

Sebastian Serna Gómez  
Código 1088283998

```
Int main () {  
  Int i= 20; → %L0  
  Int j=11; → %L1  
  If (i!=j) {  
    i++;  
  }  
  else {  
    i --  
  }  
  Return i+ j;  
}
```

Se definen los registros de la siguiente manera:

i = %L0  
j = % L1

Se pasa a lenguaje ensamblador de la siguiente manera:

```
0x0000  MOV 20, %L0  
0x0004  MOV 11, %L1  
0x0008  CMP %L0, %L1  
0x000C  BE a, FALSE  
0x 0010  ADD %L0,1,%L0  
0x0014  BA a, EXIT  
FALSE  
0x0018  SUB %L0,1,%L0  
EXIT  
0x001C  ADD %L0,%L1, %O0
```

Para pasar a lenguaje de máquina se usan los diferentes formatos (Formato 3 o 2), de acuerdo a la instrucción de la siguiente manera:

Formato3

OP	RD	OP3	RS1	i	
10	10000	000010	00000	1	0000000010100

Formato3

OP	RD	OP3	RS1	i	
10	10001	000010	00001	1	0000000001011

Formato3

OP	RD	OP3	RS1	i	Unused	RS2
10	00000	010100	10000	0	00000000	10001

Formato2

OP	a	COND	OP2	Disp22		
00	1	0001	010	0000000000000000000011		

Formato3

OP	RD	OP3	RS1	i	
10	10000	000000	10000	1	00000000000001

Formato2

OP	a	COND	OP2	Disp22		
00	1	1000	010	000000000000000000010		

Formato3

OP	RD	OP3	RS1	i	
10	10000	000100	10000	1	00000000000001

Formato3

OP	RD	OP3	RS1	i	Unused	RS2
10	01000	000000	10000	0	00000000	10001

```

Int main() {
    Int i= - 19600; → %L0
    b[2]=33;→ %L1 y para inicializar el 33 se asigna
registro %L2
    If (i – b[2])>12{
        Return i + 15;
    }
    Else
        Return i * 32
    }
}

```

Se definen los registros de la siguiente manera:

```

i = %L0
b[2]= % L1

```

Para -19600 se realiza complemento a 2, de la siguiente manera:

```

00000000000000000000100110010010000
11111111111111111011001101101111
                                     1
-----
1111111111111111101100|1101110000
0000000000000000000010011
                             1
-----
0000000000000000000010100

```

Se pasa a lenguaje ensamblador de la siguiente manera:

```

0x0000  SETHI -20 , %L0
0x0004  OR %L0,880 , %L0
0x0008  MOV 33,%L2
0x000C  ST %L2,[%L1+(2*4)]
0x0010  SUB %L1, %L2, %L2
0x0014  BL a, FALSE
0x0018  ADD %L0, %L2, %O0
0x001C  BA a, EXIT
FALSE
0x0020  SLL %L0,32,%O1
EXIT
0x0024  NOP

```