

Exercício Prático 06 - MIPS

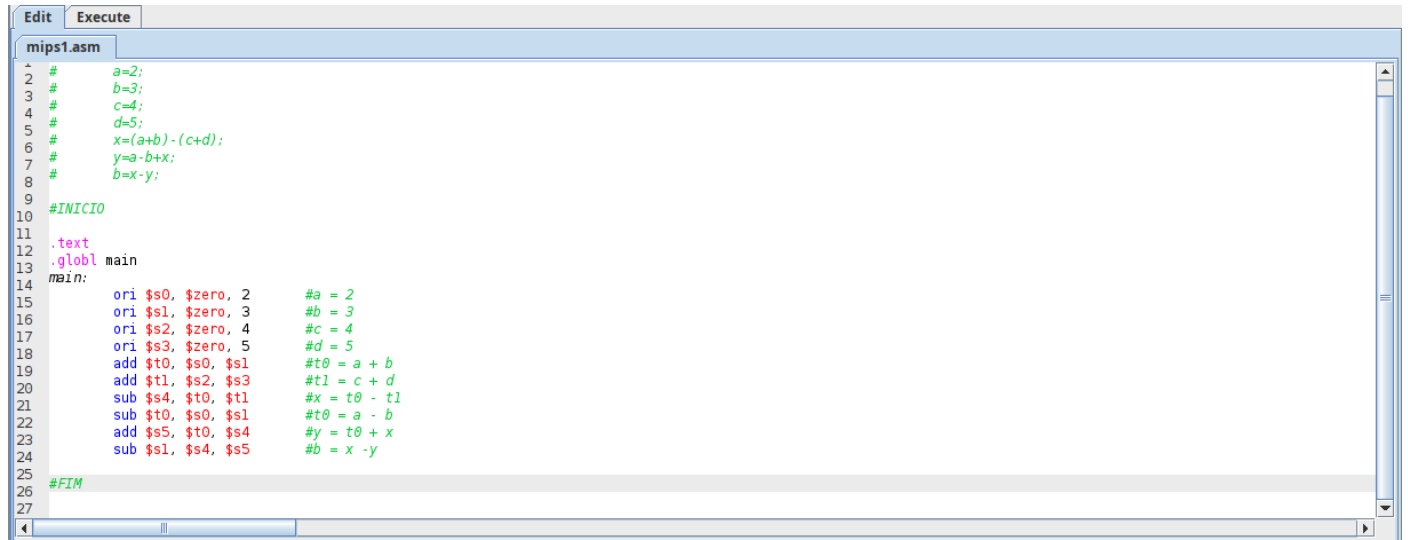
806454 - Yago Almeida Melo

Parte 1 - Questões

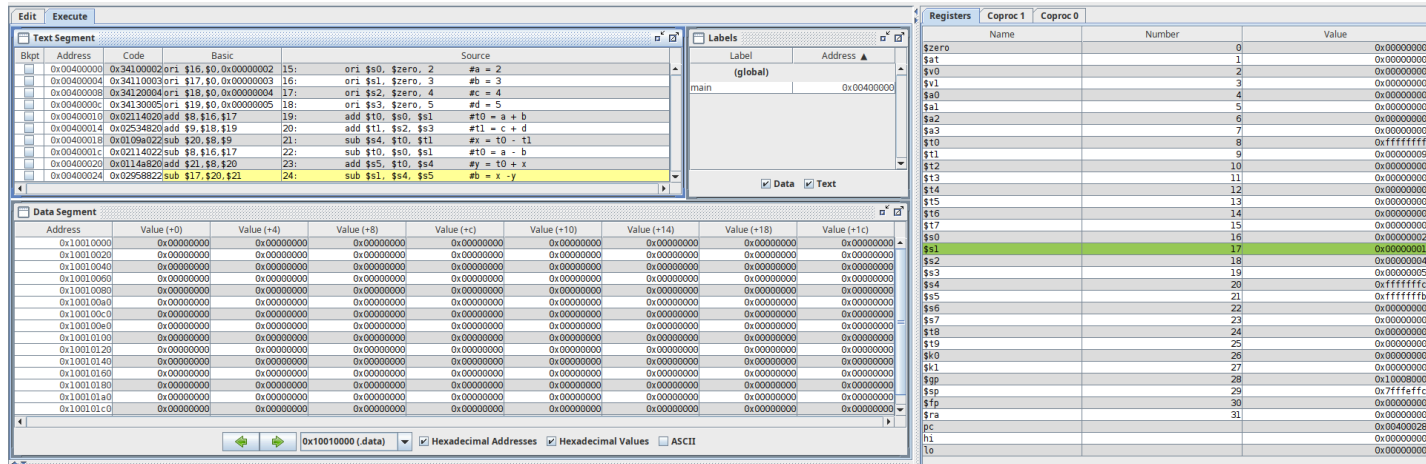
- 1) A. um arquivo de texto que contém instruções de linguagem de programação.
- 2) B. uma parte do processador que possui um padrão de bits.
- 3) A. #
- 4) C. 32
- 5) D. parte do processador que contém o endereço da próxima instrução de máquina para ser obtida.
- 6) C. 4
- 7) D. uma declaração que diz 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** window shows the assembly code with addresses and hex values. The **Registers** window shows the state of the MIPS registers, with \$s0 through \$s5 containing non-zero values.

