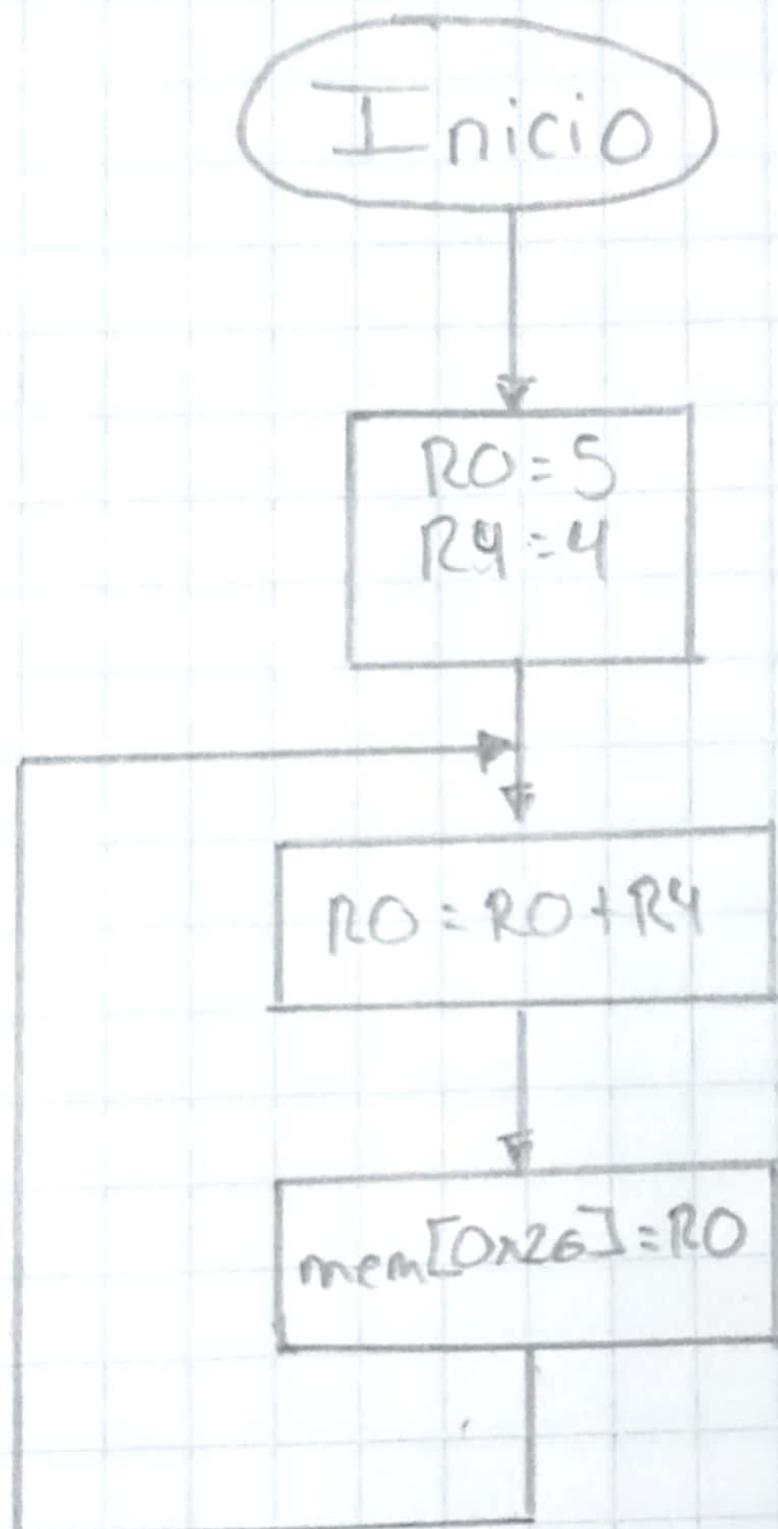


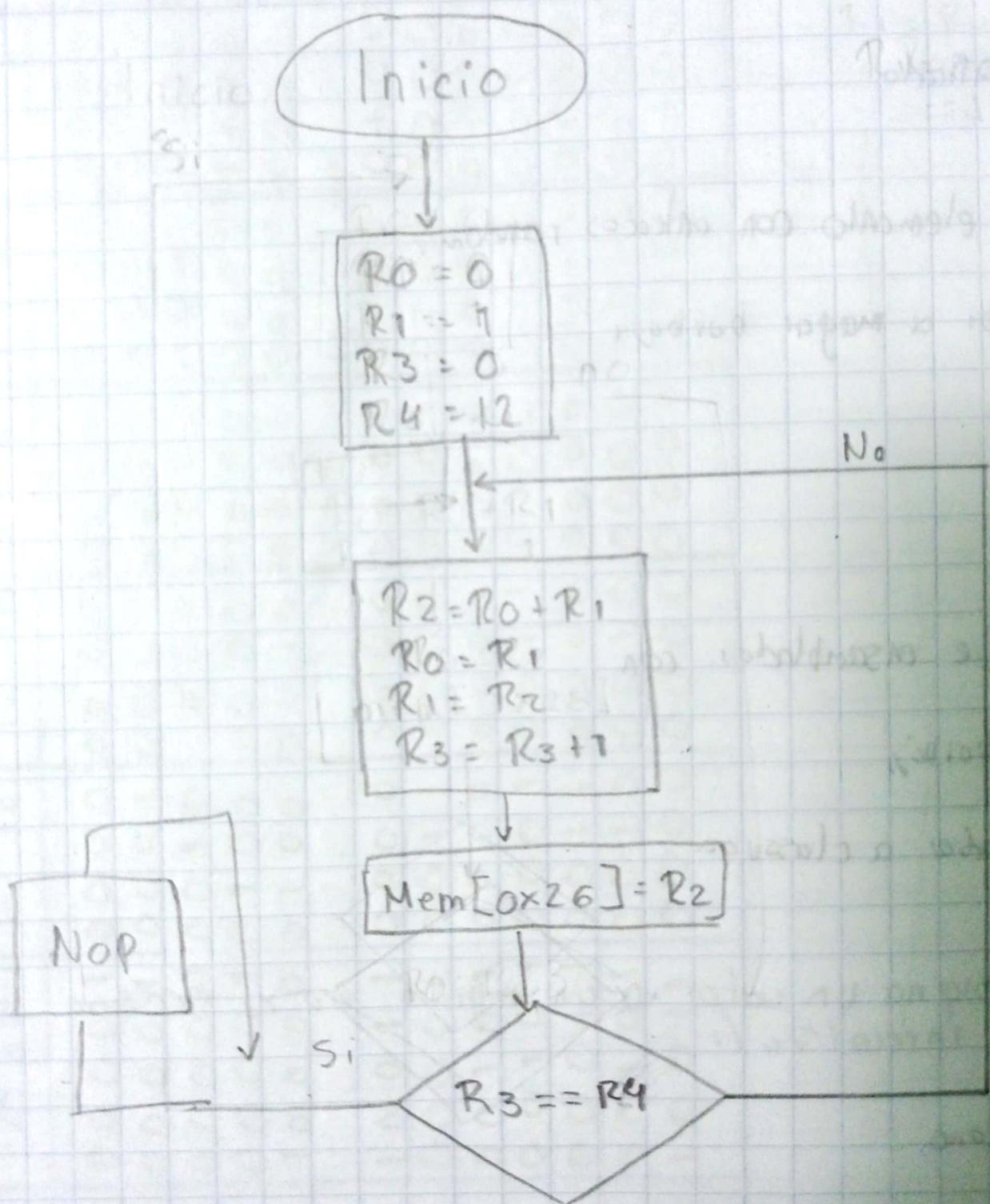
# Programa 1.



# Memoria de Programa. (Programa 1).

PC	Instrucción	Significado	24..20	19..16	15..12	11..8	7..4	3..0
0	LW R0, #5	R0 = 5	00001	0000	0000	0000	0000	0101
1	LW R4, #4	R4 = 4	00001	0100	0000	0000	0000	0100
2	ADD R0, R0, R4	R0 = R0 + R4	00000	0000	0000	0100	0000	0000
3	SWE R0, 0x26	MEM[0x26] = R0	00011	0000	0000	0000	0010	0110
4	BS Salto	PC = Salto	10011	0000	0000	0000	0000	0010

Fibonacci 12 términos, tarea.



