

1 - addi T0, zero, 9  
 addi T1, zero, 15  
 sub T0, T1, T0

imediato	rs1	F3	rd	OPCODE	
000000001001	00000	000	00101	0010011	addi
000000001111	00000	000	00110	0010011	addi
F7	rs2	rs1	F3	rd	OPCODE
0100000	00101	00110	000	00101	0110011 sub

2 - Somar 10 inteiros de um vetor, e salvar soma em T6(x31)

addi x31, x0, 0	O programa começa na posição de memória 28 (0x1C), então a partir da posição 32 (0x20) é colocada uma sequência de 10 inteiros.
addi x5, x0, 32	
lw x6, 0(x5)	
add x31, x31, x6	
addi x5, x5, 4	
lw x6, 0(x5)	
bne x6, x0, -12	
jal x0, 0	

(3-4) - binário para decimal e hexadecimal

a) 010000000

decimal  $\rightarrow$  Tem 0-8 casas, o 1 fica na casa 7, então:  
 $(1 \times 2^7) = 128$ , como os outros são 0's, (a) = 128

hexadecimal  $\rightarrow$  010001000

$\downarrow \quad \downarrow \quad \downarrow$   
 0 8 0  $\rightarrow$  80 em hexadecimal

b)  $0^0 1^9 0^8 1^6 0^5 1^4 1^3 0^2 1^1$

decimal  $\rightarrow 2^9 + 2^6 + 2^4 + 2^3 + 2^2 + 1 = 605$



hexadecimal  $\rightarrow 010, 0101, 1101$

2 5 13  $\rightarrow 25D$

c)  $0^9 0^8 1^7 0^6 0^5 0^4 0^3 1^2 1^1$

decimal  $\rightarrow 2^7 + 2^2 + 1 = 131$

hexadecimal  $\rightarrow 00, 1000, 0011$

0 8 3  $\rightarrow 83$

5 - De hexadecimal para binário

a)  $0x5567$  binário  $\rightarrow 0101, 0101, 0110, 0111$

b)  $0xABABAB$  binário  $\rightarrow 1010, 1011, 1010, 1011, 1010, 1011$

c)  $0xCFF$  binário  $\rightarrow 1100, 1111, 1111$

6 - Encontrar negativo com complemento de 2

a) $00000001$	b) $01000000$	c) $01010110$
$+ 11111111$	$+ 11000000$	$+ 10101010$
$\hline 10000000$	$\hline 10000000$	$\hline 00000000$

7 - Maior n° positivo e negativo em addi

De  $0$  a  $2.047$  e  $-2048$  a  $-1$ .

8 - Somar  $-120, 30$  e  $1$  em assembly RISC-V

00	addi T0, x0, -120
04	addi T1, x0, 30
08	addi T6, T0, 1
0C	add T6, T6, T1
10	jal x0, 0



9 - Obtém dois inteiros e salva o maior no registrador x5.

```

0 addi T1, zero, 32
4 lw    T2, 0(T1)
8 lw    T3, 4(T1)
12 blt  T2, T3, +12
16 add  T0, zero, T2
20 jal  x0, 0
24 add  T0, zero, T3
28 jal  x0, 0

```

32 e 36 são a posição para guardar números a serem comparados.

10 - Some de 1 a 100 e guarde o resultado em x0.

```

addi T1, zero, 101
addi T2, zero, 1
add  T0, T0, T2
addi T2, T2, 1
bne  T2, T1, -8
jal  x0, 0

```

Resultado = 5050  $\rightarrow$  0x13BA

11 - Lê um inteiro da memória e guarda seu negativo em x5  $\rightarrow$  (T0)

```

addi T1, x0, 64
lw    T2, 0(T1)
bge  T2, x0, +28
addi T1, x0, 0
addi T1, T1, 1
add  T3, T1, T2
bne  T3, x0, -8
add  T0, x0, T1

```

```

jal  x0, 0
addi T1, x0, 0
addi T1, T1, -1
add  T3, T2, T1
bne  T3, x0, -8
add  T0, x0, T1
jal  x0, 0

```

12 - Divide o maior pelo menor e retorna o quociente em x5

```

addi T1, x0, 60
lw    T2, 0(T1)
lw    T3, 4(T1)
addi T1, x0, 0
bge  T2, T3, +24
sub  T3, T3, T2
addi T1, T1, 1
bge  T2, x0, -8
addi T0, T1, -1

```

```

bge  T3, x0, -8
addi T0, T1, -1
jal  x0, 0
sub  T2, T2, T3
addi T1, T1, 1
bge  T2, x0, -8
addi T0, T1, -1

```

```

jal  x0, 0

```



13 - Divide o maior pelo menor e x5(T0) recebe o resto.

addi x6, x0, 48	}	add x5, x28, x7
lw x7, 0(x6)		jal x0, 0
lw x28, 4(x6)		sub x7, x7, x28
bge x7, x28, +20		bge x7, x0, -4
sub x28, x28, x7		add x5, x7, x28
bge x28, x0, -4		jal x0, 0

14 - Converta a string "RONCADOR" para minúsculo e imprima no display

```
lui T1, 0x00000000
addi T0, x0, 40
lb T2, 0(T0)
addi T2, T2, 32
sb T2, 0(T0)
sb T2, 0(T1)
addi T0, T0, 1
bne T2, x0, -20
jal x0, 0
```