Blkpt	Address	Code	Basic	Source
	0x00400000	ori \$s0, \$zero, 2	15:	ori \$s0, \$zero, 2 #a = 2
	0x00400004	ori \$s1, \$zero, 3	16:	ori \$s1, \$zero, 3 #b = 3
	0x00400008	ori \$s2, \$zero, 4	17:	ori \$s2, \$zero, 4 #c = 4
	0x0040000c	ori \$s3, \$zero, 5	18:	ori \$s3, \$zero, 5 #d = 5
	0x00400010	add \$t0, \$s0, \$s1	19:	add \$t0, \$s0, \$s1 #t0 = a + b
	0x00400014	add \$t1, \$s2, \$s3	20:	add \$t1, \$s2, \$s3 #t1 = c + d
	0x00400018	sub \$s4, \$t0, \$t1	21:	sub \$s4, \$t0, \$t1 #x = t0 - t1
	0x0040001c	sub \$t0, \$s0, \$s1	22:	sub \$t0, \$s0, \$s1 #t0 = a - b
	0x00400020	add \$s5, \$t0, \$s4	23:	add \$s5, \$t0, \$s4 #y = t0 + x
	0x00400024	sub \$s1, \$s4, \$s5	24:	sub \$s1, \$s4, \$s5 #b = x - y

Register	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	0xffffffff
\$t1		9	0x00000009
\$t2		10	0x00000000
\$t3		11	0x00000000
\$t4		12	0x00000000
\$t5		13	0x00000000
\$t6		14	0x00000000
\$t7		15	0x00000000
\$s0		16	0x00000002
\$s1		17	0x00000001
\$s2		18	0x00000004
\$s3		19	0x00000005
\$s4		20	0xffffffff
\$s5		21	0xffffffff
\$s6		22	0x00000000
\$s7		23	0x00000000
\$t8		24	0x00000000
\$t9		25	0x00000000
\$t10		26	0x00000000
\$t11		27	0x00000000
\$t12		28	0x00000000
\$t13		29	0x7fffffff
\$t14		30	0x00000000
\$t15		31	0x00000000
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	
Offset	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,\$t7,0x0000000f	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)	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