Memoria de programa que calcula 12 términos de la Serie de Fibonacci.  
(Tarea).

PC	Instrucción	Significado	24...20	19...16	15...12	11...8	7..4	3..0
0	L1 R0, #0	$R0 = 0$	00001	0000	0000	0000	0000	0000
1	L1 R1, #1	$R1 = 1$	00001	0001	0000	0000	0000	0001
2	L1 R3, #0	$R3 = 0$	00001	0011	0000	0000	0000	0000
3	L1 R4, #12	$R4 = 12$	00001	0100	0000	0000	0000	1100
4	ADD R2, R0, R1	$R2 = R0 + R1$	00000	0010	0000	0001	0000	0000
5	ADDI R0, R1, #0	$R0 = R1 + 0$	00101	0000	0001	0000	0000	0000
6	ADDI R1, R2 +#0	$R1 = R2 + 0$	00101	0001	0010	0000	0000	0000
7	ADDI R3, R3, #1	$R3 = R3 + 1$	00101	0011	0011	0000	0000	0001
8	SWI R2, 0x26	$Mem[0x26] = R2$	00011	0010	0000	0000	0010	0110
9	BEGI R4, R3, Salto	IF( $R3 == R4$ ) $PC = 9 + 2$	01101	0100	0011	0000	0000	0010
10	NOP		10110	0000	0000	0000	0000	0000
11	B FINAL	$B, PC = 10$	10011	0000	0000	0000	0000	1010
12	B AGAIN	$B, PC = 4$	10011	0000	0000	0000	0000	0100

Memoria de programa 12 términos Fibonacci  
Ejercicio en clase.

IC	Instiucción	Significado	24 a 20	19..16	15..12	11..8	7..4	3..0
0	LI R0, #0	$R0 = 0$	00001	00000	0000	0000	0000	0000
1	LI R1, #1	$R1 = 1$	00001	0001	0000	0000	0000	0000
2	LI R2, #0	$R2 = 0$	00001	00000	0000	0000	0000	0000
3	LI R4, #12	$R4 = 12$	00001	0100	0000	0000	0000	1100
4	BLETI R4,R2,Salto	if( $R2 <= R4$ ) PC = 4+3	10000	0100	0010	0000	0000	0011
5	NOP							
6	B FIN	$PC = 5$	10011	00000	0000	0000	0000	0101
7	ADD R3,R0,R1	$R3 = R0 + R1$	0000000011	0010	0001	0000	0000	0000
8	ADDI R0,R1,#0	$R0 = R1 + 0$	00101	00000	0001	0000	0000	0000
9	ADDI R1,R3,#0	$R1 = R3 + 0$	00101	0001	0011	0000	0000	0000
10	SWI R3,0x78	Mem[0x78] = R3	00011	0011	0000	0000	0111	1000
11	ADDI R2,R2,#1	$R2 = R2 + 1$	00101	0010	0010	0000	0000	0000
12	B Iteración	$PC = 4$	10011	00000	00000	0000	0000	0100

