



Aluno: Davi Ventura Cardoso Perdigão

CIU: 82148

1.

a)  $a = b - c$

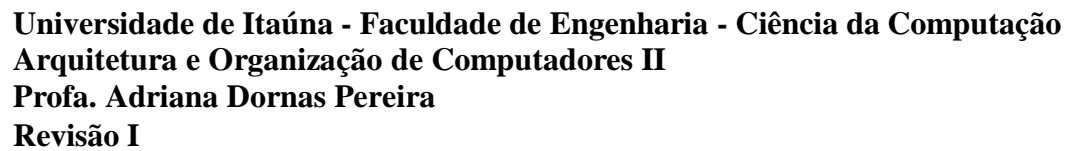
Código Assembly:	sub \$s0, \$s1, \$s2
Linguagem de Máquina:	sub \$16, \$17, \$18
Representação:	0   17   18   16   0   34
Código de máquina:	000000 01001 10010 10000 00000 100010

b)  $b = a + c$

Código Assembly:	add \$s1, \$s0, \$s2
Linguagem de Máquina:	sub \$17, \$16, \$18
Representação:	0   16   18   17   0   32
Código de máquina:	000000 10000 10010 01001 00000 100000

c)  $d = (a + b - c)$

Código Assembly:	add \$t1, \$s0, \$s1
	sub \$s3, \$t1, \$s2
Linguagem de Máquina:	add \$9, \$16, \$17
	sub \$19, \$9, \$18
Representação:	0   16   17   9   0   32
	0   9   18   19   0   34
Código de máquina:	000000 10000 10001 01001 00000 100000
	000000 01001 10010 10011 00000 100010



**d)  $\mathbf{f} = (\mathbf{a} + \mathbf{b}) - \mathbf{d}$**

Código Assembly:

```
add $t1, $s0, $s1  
  
sub $s5, $t1, $s3
```

Linguagem de Máquina:

```
add $9, $16, $17  
sub $21, $9, $19
```

Representação:  $|0|16|17|9|0|32|$   
 $|0|9|19|21|0|34|$

Código de máquina:

	000000	10000	10001	01001	00000	100000
	000000	01001	10011	10101	00000	100010

e)  $\mathbf{c} = \mathbf{a} - (\mathbf{b} + \mathbf{d})$

Código Assembly:                   add \$t1, \$s1, \$s3

  sub \$s2, \$s0, \$t1

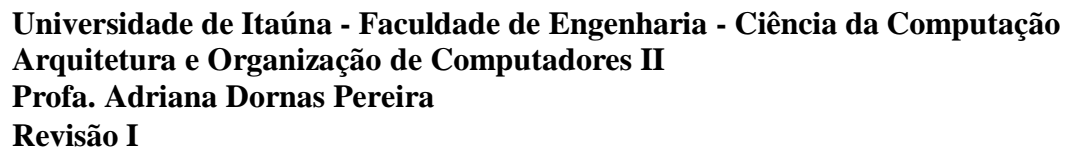
Linguagem de Máquina:

	add \$9, \$17, \$19
	sub \$18, \$16, \$9

Representação:  $|0|17|19|9|0|32|$   
 $|0|16|9|18|0|34|$

Código de máquina:

	000000	10001	10011	01001	00000	100000
	000000	10000	01001	10010	00000	100010



**f)  $\mathbf{e} = (\mathbf{a} - (\mathbf{b} - \mathbf{c}))$**

Código Assembly:                   sub \$t1, \$s1, \$s2

  sub \$s4, \$s0, \$t1

Linguagem de Máquina: sub \$9, \$17, \$18  
sub \$20, \$16, \$9

Representação:  $|0|16|17|9|0|32|$   
 $|0|9|19|21|0|34|$

Código de máquina: 000000 10000 10001 01001 00000 100000  
000000 01001 10011 10101 00000 100010

**g)  $e = (a - (b - c) + f)$**

```
Código Assembly:      sub $t1, $s1, $s2
                      sub $t2, $s0, $t1
                      add $s4, $t2, $s5
```

Linguagem de Máquina:

sub	\$9, \$17, \$18
sub	\$10, \$16, \$9
add	\$20, \$10, \$21

Representação:

0   17   18   9   0   34
0   16   9   10   0   34
0   10   21   20   0   32

Código de máquina:

	000000	10001	10010	01001	00000	100010
	000000	10000	01001	01010	00000	100010
	000000	01010	10101	10100	00000	100000



h)  $f = e - (a - b) + (b - c)$

Código Assembly:

```
sub $t0, $s0, $s1  
sub $t1, $s1, $s2  
sub $t2, $s4, $t0  
add $s5, $t2, $t1
```

Linguagem de Máquina:

```
sub $8, $16, $17  
sub $9, $17, $18  
sub $10, $20, $8  
add $21, $10, $9
```

Representação:

```
| 0 | 16 | 17 | 8 | 0 | 32 |  
| 0 | 17 | 18 | 9 | 0 | 32 |  
| 0 | 20 | 8 | 10 | 0 | 32 |  
| 0 | 10 | 9 | 21 | 0 | 34 |
```

Código de máquina:

```
000000 10000 10001 01000 00000 100000  
000000 10001 10010 01001 00000 100000  
000000 10010 01000 01001 00000 100000  
000000 01001 01001 10101 00000 100010
```



2.