\$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

```

Text segment				LADs				Disasm			
Bit	Address	Code	Basic	Source	Label	Address		Byte	Comment	Offset	Disassembly
<input type="checkbox"/>	0x00400020	0x0108482add \$9,\$8,\$8	24:	add \$t1, \$t0, \$t0	#t1 = 4*			01	0x00000000	0	0x00000000
<input type="checkbox"/>	0x00400024	0x0129582add \$10,\$9,\$9	25:	add \$t2, \$t1, \$t1	#t2 = 9*			01	0x00000000	1	0x00000000
<input type="checkbox"/>	0x00400028	0x0143482add \$t1,\$t0,\$t0	26:	add \$t3, \$t2, \$t2	#t3 = 19*			02	0x00000000	2	0x00000000
<input type="checkbox"/>	0x0040002c	0x010b402add \$t2,\$t1,\$t1	27:	add \$t4, \$t3, \$t3	#t4 = 32*			03	0x00000000	3	0x00000000
<input type="checkbox"/>	0x00400030	0x010c482add \$t3,\$t2,\$t2	28:	add \$t5, \$t4, \$t4	#t5 = 64*			04	0x00000000	4	0x00000000
<input type="checkbox"/>	0x00400034	0x0111702add \$t4,\$t3,\$t3	29:	add \$t6, \$t5, \$t5	#t6 = 3*			05	0x00000000	5	0x00000000
<input type="checkbox"/>	0x00400038	0x01a1782add \$t5,\$t4,\$t4	30:	add \$t7, \$t6, \$t6	#t7 = 67*			06	0x00000027	6	0x00000027
<input type="checkbox"/>	0x0040003c	0x024f902add \$t6,\$t5,\$t5	31:	add \$t2, \$t2, \$t2	#t2 = 15* + 67*			07	0x00000001	7	0x00000001
<input type="checkbox"/>	0x00400040	0x0205402add \$t7,\$t6,\$t6	32:	add \$t2, \$t2, \$t2	#t2 = 2*			08	0x00000020	8	0x00000020
<input type="checkbox"/>	0x00400044	0x0108902add \$t8,\$t7,\$t7	33:	add \$t2, \$t0, \$t0	#t2 = 4*t0			09	0x00000040	9	0x00000040
<input type="checkbox"/>	0x00400048							10	0x00000080	10	0x00000080
<input type="checkbox"/>	0x0040004c							11	0x00000080	11	0x00000080
<input type="checkbox"/>	0x00400050							12	0x00000080	12	0x00000080
<input type="checkbox"/>	0x00400054							13	0x00000000	13	0x00000000
<input type="checkbox"/>	0x00400058							14	0x00000000	14	0x00000000
<input type="checkbox"/>	0x0040005c							15	0x00000000	15	0x00000000
<input type="checkbox"/>	0x00400060							16	0x00000000	16	0x00000000
<input type="checkbox"/>	0x00400064							17	0x00000000	17	0x00000000
<input type="checkbox"/>	0x00400068							18	0x00000000	18	0x00000000
<input type="checkbox"/>	0x0040006c							19	0x00000000	19	0x00000000
<input type="checkbox"/>	0x00400070							20	0x00000000	20	0x00000000
<input type="checkbox"/>	0x00400074							21	0x00000000	21	0x00000000
<input type="checkbox"/>	0x00400078							22	0x00000000	22	0x00000000
<input type="checkbox"/>	0x0040007c							23	0x00000000	23	0x00000000
<input type="checkbox"/>	0x00400080							24	0x00000000	24	0x00000000
<input type="checkbox"/>	0x00400084							25	0x00000000	25	0x00000000
<input type="checkbox"/>	0x00400088							26	0x00000000	26	0x00000000
<input type="checkbox"/>	0x0040008c							27	0x00000000	27	0x00000000
<input type="checkbox"/>	0x00400090							28	0x00000000	28	0x00000000
<input type="checkbox"/>	0x00400094							29	0x00000000	29	0x00000000
<input type="checkbox"/>	0x00400098							30	0x00000000	30	0x00000000
<input type="checkbox"/>	0x0040009c							31	0x00000000	31	0x00000000
<input type="checkbox"/>	0x004000a0							32	0x00000000	32	0x00000000
<input type="checkbox"/>	0x004000a4							33	0x00000000	33	0x00000000
<input type="checkbox"/>	0x004000a8							34	0x00000000	34	0x00000000
<input type="checkbox"/>	0x004000ac							35	0x00000000	35	0x00000000
<input type="checkbox"/>	0x004000b0							36	0x00000000	36	0x00000000
<input type="checkbox"/>	0x004000b4							37	0x00000000	37	0x00000000
<input type="checkbox"/>	0x004000b8							38	0x00000000	38	0x00000000
<input type="checkbox"/>	0x004000bc							39	0x00000000	39	0x00000000
<input type="checkbox"/>	0x004000c0							40	0x00000000	40	0x00000000
<input type="checkbox"/>	0x004000c4							41	0x00000000	41	0x00000000
<input type="checkbox"/>	0x004000c8							42	0x00000000	42	0x00000000
<input type="checkbox"/>	0x004000cc							43	0x00000000	43	0x00000000
<input type="checkbox"/>	0x004000d0							44	0x00000000	44	0x00000000
<input type="checkbox"/>	0x004000d4							45	0x00000000	45	0x00000000
<input type="checkbox"/>	0x004000d8							46	0x00000000	46	0x00000000
<input type="checkbox"/>	0x004000dc							47	0x00000000	47	0x00000000
<input type="checkbox"/>	0x004000e0							48	0x00000000	48	0x00000000
<input type="checkbox"/>	0x004000e4							49	0x00000000	49	0x00000000
<input type="checkbox"/>	0x004000e8							50	0x00000000	50	0x00000000
<input type="checkbox"/>	0x004000ec							51	0x00000000	51	0x00000000
<input type="checkbox"/>	0x004000f0							52	0x00000000	52	0x00000000
<input type="checkbox"/>	0x004000f4							53	0x00000000	53	0x00000000
<input type="checkbox"/>	0x004000f8							54	0x00000000	54	0x00000000
<input type="checkbox"/>	0x004000fc							55	0x00000000	55	0x00000000
<input type="checkbox"/>	0x00400100							56	0x00000000	56	0x00000000
<input type="checkbox"/>	0x00400104							57	0x00000000	57	0x00000000
<input type="checkbox"/>	0x00400108							58	0x00000000	58	0x00000000
<input type="checkbox"/>	0x0040010c							59	0x00000000	59	0x00000000
<input type="checkbox"/>	0x00400110							60	0x00000000	60	0x00000000
<input type="checkbox"/>	0x00400114							61	0x00000000	61	0x00000000
<input type="checkbox"/>	0x00400118							62	0x00000000	62	0x00000000
<input type="checkbox"/>	0x0040011c							63	0x00000000	63	0x00000000
<input type="checkbox"/>	0x00400120							64	0x00000000	64	0x00000000
<input type="checkbox"/>	0x00400124							65	0x00000000	65	0x00000000
<input type="checkbox"/>	0x00400128							66	0x00000000	66	0x00000000
<input type="checkbox"/>	0x0040012c							67	0x00000000	67	0x00000000
<input type="checkbox"/>	0x00400130							68	0x00000000	68	0x00000000
<input type="checkbox"/>	0x00400134							69	0x00000000	69	0x00000000
<input type="checkbox"/>	0x00400138							70	0x00000000	70	0x00000000
<input type="checkbox"/>	0x0040013c							71	0x00000000	71	0x00000000
<input type="checkbox"/>	0x00400140							72	0x00000000	72	0x00000000
<input type="checkbox"/>	0x00400144							73	0x00000000	73	0x00000000
<input type="checkbox"/>	0x00400148							74	0x00000000	74	0x00000000
<input type="checkbox"/>	0x0040014c							75	0x00000000	75	0x00000000
<input type="checkbox"/>	0x00400150							76	0x00000000	76	0x00000000
<input type="checkbox"/>	0x00400154							77	0x00000000	77	0x00000000
<input type="checkbox"/>	0x00400158							78	0x00000000	78	0x00000000
<input type="checkbox"/>	0x0040015c							79	0x00000000	79	0x00000000
<input type="checkbox"/>	0x00400160							80	0x00000000	80	0x00000000
<input type="checkbox"/>	0x00400164							81	0x00000000	81	0x00000000
<input type="checkbox"/>	0x00400168							82	0x00000000	82	0x00000000
<input type="checkbox"/>	0x0040016c							83	0x00000000	83	0x00000000
<input type="checkbox"/>	0x00400170							84	0x00000000	84	0x00000000
<input type="checkbox"/>	0x00400174							85	0x00000000	85	0x00000000
<input type="checkbox"/>	0x00400178							86	0x00000000	86	0x00000000
<input type="checkbox"/>	0x0040017c							87	0x00000000	87	0x00000000
<input type="checkbox"/>	0x00400180							88	0x00000000	88	0x00000000
<input type="checkbox"/>	0x00400184							89	0x00000000	89	0x00000000
<input type="checkbox"/>	0x00400188							90	0x00000000	90	0x00000000
<input type="checkbox"/>	0x0040018c							91	0x00000000	91	0x00000000
<input type="checkbox"/>	0x00400190							92	0x00000000	92	0x00000000
<input type="checkbox"/>	0x00400194							93	0x00000000	93	0x00000000
<input type="checkbox"/>	0x00400198							94	0x00000000	94	0x00000000
<input type="checkbox"/>	0x0040019c							95	0x00000000	95	0x00000000
<input type="checkbox"/>	0x004001a0							96	0x00000000	96	0x00000000
<input type="checkbox"/>	0x004001a4							97	0x00000000	97	0x00000000
<input type="checkbox"/>	0x004001a8							98	0x00000000	98	0x00000000
<input type="checkbox"/>	0x004001ac							99	0x00000000	99	0x00000000
<input type="checkbox"/>	0x004001b0							100	0x00000000	100	0x00000000
<input type="checkbox"/>	0x004001b4							101	0x00000000	101	0x00000000
<input type="checkbox"/>	0x004001b8							102	0x00000000	102	0x00000000
<input type="checkbox"/>	0x004001bc							103	0x00000000	103	0x00000000
<input type="checkbox"/>	0x004001c0							104	0x00000000	104	0x00000000
<input type="checkbox"/>	0x004001c4							105	0x00000000	105	0x00000000
<input type="checkbox"/>	0x004001c8							106	0x00000000	106	0x00000000
<input type="checkbox"/>	0x004001cc							107	0x00000000	107	0x00000000
<input type="checkbox"/>	0x004001d0							108	0x00000000	108	0x00000000
<input type="checkbox"/>	0x004001d4							109	0x00000000	109	0x00000000
<input type="checkbox"/>	0x004001d8							110	0x00000000	110	0x00000000
<input type="checkbox"/>	0x004001dc							111	0x00000000	111	0x00000000
<input type="checkbox"/>	0x004001e0							112	0x00000000	112	0x00000000
<input type="checkbox"/>	0x004001e4							113	0x00000000	113	0x00000000
<input type="checkbox"/>	0x004001e8							114	0x00000000	114	0x00000000
<input type="checkbox"/>	0x004001ec							115	0x00000000	115	0x00000000
<input type="checkbox"/>	0x004001f0							116	0x00000000	11	

4)