30 / Agosto / 2021

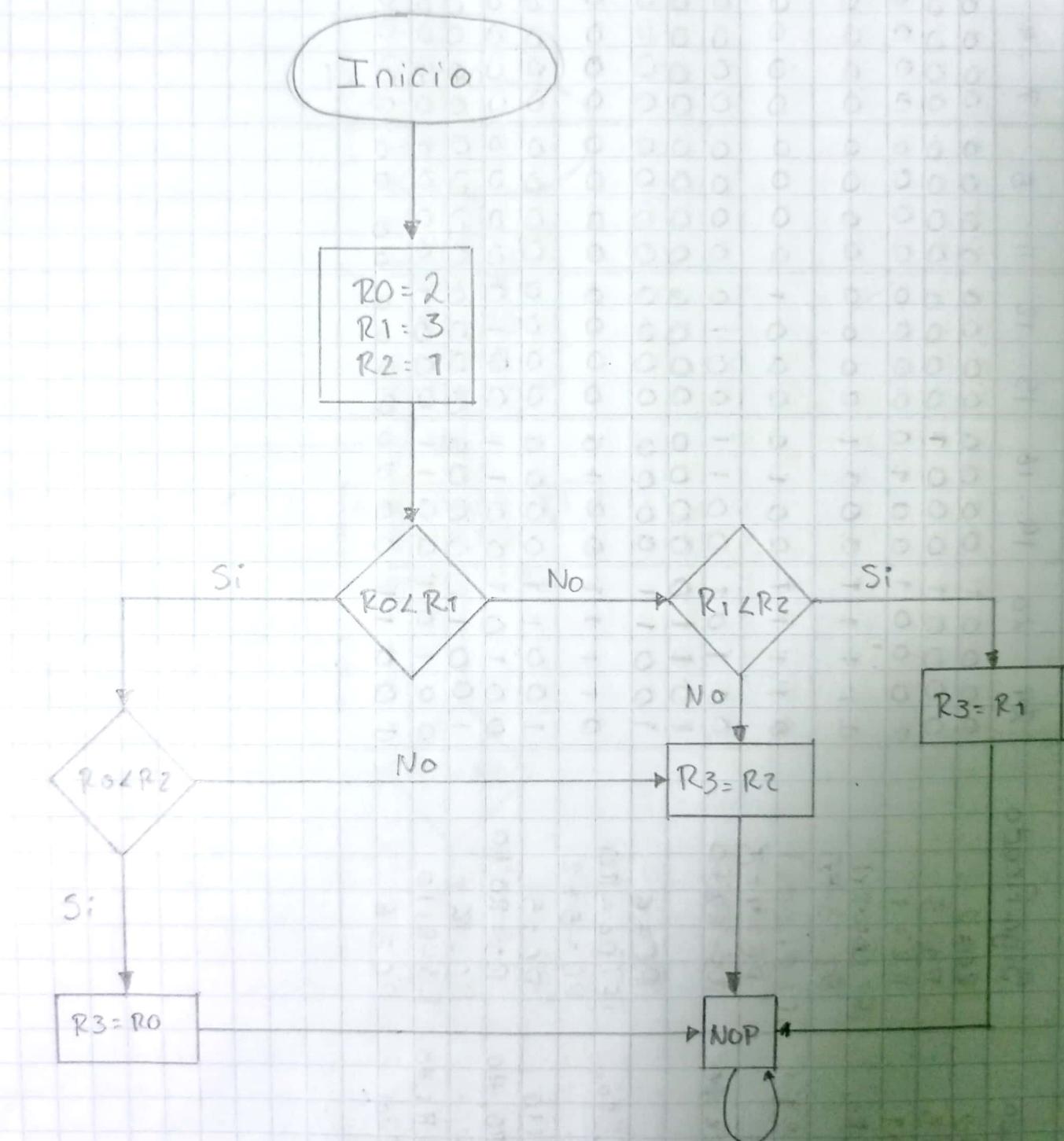
# Memoria de Programa que encuentra el menor de 3 números.

Ejemplo de Clase.

2 de Septiembre 2021

PC	Instrucción	Significado	24... 20	19... 16	15... 12	11... 8	7... 4	3... 0
0	LI R0, #75	$R0 = 75$	0 0 0 0 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	1 0 0 1 0 1 1	
1	LI R1, #2	$R1 = 2$	0 0 0 0 1	0 0 0 1 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 1 0
2	LI RS, #234	$RS = 234$	0 0 0 0 1	0 1 0 1 0	1 0 0 0 0	0 0 0 0 1	1 1 1 0 1 0 0	
3	BLTI, R0, R1 First	$IF(R1 < R0)$ $PC = 3 + 5$	0 1 1 1 1	0 0 0 0 0	0 0 0 1	0 0 0 0 0	0 0 0 0 0	1 0 1
4	BLTI, R0, RS Second	$IF(RS < R0)$ $PC = 4 + 6$	0 1 1 1 1	0 0 0 0 0	1 0 1	0 0 0 0 0	0 0 0 0 0	0 1 1 0
5	ADDI, RS, R0, H0	$RS = R0 + 0$	0 0 1 0 1	0 1 0 1 0	0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0
6	NOP		1 0 1 1 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0
7	B Fin	$PC = 6$	1 0 0 1 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 1 1 0
8	BLTI RS, R1 First (1..1)	$IF(R1 < RS)$ $PC = 8 + 4$	0 1 1 1 1	0 1 0 1 0	0 0 0 1	0 0 0 0 0	0 0 0 0 1	1 1 0 0
9	B - RS igual R0	$PC = 5$	1 0 0 1 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 1 0 1
10	ADDI R6, RS H0	$R6 = RS + 0$	0 0 1 0 1	0 1 1 0 0	1 0 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0
11	B Salto o Fin	$PC = 7$	1 0 0 1 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 1 1 1
12	ADDI R6, R1 H0	$R6 = R1 + 0$	0 0 1 0 1	0 1 1 0 0	0 0 0 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0
13	B Salto final	$PC = 7$	1 0 0 1 1	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 0 0 0 0	0 1 1 1