**a)  $a = b[15] - c;$**

Código Assembly:                      lw \$t0, 60(\$s1)    #  $15 * 4 = 60$ , assim:  $\$t0 = \text{mem}[\$s1+60]$   
sub \$s0, \$t0, \$s2

Linguagem de Máquina:              lw \$8, 60(\$17)  
sub \$16, \$8, \$8

Representação:                      | 35 | 8 | 17 | 60 |  
| 0 | 8 | 18 | 16 | 0 | 34 |

Código de máquina:                100011 01000 10001 0000000000111100  
000000 01000 10010 1000 00000 100010

**b)  $b = a[5] + c[3];$**

Código Assembly:                      lw \$t0, 20(\$s0)    #  $5 * 4 = 20$ , assim:  $\$t0 = \text{mem}[\$s0+20]$   
lw \$t1, 12(\$s2)    #  $3 * 4 = 12$ , assim:  $\$t1 = \text{mem}[\$s2+12]$   
add \$s1, \$t0, \$t1

Linguagem de Máquina:              lw \$8, 20(\$16)  
lw \$9, 12(\$18)  
add \$17, \$8, \$9

Representação:                      | 35 | 8 | 16 | 20 |  
| 35 | 9 | 18 | 12 |  
| 0 | 8 | 9 | 17 | 0 | 32 |

Código de máquina:                100011 01000 10000 0000000000010100  
100011 01001 10010 0000000000001100  
000000 01000 01001 10001 00000 100000



c)  $c = b - a[21];$

Código Assembly: `lw $t0, 84($s0) # 21 * 4 = 84, assim: $t0 = mem[$s0+84]`

`sub $s2, $s1, $t0`

Linguagem de Máquina: `lw $8, 84($16)`

`sub $18, $17, $8`

Representação: `| 35 | 8 | 16 | 84 |`

`| 0 | 17 | 8 | 0 | 0 | 34 |`

Código de máquina: `100011 01000 10000 0000000001010100`

`000000 10001 01000 10010 00000 100010`

3.

a)  $a[10] = b - c;$

Código Assembly: `sub $t0, $s1, $s2`

`sw $t0, 40($s0) # 10 * 4 = 40, assim: mem[$s0+40] = $t0`

Linguagem de Máquina: `sub $8, $17, $18`

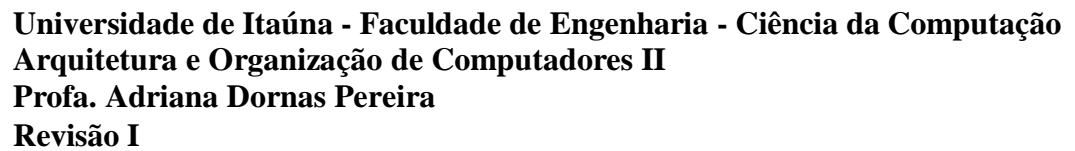
`sw $8, 40($s0)`

Representação: `| 0 | 17 | 18 | 8 | 0 | 34`

`| 43 | 8 | 16 | 40 |`

Código de máquina: `0000000 10001 10000 01000 00000 100010`

`101011 01000 01000 0000000000101000`



Código Assembly:	add \$t0, \$s0, \$s2
	sw \$t0, (\$s1) # 245 * 4 = 980, assim: mem[\$s1+980] = \$t0
Linguagem de Máquina:	add \$8, \$16, \$18
	sw \$8, 980(\$17)
Representação:	0   16   18   8   0   32
	43   8   17   980
Código de máquina:	0000000 10000 10010 01000 00000 100000
	101011 01000 10001 0000001111010100

**c)  $c[0] = b - a$ ;**

Código Assembly:	sub \$t0, \$s1, \$s0  sw \$t0, (\$s2) # 0 * 4 = 0, assim: mem[\$s2+0] = \$t0
Linguagem de Máquina:	sub \$8, \$17, \$16  sw \$8, 0(\$18)
Representação:	0   17   16   8   0   34      43   8   18   0
Código de máquina:	0000000 10001 10000 01000 00000 100010  101011 01000 10010 0000000000000000



4.

a)  $a[34] = b[3] + c - d$

Código Assembly:

lw \$t0, \$12(\$s1)

add \$t1, \$t0, \$s2

sub \$t2, \$t1, \$s4

sw \$t1, 136(\$s0)