```

mips0.asm  mips4.asm
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

Text Segment					Labels		Registers													
Program Arguments:					Label		Address													
Blkpt	Address	Code	Basic	Source	main	(global)		0x00400000	\$zero	0										
	0x00400000	0x34100003	ori \$s0, \$zero, 3	#x = 3					\$s0	1										
	0x00400004	0x34110004	ori \$s1, \$zero, 4	#y = 4					\$s1	2										
	0x00400008	0x00104105	sll \$t0, \$s0, 4	#t0 = 16x					\$s2	3										
	0x0040000c	0x01104822	sub \$t1, \$t0, \$s0	#t1 = t0 - (15x)					\$s3	4										
	0x00400010	0x00115180	sll \$t2, \$s1, 6	#t2 = 64y					\$s4	5										
	0x00400014	0x00115401	sll \$t3, \$s1, 1	#t3 = 2y					\$s5	6										
	0x00400018	0x01715820	add \$t3, \$t3, \$s1	#t3 = 3y					\$s6	7										
	0x0040001c	0x016a5820	add \$t3, \$t3, \$t2	#t3 = 67y					\$s7	8										
	0x00400020	0x01266020	add \$t4, \$t1, \$t3	#t4 = 15x + 67y					\$s8	9										
	0x00400024	0x00c90803	sll \$s2, \$t4, 2	#z = (15x + 67y)*4					\$s9	10										
Data Segment										\$10	11									
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	0x00000000	\$s1	17
0x10010020										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s2	18
0x10010040										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s3	19
0x10010060										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s4	20
0x10010080										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s5	21
0x100100a0										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s6	22
0x100100c0										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s7	23
0x100100e0										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s8	24
0x10010100										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s9	25
0x10010120										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s0	26
0x10010140										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s1	27
0x10010160										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s2	28
0x10010180										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s3	29
0x100101a0										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s4	30
0x100101c0										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s5	31
0x100101e0										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s6	32
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s7	33
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s8	34
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s9	35
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s0	36
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s1	37
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s2	38
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s3	39
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s4	40
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s5	41
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s6	42
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s7	43
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s8	44
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s9	45
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s0	46
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s1	47
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s2	48
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s3	49
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s4	50
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s5	51
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s6	52
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s7	53
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s8	54
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s9	55
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s0	56
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s1	57
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s2	58
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s3	59
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s4	60
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s5	61
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s6	62
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s7	63
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s8	64
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s9	65
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s0	66
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s1	67
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s2	68
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s3	69
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s4	70
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s5	71
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s6	72
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s7	73
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s8	74
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s9	75
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s0	76
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s1	77
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s2	78
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s3	79
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s4	80
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s5	81
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s6	82
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s7	83
										0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	\$s8	84
										0x00000000	0x00000									

```

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

```

[illegible]

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         sll $8, $8, 31
12         sra $8, $8, 31
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        #11 = 0x00000078
23
24 #FIM
25
```

Text Segment

Program Arguments:

Expr	Address	Code	Basic	Source
	0x00400000	0x34001234	ori \$8, \$0, 0x00001234	13: ori \$t0, \$0, 0x1234 #t0 = 0x00001234
	0x00400004	0x00084400	sll \$t0, \$t0, 16	14: sll \$t0, \$t0, 16 #t0 = 0x12340000
	0x00400008	0x35085678	ori \$8, \$8, 0x00005678	15: ori \$8, \$t0, 0x5678 #8 = 0x12345678
	0x0040000c	0x00084e02	srl \$9, \$8, 24	16: srl \$9, \$8, 24 #9 = 0x00000012
	0x00400010	0x00086600	sll \$t5, \$8, 8	17: sll \$t5, \$8, 8 #t5 = 0x34567800
	0x00400014	0x000d5602	srl \$10, \$t5, 24	18: srl \$10, \$t5, 24 #10 = 0x00000034
	0x00400018	0x00087400	sll \$t6, \$8, 16	19: sll \$t6, \$8, 16 #t6 = 0x56780000
	0x0040001c	0x000a5e02	srl \$11, \$t6, 24	20: srl \$11, \$t6, 24 #11 = 0x00000056
	0x00400020	0x00087600	sll \$t7, \$8, 24	21: sll \$t7, \$8, 24 #t7 = 0x78000000
	0x00400024	0x000f5602	srl \$12, \$t7, 24	22: srl \$12, \$t7, 24 #11 = 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
\$sp	28	0x10000000
\$fp	29	0x7fffffc0
\$ra	30	0x00000000
\$pc	31	0x00000000
hi		0x00000000
lo		0x00000000

9)

```

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

```

Line: 34 Column: 5 ☒ Show Line Numbers

Text Segment

Program Arguments:

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x34081001	ori \$8,\$0,0x00001001	20: ori \$t0, \$0, 0x1001 #t0 = 0x1001
<input type="checkbox"/>	0x00400004	0x00844005	sll \$8,\$0,0x00000010	21: sll \$t0, \$t0, 16 #t0 = 0x10010000
<input type="checkbox"/>	0x00400008	0x8d100000	lw \$16,\$0x00000000(\$8)	23: lw \$s0, (\$t0) #s0 = x1
<input type="checkbox"/>	0x0040000c	0x8d110004	lw \$17,\$0x00000004(\$8)	24: lw \$s1, 4(\$t0) #s1 = x2
<input type="checkbox"/>	0x00400010	0x8d120008	lw \$18,\$0x00000008(\$8)	25: lw \$s2, 8(\$t0) #s2 = x3
<input type="checkbox"/>	0x00400014	0x8d13000c	lw \$19,\$0x0000000c(\$8)	26: lw \$s3, 12(\$t0) #s3 = x4
<input type="checkbox"/>	0x00400018	0x0211a020	add \$20,\$16,\$17	28: add \$s4, \$s0, \$s1 #s4 = x1 + x2
<input type="checkbox"/>	0x0040001c	0x0202a020	add \$20,\$20,\$18	29: add \$s4, \$s4, \$s2 #s4 = s4 + x3
<input type="checkbox"/>	0x00400020	0x0203a020	add \$20,\$20,\$19	30: add \$s4, \$s4, \$s3 #s4 = s4 + x4
<input type="checkbox"/>	0x00400024	0xad140010	sw \$20,\$0x00000010(\$8)	32: sw \$s4, 16(\$t0) #MEM[10010016] = soma (s4)

Labels

Label	Address
(global)	
main	0x00400000
mips9.asm	
x1	0x10010000
x2	0x10010004
x3	0x10010008
x4	0x1001000c
soma	0x10010010

☒ Data ☒ Text

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000001
\$a1	5	0x7ffff1ff
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x10010000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x000000ff
\$s1	17	0x00000019
\$s2	18	0x00000004
\$s3	19	0x00000011
\$s4	20	0x00000046
\$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	0x7ffff1ff
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400028
hi		0x00000000
lo		0x00000000

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 #FIM
40

