

Ejercicio

```
int main (){
    int i=20;
    int j=11;
    if (i!=j){
        i++;
    }else{
        i--;
    }
    return i+j;
}
```

IMPLEMENTACION

- Definir los registros de las variables
- Inicializar las variables en cero con la variable global [0](%G0) utilizando la variable sintética MOV
- Realizar las operaciones entre los registros, se utiliza la variable sintética CMP del operador SUB para hacer comparación.
- Volver a usar los registros que no se usan mas
- Una vez se tenga obtenga la conversión de lenguaje ensamblador puedo hacer la conversión a Lenguaje Máquina (Binario)
- Utilizar el formato #3, el cual se utiliza para las instrucciones aritmético-lógicas y desplazamientos (SHIFT) y utilizó y formato #2
- Instrucciones de Salto (SETHI, BRANCH y NOP), en caso de no realizar las operaciones anteriores se puede utilizar el formato #1 Instrucciones de llamado (CALL)
- Utilizar el formato OP de las operaciones a realizar, en este caso se utiliza el OP 10 (Aritmético-lógica y shift) y OP 00 (Branch, Nop y Sethi).
- Utilizar el formato OP3 de las instrucciones y OP2, en este caso se utiliza el OP3 del OR-MOV (000010), ADD (000000), SUB (000100) y el OP2 del BA y BGE (010), NOP (100), SETHI (100).
- Diligenciar el formato #3 (ADD, SUB, OR y SHIFT) y formato #2 (BRANCH, NOP y SETHI) de cada

LENGUAJE ENSAMBLADO

R
Definición de registros
i=%L0
j=%L1
Inicialización de variables
mov 20,%L0
mov 11,%L1
Ejecución de operaciones
BE i FALSE
ADD %L0,1,%L0
BA i EXIT
FALSE
SUB %L0,1,%L0
EXIT
ADD %L0,%L1,%L0
EXIT
NOP

DIRECCIONES	OP	RD	OP3	RS1	i	Unusued/zero	RS2
OX0000	10	10000	000010	00000	1	00000000101000	
OX0004	10	10001	000010	00000	1	000000001011	
	OP	a	cond	OP2	disp22		
OX0008	00	1	0001	0010	000000000000000010100		
	OP	RD	OP3	RS1	i	Unusued/zero	RS2
OX000C	10	10000	000000	10000	1	0000000000001	
	OP	a	cond	OP2	disp22		
OX0010	00	1	1000	0010	000000000000000010100		
	OP	RD	OP3	RS1	i	Unusued/zero	RS2
OX0014	10	10000	000100	10000	1	0000000000001	
OX0018	10	1000	000000	10000	0	00000000	10001
	OP	RD	OP2	Imm22			
OX001C	00	00000	100	000000000000000000000000			