Linguagem de Máquina:

lw \$8, \$12(\$17)

add \$9, \$8, \$18

sub \$10, \$9, \$20

sw \$10, 136(\$16)

Representação:

| 35 | 8 | 17 | 12 |

| 0 | 8 | 18 | 9 | 0 | 32 |

| 0 | 9 | 20 | 10 | 0 | 34 |

| 43 | 10 | 16 | 136 |

Código de máquina:

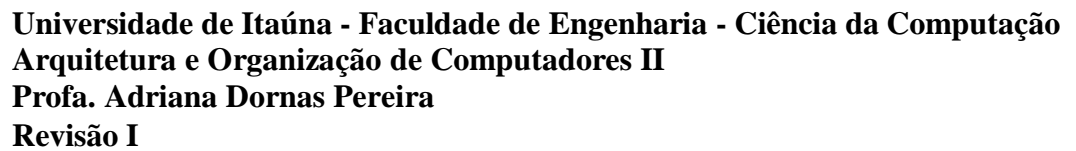
100011 01000 10001 0000000000001100

000000 01000 10010 01001 00000 100000

000000 01001 10100 01010 00000 100010

101011 01010 10000 0000000010001000





```
Código Assembly:      sub $t0, $s1, $s2
                      lw $t1, 268($20)
                      add $t0, $t0, $t1
                      sw $t0, 180($s0)
```

Representação:

0	17	18	8	0	34
35	9	20	268		
0	8	9	8	0	32
43	8	16	180		

Código de máquina:

000000	10001	10010	01000	00000	100010
100011	01001	10100	0000000	1000011	00
000000	01000	01001	01000	00000	100000
101011	01000	10000	00000000	10110100	



c)  $a[79] = b - c[18] + d$

Código Assembly:

```
lw $t0, 72($s2)
sub $t1, $s1,$t0
add $t1, $t1, $s4
sw $t1, 316($s0)
```

Linguagem de Máquina:

```
lw $8, 72($182)
sub $9, $17, $8
add $9, $9, $20
sw $9, 316($16)
```

Representação:

```
| 35 | 8 | 18 | 72 | | |
| 0 | 17 | 8 | 9 | 0 | 34 |
| 0 | 9 | 20 | 9 | 0 | 32 |
| 43 | 9 | 16 | 316 |
```

Código de máquina:

```
100011 01000 10010 000000000001001000
000000 10001 01000 00000 100010
00000 01001 10100 00000 100000
101011 01001 10000 0000000100111100
```



d)  $a[82] = b[2] - c[4]$

Código Assembly:

```
lw $t0, 4($s1)
lw $t1, 15($s2)
sub $t2, $t0, $t1
sw $t2, 328($s0)
```

Linguagem de Máquina:

```
lw $8, 4($17)
lw $9, 16($18)
sub $10, $8, $9
sw $10, 328($16)
```

Representação:

```
| 35 | 8 | 17 | 4 | | |
| 35 | 9 | 18 | 16 |
| 0 | 8 | 9 | 10 | 0 | 34 |
| 43 | 16 | 10 | 328
```

Código de máquina:

```
10001101000 10001 0000000000000100
100011 01001 10010 00000000000010000
000000 01000 01001 01010 00000 100010
101011 10000 01010 0000000101001000
```



5. Complete os espaços com as respectivas palavras:

**Bibliotecas / Memória / Instruções / MIPS / Compiladores / Sistemas Operacionais / Processador / Linguagem de montagem / Assembly / Registradores**

- a) MIPS deve fornecer instruções para transferir dados entre a memória e os **Compiladores**.
- b) As **Instruções** nada mais são que um conjunto de bits inteligíveis pelo computador e que podem ser associados a números.
- c) A **Linguagem de montagem** obriga o programador a escrever uma linha para cada instrução a ser executada pela máquina, forçando-o a raciocinar como a máquina.
- d) **Registradores** são programas que aceitam uma notação mais natural, muito próxima da nossa linguagem.
- e) A Linguagem de montagem ou **Assembly** é uma notação legível por humanos para o código de máquina que uma arquitetura de computador específica utiliza.
- f) As **Bibliotecas** foram criadas e compartilhadas, e suas primeiras versões tinham o objetivo de controlar E/S (impressoras – a presença do papel no início da impressão, controle de discos, fitas, vídeos, etc.)
- g) O **Processador** é justamente a parte ativa da placa-mãe, responsável direto pela execução das instruções de um programa.
- h) A **Memória** é o local onde os programas ficam armazenados enquanto estão sendo processados; lá também ficam guardados os dados necessários à execução dos programas.
- i) **Sistemas Operacionais** são programas que gerenciam os recursos de um computador em benefício dos programas que rodam naquela máquina.
- j) A tradução das instruções em C para a linguagem de montagem do MIPS é realizada por um compilador. O código **MIPS** é produzido pelo compilador C.