```

Program Arguments:			
Bkpt	Address	Code	Basic
<input type="checkbox"/>	0x04000000	ori \$t0,\$0,0x00000001	22: ori \$t0,\$0,0x00000001
<input type="checkbox"/>	0x04000004	orl \$t0,\$0,0x00000001	23: orl \$t0,\$0,0x00000001
<input type="checkbox"/>	0x04000008	sll \$t0,\$10,16	#t0 = 0x10010000
<input type="checkbox"/>	0x0400000c	lw \$s0,(\$t0)	#s0 = x
<input type="checkbox"/>	0x04000010	sw \$s1,4(\$t0)	\$s1 = z
<input type="checkbox"/>	0x04000014	lw \$s2,8(\$t0)	#s2 = y
<input type="checkbox"/>	0x04000018	sll \$t1,\$s0,7	#t1 = 128z
<input type="checkbox"/>	0x0400001c	add \$t1,\$t1,\$s0	#t1 = 127z
<input type="checkbox"/>	0x04000020	sll \$t2,\$s1,6	#t2 = 64z
<input type="checkbox"/>	0x04000024	add \$t2,\$t2,\$s1	#t2 = 65z
<input type="checkbox"/>	0x04000028	sub \$t3,\$t1,\$t2	#t3 = 127z - 65z
<input type="checkbox"/>	0x0400002c	addi \$s2,\$t3,-1	#s2 = 127z - 65z + 1
<input type="checkbox"/>	0x04000030	sw \$s2,8(\$t0)	#MEM[10010016] = soma(52)

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$al	4	0x00000001
\$a2	5	0x7fffffe0
\$a3	6	0x00000000
\$t0	8	0x10010000
\$t1	9	0x0000027f
\$t2	10	0x00000001
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$d0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10980000
\$fp	29	0x7fffffe0
\$ra	31	0x00000000
pc		0x04000030
n1		0x00000000
t6		0x00000000

Address	Value (+0)	Value (+4)	Value (+8)	Value (+C)	Value (+10)	Value (+14)	Value (+18)	Value (+1C)
0x10010000	0x00000005	0x00000007	0x00000005	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

Legend: ☒ Data ☒ Text

Navigation

```

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 $s0, 0($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

```

[illegible]

```

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 modulo
3  #      O 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
```

Line: 29 Column: 1 ☒ Show Line Numbers

Text Segment				Labels		Name	Number	Value
Program Arguments:								
Bkpt	Address	Code	Basic	Source	Label	Address		
<input type="checkbox"/>	0x00400000	0x34081001	ori \$t0, \$0, 0x00010001	#t0 = 0x1001	(global)	0x00400000	\$zero	0x00000000
<input type="checkbox"/>	0x00400001	0x002840c0	all \$t0, \$t0, 16	#t0 = 0x10010000	main		\$at	0x00000001
<input type="checkbox"/>	0x00400008	0x8d100000	lw \$t0, 0(\$t0)	#t0 = A	mips13.asm		\$v0	0x00000002
<input type="checkbox"/>	0x0040000a	0x00104fc3	sra \$t1, \$t0, 31	#t1 = \$t0 >> 31 (manten o bit de sinal)	positive	0x00400018	\$v1	0x00000003
<input type="checkbox"/>	0x00400010	0x11200001	breq \$t0, \$t0, positive	#if(t0==0) goto positivo;	A	0x10010000	\$a0	0x00000004
<input type="checkbox"/>	0x00400011	0x00108022	sub \$t0, \$t0, \$t0	#t0 = 0 - \$t0			\$a2	0x00000005
<input type="checkbox"/>	0x00400018	0xad100000	sw \$t0, 0(\$t0)	#MEM[t0] = \$t0			\$a3	0x00000006
<input checked="" type="checkbox"/> Data <input checked="" type="checkbox"/> Text								
Data Segment								
Address	Value (-0)	Value (+0)	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
Mars Messages Run I/O								

14)

```

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:  ori $s1, $0, 1       #s1 = 0 | 1
27 fim:   sw $s1, 4($t0)       #MEM[4+t0] = s1
28
29 #FIM
30

```

Text Segment

Program Arguments:

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x34081001	ori \$8,\$0,0x0001001	19: ori \$t0, \$0, 0x1001 #t0 = 0x1001
<input type="checkbox"/>	0x00400004	0x00084400	sll \$8,\$8,0x0000010	20: sll \$t0, \$t0, 16 #t0 = 0x10010000
<input type="checkbox"/>	0x00400008	0x00100000	lw \$16,0x0000000(\$8)	21: lw \$s0, 0(\$t0) #s0 = A
<input type="checkbox"/>	0x0040000c	0x32090001	andi \$9,\$16,0x00000001	22: andi \$t1, \$s0, 1 #t1 = s0 & 0x00000001
<input type="checkbox"/>	0x00400010	0x15200002	bne \$9,\$0,0x00000002	23: bne \$t1, \$0, else #if(t1!=0){ goto else; }
<input type="checkbox"/>	0x00400014	0x00008824	and \$17,\$0,\$0	24: if: and \$s1, \$0, \$0 #s1 = 0 & 0
<input type="checkbox"/>	0x00400018	0x00100008	j \$0	25: j fim #goto fim
<input type="checkbox"/>	0x0040001c	0x34110001	ori \$17,\$0,0x00000001	26: else: ori \$s1, \$0, 1 #s1 = 0 1
<input type="checkbox"/>	0x00400020	0xad110004	sw \$17,0x00000004(\$8)	27: fim: sw \$s1, 4(\$t0) #MEM[4+t0] = s1

