000000 |
| | 0x0040008c | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t9 0x00000000 |
| | 0x0040008d | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t0 0x00000000 |
| | 0x0040008e | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t1 0x00000000 |
| | 0x0040008f | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t2 0x00000000 |
| | 0x00400090 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t3 0x00000000 |
| | 0x00400091 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t4 0x00000000 |
| | 0x00400092 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t5 0x00000000 |
| | 0x00400093 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t6 0x00000000 |
| | 0x00400094 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t7 0x00000000 |
| | 0x00400095 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t8 0x00000000 |
| | 0x00400096 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t9 0x00000000 |
| | 0x00400097 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t0 0x00000000 |
| | 0x00400098 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t1 0x00000000 |
| | 0x00400099 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t2 0x00000000 |
| | 0x0040009a | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t3 0x00000000 |
| | 0x0040009b | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t4 0x00000000 |
| | 0x0040009c | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t5 0x00000000 |
| | 0x0040009d | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t6 0x00000000 |
| | 0x0040009e | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t7 0x00000000 |
| | 0x0040009f | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t8 0x00000000 |
| | 0x004000a0 | 0x00000012 | mflo \$t3 | #t3 = x^3 | | | | \$t9 0x00 |

21)

```

1  #      Programa 21
2  #      Ler x da primeira pos da memoria, se x>0 y = x^4 + x^3 - 2x^2
3  #      se x<=0 y = x^5 - x^3 + 1, escrever y na segunda pos da memoria
4  #INICIO
5  .data
6  x: .word -3
7  .text
8  .globl main
9  main:
10     ori $t0, $0, 0x1001      #t0 -> 0x10010000 (first position)
11     sll $t0, $t0, 16        #t0 << 16
12
13     lw $s0, 0($t0)          #s0 = MEM[$t0]
14     mult $s0, $s0           #s0 * s0
15     mflo $t2                #t2 = x^2
16     mult $t2, $s0           #t2 * s0
17     mflo $t3                #t3 = x^3
18     mult $t3, $s0           #t3 * s0
19     mflo $t4                #t4 = x^4
20     sle $t1, $s0, $0        #ifs0 <=0 { t1 = 1 } else { t1 = 0 }
21     bne $t1, $0, maior      #if(t1!=0){ goto impar }
22     addi $s1, $t3, 1        #s1 = t3 + 1
23     j fim                   #goto fim
24 maior: addi $s1, $t4, -1     #s1 = t4 - 1
25 fim:   sw $s1, 4($t0)       #MEM[4+$t0] = s1
26
27 #FIM
28

```

Text Segment

Program Arguments:

| Bkpt | Address | Code | Basic | Source |
|------|------------|------------|------------|----------------------|
| | 0x00400000 | 0x00000000 | 0x00000000 | lw \$s0, 0(\$t0) |
| | 0x00400004 | 0x00000016 | 0x00000016 | mult \$s0, \$s0 |
| | 0x00400008 | 0x00000012 | 0x00000012 | mflo \$t2 |
| | 0x0040000c | 0x00000018 | 0x00000018 | mult \$t2, \$s0 |
| | 0x00400010 | 0x00000012 | 0x00000012 | mflo \$t3 |
| | 0x00400014 | 0x00000018 | 0x00000018 | mult \$t3, \$s0 |
| | 0x00400018 | 0x00000012 | 0x00000012 | mflo \$t4 |
| | 0x0040001c | 0x00000011 | 0x00000011 | sle \$t1, \$s0, \$0 |
| | 0x00400020 | 0x00000011 | 0x00000011 | bne \$t1, \$0, maior |
| | 0x00400024 | 0x00000011 | 0x00000011 | addi \$s1, \$t3, 1 |
| | 0x00400028 | 0x00000000 | 0x00000000 | j fim |
| | 0x0040002c | 0x00000011 | 0x00000011 | addi \$s1, \$t4, -1 |
| | 0x00400030 | 0x00000011 | 0x00000011 | sw \$s1, 4(\$t0) |

Labels

| Label | Address |
|----------|------------|
| (global) | 0x00400000 |
| main | |
| maior | 0x0040003c |
| fin | 0x00400040 |

Data Segment

| Address | Value (+0) | Value (+4) | Value (+8) | Value (+c) | Value (+10) | Value (+14) | Value (+18) | Value (+1c) |
|------------|------------|------------|------------|------------|-------------|-------------|-------------|-------------|
| 0x10010000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010004 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010008 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001000c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010010 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010014 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010018 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001001c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010020 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010024 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010028 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001002c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010030 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010034 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010038 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x1001003c | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| 0x10010040 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |

0x10010000 (.data)

Hexadecimal Addresses

Hexadecimal Values

ASCII

0zero

\$t0

\$v1

\$a0

\$a1

\$a2

\$a3

\$a4

\$t1

\$t2

\$t4

\$t5

\$t6

\$t7

\$a0

\$s1

\$s3

\$s4

\$s5

\$s6

\$t8

\$t9

\$k1

\$gp

\$gp

\$fp

\$ra

pc

hi

lo

0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

0x00000000

0x00000001

0x00000002

0x00000003

0x00000004

0x00000005

0x00000006

0x00000007

0x00000008

0x00000009

0xfffffff5

0xfffffff6

0xfffffff7

0xfffffff8

0xfffffff9

0xffffffa0

0xffffffa1

0xffffffa2

0xffffffa3

0xffffffa4

0xffffffa5

0xffffffa6

0xffffffa7

0xffffffa8

0xffffffa9

0xffffffa0

0xffffffa1

0xffffffa2

0xffffffa3

0xffffffa4

0xffffffa5

0xffffffa6

0xffffffa7

0xffffffa8

0xffffffa9

0xffffffa0

0xffffffa1

0xffffffa2

0xffffffa3

0xffffffa4

0xffffffa5

0xffffffa6

0xffffffa7

0xffffffa8

0xffffffa9