Labels

Label	Address
(global)	
main	0x00400000
mips14.asm	

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	0x0040001c
\$a4	8	0x00100000
\$t1	9	0x00000001
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000001
\$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
\$sp	28	0x00000000
\$fp	29	0x77ffeffc
\$ra	30	0x00000000
\$f0	31	0x00000000
\$f1		0x00000000
\$f2		0x00000000
\$f3		0x00000000
\$f4		0x00000000
\$f5		0x00000000
\$f6		0x00000000
\$f7		0x00000000
\$f8		0x00000000
\$f9		0x00000000
\$lo		0x00000000

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000001	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


```

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

[illegible]

16)

```

mips16.asm
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 $t0, $s2           #s3 = t0 / s2
22     mflo $t1
23     sll $s3, $t1, 8        #s3 << 8
24
25 #FIM
26

```

Text Segment					Labels		Registers		PC
Block	Address	Code	Basic	Source	Label	Address	Register	Value	
Program Arguments:					(global)		\$zero	0	0x00000000
	0x00400000	0x3410186a	ori \$16,\$0,0x0000186a	15: ori \$s0, \$0, 0x186A #s0 = 0x186A	main	0x00400000	\$t0	1	0x00000000
	0x00400004	0x34111388	ori \$17,\$0,0x00001388	16: ori \$s1, \$0, 0x1388 #s1 = 0x1388			\$t1	2	0x00000000
	0x00400008	0x341261a8	ori \$18,\$0,0x000061a8	17: ori \$s2, \$0, 0x61A8 #s2 = 0x61A8			\$t2	3	0x00000000
	0x0040000c	0x02110019	mult \$16,\$17	19: mult \$s0,\$s1 #s0 * s1			\$t3	4	0x00000000
	0x00400010	0x00004012	mflo \$8	20: mflo \$t0 #t0 = s0 * s1			\$t4	5	0x00000000
	0x00400014	0x16400001	bne \$18,\$0,0x00000001	21: div \$s3,\$t0, \$s2 #s3 = t0 / s2			\$t5	6	0x00000000
	0x00400018	0x00000000	break				\$t6	7	0x00000000
	0x0040001c	0x0112001e	div \$8,\$18				\$t7	8	0x00000000
	0x00400020	0x00009812	mflo \$19				\$t8	9	0x00000000
	0x00400024	0x00139a00	sll \$19,\$19,0x00000008	22: sll \$s3, \$s3, 8 #s3 << 8			\$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	0x00000000
							\$t83	84	0x00000000
							\$t84	85	0x00000000
							\$t85	86	0x00000000
							\$t86	87	0x00000000
							\$t87	88	0x00000000
							\$t88	89	0x00000000
							\$t89	90	0x00000000
							\$t90	91	0x00000000
							\$t91	92	0x00000000
							\$t92	93	0x00000000
							\$t93	94	0x00000000
							\$t94	95	0x00000000
							\$t95	96	0x00000000
							\$t96	97	0x00000000
							\$t97	98	0x00000000
							\$t98	99	0x00000000
							\$t99	100	0x00000000

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

Text Segment

Program Arguments:

Bkpt	Address	Code	Basic	Source
	0x00400000	0x34081001	ori \$t0, \$0, 0x00001001	18: ori \$t0, \$0, 0x1001 #t0 = 0x1001
	0x00400004	0x00844000	sll \$t0, \$t0, 16	19: sll \$t0, \$t0, 16 #t0 = 0x10010000
	0x00400008	0x8d100000	lw \$s0, 0(\$t0)	21: lw \$s0, 0(\$t0) #s0 = MEM[\$t0]
	0x0040000c	0x8d110004	lw \$s1, 4(\$t0)	22: lw \$s1, 4(\$t0) #s1 = MEM[\$t0+4]
	0x00400010	0x00004824	and \$t1, \$0, \$0	24: and \$t1, \$0, \$0 #t1 = 0
	0x00400014	0x11310003	beq \$t1, \$s1, fim	25: if: beq \$t1, \$s1, fim #if(t1==s1){ goto fim }
	0x00400018	0x02509020	add \$s2, \$s2, \$s0	26: add \$s2, \$s2, \$s0 #t2 = t2 + s0
	0x0040001c	0x21290001	addi \$t1, \$t1, 1	27: addi \$t1, \$t1, 1 #t1 = t1++
	0x00400020	0x08100005	j if	28: j if #goto if
	0x00400024	0xad120008	sw \$s2, 8(\$t0)	30: sw \$s2, 8(\$t0) #MEM[\$t0+8] = s2

Labels

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

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
0x10010044	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010048	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001004c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010050	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010054	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010058	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001005c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010064	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010068	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001006c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010070	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010074	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010078	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001007c	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

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	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$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	0x7ffffcfc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400028
hi		0x00000000
lo		0x00000000

```

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

```

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 3
7  y: .word 4
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     beq $t6, $0, maior           #if(t6!=1){ goto maior }
35     or $s2, $0, $t5              #s2 = t5
36     mflo $s2                     #s2 = lo
37     j fim2                       #goto fim
38 maior: mfhi $s2                  #s2 = hi
39     mflo $s3                     #s3 = lo
40 fim2:
41
42 #FIM

```

Program Arguments

Bkpt	Address	Code	Basic	Source
	0x00000000	0x00000000	0x00000000	
	0x00000001	0x00000000	0x00000000	
	0x00000002	0x00005842	0x00000000	24: srl \$t0, \$t0, 1
	0x00000003	0x00000000	0x00000000	25: srl \$t3, \$t3, 1
	0x00000004	0x00000000	0x00000000	26: j if
	0x00000005	0x11800003	0x00000000	27: if2: beq \$t4, \$0, fin2
	0x00000006	0x21200001	0x00000000	28: addi \$t1, \$t1, 1
	0x00000007	0x00006442	0x00000000	29: srl \$t4, \$t4, 1
	0x00000008	0x00000000	0x00000000	30: j if2
	0x00000009	0x00000000	0x00000000	31: fin: mult \$t0, \$t1
	0x0000000A	0x00000000	0x00000000	32: nflo \$t5
	0x0000000B	0x20000001	0x00000000	33: slti \$t6, \$t5, 32
	0x0000000C	0x00000000	0x00000000	34: bne \$t6, 1, major
	0x0000000D	0x14200003	0x00000000	35: or \$s2, \$0, \$t5
	0x0000000E	0x00000000	0x00000000	36: nflo \$s2
	0x0000000F	0x00000000	0x00000000	37: j fin2

Labels

Label	Address
(global)	0x00400000
main	0x00400000
mips19.asm	0x00400000
if	0x00400000
if2	0x00400000
fin	0x00400000
major	0x00400000
fin2	0x00400000
x	0x10010000
y	0x10010000

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+C)	Value (+10)	Value (+14)	Value (+18)	Value (+1C)
0x10010000	0x00000000	0x00000004	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010002	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010004	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010006	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010008	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001000A	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001000C	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001000E	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010010	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010012	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010014	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010016	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010018	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x1001001A	0x00000000	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

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	0x00000018	mult \$t0, \$t0	#t2 = x^2				\$t0 1 0x00000000
	0x0040001c	0x00000518	mflo \$t3	#t3 = x^3				\$v0 2 0x00000000
	0x00400023	0x00700018	mult \$t0, \$t0	#t3 = x^3				\$v1 3 0x00000000
	0x00400024	0x00000018	mflo \$t3	#t3 = x^3				\$a0 4 0x00000000
	0x00400028	0x00000018	mflo \$t3	#t3 = x^3				\$a1 5 0x00000000
	0x00400029	0x00000018	mflo \$t3	#t3 = x^3				\$a2 6 0x00000000
	0x0040002c	0x00000018	mflo \$t3	#t3 = x^3				\$a3 7 0x00000000
	0x0040002d	0x00000018	mflo \$t3	#t3 = x^3				\$t0 8 0x10010000
	0x0040002e	0x00000018	mflo \$t3	#t3 = x^3				\$t1 9 0x00000001
	0x00400030	0x15200004	bne \$t1, \$0, impar	#if(t1!=0){ goto impar }				\$t2 10 0x00000009
	0x00400034	0x00405020	add \$t2, \$t2, \$t2	#t2 = t2 + t2				\$t3 11 0x0000001b
	0x00400038	0x001b8820	add \$s1, \$t4, \$t3	#s1 = t4 + t3				\$t4 12 0x00000051
	0x0040003c	0x022a8827	sub \$s1, \$s1, \$t2	#s1 = s1 - t2				\$t5 13 0x000000f3
	0x00400040	0x00180013	j fim	#goto fim				\$t6 14 0x00000000
	0x00400044	0x001ab822	sub \$s1, \$t5, \$t3	#s1 = t5 - t3				\$t7 15 0x00000003
	0x00400048	0x22310001	addi \$s1, \$s1, 1	#s1 = s1 + 1				\$s1 16 0x00000000
	0x0040004c	0xad110004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$s2 17 0x00000009
	0x0040004d	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$s3 18 0x00000000
	0x0040004e	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$s4 19 0x00000000
	0x0040004f	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$s5 20 0x00000000
	0x00400050	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$s6 21 0x00000000
	0x00400051	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$s7 22 0x00000000
	0x00400052	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$s8 23 0x00000000
	0x00400053	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t8 24 0x00000000
	0x00400054	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t9 25 0x00000000
	0x00400055	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t0 26 0x00000000
	0x00400056	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t1 27 0x00000000
	0x00400057	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t2 28 0x10000000
	0x00400058	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t3 29 0x7ffffeffc
	0x00400059	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t4 30 0x00000000
	0x0040005a	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t5 31 0x00000000
	0x0040005b	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t6 32 0x00000000
	0x0040005c	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t7 33 0x00000000
	0x0040005d	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t8 34 0x00000000
	0x0040005e	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t9 35 0x00000000
	0x0040005f	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t0 36 0x00000000
	0x00400060	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t1 37 0x00000000
	0x00400061	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t2 38 0x00000000
	0x00400062	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t3 39 0x00000000
	0x00400063	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t4 40 0x00000000
	0x00400064	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t5 41 0x00000000
	0x00400065	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t6 42 0x00000000
	0x00400066	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t7 43 0x00000000
	0x00400067	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t8 44 0x00000000
	0x00400068	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t9 45 0x00000000
	0x00400069	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t0 46 0x00000000
	0x0040006a	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t1 47 0x00000000
	0x0040006b	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t2 48 0x00000000
	0x0040006c	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t3 49 0x00000000
	0x0040006d	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t4 50 0x00000000
	0x0040006e	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t5 51 0x00000000
	0x0040006f	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t6 52 0x00000000
	0x00400070	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t7 53 0x00000000
	0x00400071	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t8 54 0x00000000
	0x00400072	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t9 55 0x00000000
	0x00400073	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t0 56 0x00000000
	0x00400074	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t1 57 0x00000000
	0x00400075	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t2 58 0x00000000
	0x00400076	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t3 59 0x00000000
	0x00400077	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t4 60 0x00000000
	0x00400078	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t5 61 0x00000000
	0x00400079	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t6 62 0x00000000
	0x0040007a	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t7 63 0x00000000
	0x0040007b	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t8 64 0x00000000
	0x0040007c	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t9 65 0x00000000
	0x0040007d	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t0 66 0x00000000
	0x0040007e	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t1 67 0x00000000
	0x0040007f	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t2 68 0x00000000
	0x00400080	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t3 69 0x00000000
	0x00400081	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t4 70 0x00000000
	0x00400082	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t5 71 0x00000000
	0x00400083	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t6 72 0x00000000
	0x00400084	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t7 73 0x00000000
	0x00400085	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t8 74 0x00000000
	0x00400086	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t9 75 0x00000000
	0x00400087	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t0 76 0x00000000
	0x00400088	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t1 77 0x00000000
	0x00400089	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t2 78 0x00000000
	0x0040008a	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t3 79 0x00000000
	0x0040008b	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t4 80 0x00000000
	0x0040008c	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t5 81 0x00000000
	0x0040008d	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t6 82 0x00000000
	0x0040008e	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t7 83 0x00000000
	0x0040008f	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t8 84 0x00000000
	0x00400090	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t9 85 0x00000000
	0x00400091	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t0 86 0x00000000
	0x00400092	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t1 87 0x00000000
	0x00400093	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t2 88 0x00000000
	0x00400094	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t3 89 0x00000000
	0x00400095	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t4 90 0x00000000
	0x00400096	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t5 91 0x00000000
	0x00400097	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t6 92 0x00000000
	0x00400098	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t7 93 0x00000000
	0x00400099	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t8 94 0x00000000
	0x0040009a	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t9 95 0x00000000
	0x0040009b	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t0 96 0x00000000
	0x0040009c	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t1 97 0x00000000
	0x0040009d	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t2 98 0x00000000
	0x0040009e	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t3 99 0x00000000
	0x0040009f	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t4 100 0x00000000
	0x004000a0	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t5 101 0x00000000
	0x004000a1	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t6 102 0x00000000
	0x004000a2	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t7 103 0x00000000
	0x004000a3	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t8 104 0x00000000
	0x004000a4	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t9 105 0x00000000
	0x004000a5	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t0 106 0x00000000
	0x004000a6	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t1 107 0x00000000
	0x004000a7	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t2 108 0x00000000
	0x004000a8	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t3 109 0x00000000
	0x004000a9	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t4 110 0x00000000
	0x004000aa	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t5 111 0x00000000
	0x004000ab	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t6 112 0x00000000
	0x004000ac	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t7 113 0x00000000
	0x004000ad	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t8 114 0x00000000
	0x004000ae	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t9 115 0x00000000
	0x004000af	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t0 116 0x00000000
	0x004000b0	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t1 117 0x00000000
	0x004000b1	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t2 118 0x00000000
	0x004000b2	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s1				\$t3 119 0x00000000
	0x004000b3	0x00000004	sw \$s1, 4(\$t0)	#MEM[4+\$t0] = s				

21)

```

mips21.asm
1  #      Programa 21
2  #      Ler x da primeira pos da memoria, se x>0 y = x3 + 1
3  #      se x<=0 y = x4 - 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     lw $s0, 0($t0)          #s0 = MEM[$t0]
13     mult $s0, $s0            #s0 * s0
14     mflo $t2                 #t2 = x2
15     mult $t2, $s0            #t2 * s0
16     mflo $t3                 #t3 = x3
17     mult $t3, $s0            #t3 * s0
18     mflo $t4                 #t4 = x4
19     slt $t1, $0, $s0         #if(x>0){ t1 = 1 } else { t1 = 0 }
20     beq $t1, $0, menorIgual  #if(t1 == 0){ goto maior }
21     addi $s1, $t3, 1         #s1 = t3 + 1
22     j fim                    #goto fim
23  menorIgual:
24     addi $s1, $t4, -1        #s1 = t4 - 1
25  fim:      sw $s1, 4($t0)     #MEM[4+$t0] = s1
26
27  #FIM

```

Text Segment					Labels					Name			Number			Value		
Blkpt	Address	Code	Basic	Source	Label	Address												
	0x00400000	0x34081001	ori \$8,\$0,0x00001001	10: ori \$t0, \$0, 0x1001 #t0 -> 0x10010000 (first position)						\$zero		0				0x00000000		
	0x00400004	0x00844000	sll \$8,\$8,0x00000010	11: sll \$t0, \$t0, 16 #t0 << 16						\$at		1				0x00000000		
	0x00400008	0x84100000	lw \$16,0x00000000(\$8)	12: lw \$s0, 0(\$t0) #s0 = MEM[\$t0]	main	0x00400000				\$v0		2				0x00000000		
	0x0040000c	0x02100018	mult \$16,\$16	13: mult \$s0, \$s0 #s0 * s0						\$v1		3				0x00000000		
	0x00400010	0x00005012	mflo \$10	14: mflo \$t2 #t2 = x ²						\$a0		4				0x00000000		
	0x00400014	0x01500018	mult \$10,\$16	15: mult \$t2, \$s0 #t2 * s0						\$a1		5				0x00000000		
	0x00400018	0x00005012	mflo \$11	16: mflo \$t3 #t3 = x ³	menorIgual	0x00400014				\$a2		6				0x00000000		
	0x0040001c	0x01700018	mult \$11,\$16	17: mult \$t3, \$s0 #t3 * s0	fim	0x00400038				\$a3		7				0x00000000		
	0x00400020	0x00005012	mflo \$12	18: mflo \$t4 #t4 = x ⁴	x	0x10010000				\$t0		8				0x10010000		
	0x00400024	0x0010462a	slt \$10,\$16	19: slt \$t1, \$0, \$s0 #if(x>0){ t1 = 1 } else { t1 = 0 }						\$t1		9				0x00000000		
	0x00400028	0x11200002	beq \$9,\$0,0x00000002	20: beq \$t1, \$0, menorIgual #if(t1 == 0){ goto maior }						\$t2		10				0x00000000		
	0x0040002c	0x21710001	addi \$17,\$11,0x0000...	21: addi \$s1, \$t3, 1 #s1 = t3 + 1						\$t3		11				0xffffffff5		
	0x00400030	0x0810000e	j 0x00400038	22: j fim #goto fim						\$t4		12				0x00000051		
	0x00400034	0x2191ffff	addi \$17,\$12,0xffff...	24: addi \$s1, \$t4, -1 #s1 = t4 - 1						\$t5		13				0x00000000		
	0x00400038	0x84110004	sw \$17,0x00000004(\$8)	25: sw \$s1, 4(\$t0) #MEM[4+\$t0] = s1						\$t6		14				0x00000000		
										\$t7		15				0x00000000		
										\$s0		16				0xffffffffd		
										\$s1		17				0x00000050		
										\$s2		18				0x00000000		
										\$s3		19				0x00000000		
										\$s4		20				0x00000000		
										\$s5		21				0x00000000		
										\$s6		22				0x00000000		
										\$s7		23				0x00000000		
										\$t8		24				0x00000000		
										\$t9		25				0x00000000		
										\$t0		26				0x00000000		
										\$t1		27				0x00000000		
										\$gp		28				0x10080000		
										\$sp		29				0x7ffffeffc		
										\$fp		30				0x00000000		
										\$ra		31				0x00000000		
										pc						0x0040003c		
										hi						0x00000000		
										lo						0x00000051		