Diagrama de Flujo, algoritmo que calcula el menor de 3 números. (Tarcá).



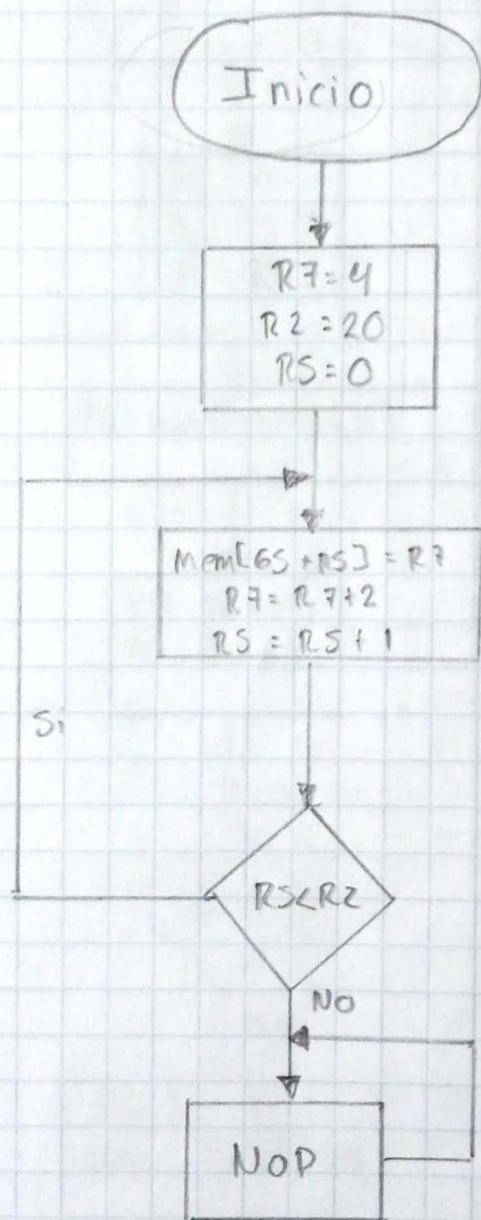
Memoria de Programa que encuentra el menor de 3 números. Tarea.

PC	Instrucción	Significado	24...20	19...16	15...12	11...8	7...4	3...0
0	LI R0, #2	$R_0 = 2$	00001	00000	0000	0000	0000	0010
1	LI R1, #3	$R_1 = 3$	00001	0001	0000	0000	0000	0011
2	LI R2, #1	$R_2 = 1$	00001	001	00000	0000	0000	0001
3	BLTI R1, R0, C1	IF( $R_0 < R_1$ ) $PC = 3 + 5$	01111	0001	0000	0000	0000	0101
4	BLTI R2, R1, C1, 7	IF( $R_1 < R_2$ ) $PC = 4 + 8$	01111	0010	0001	0000	0000	1000
5	ADDI R3, R2, #0	$R_3 = R_2 + 0$	00101	0011	0010	0000	0000	0000
6	NOP		10110	0000	0000	0000	0000	0000
7	B Final	$PC = 6$	10011	0000	0000	0000	0000	0110
8	BLTI R2, R0, C1, 1, 2	IF( $R_0 < R_2$ ) $PC = 6 + 2$	01111	0010	0000	0000	0000	0010
9	B R3=R2+0	$PC = 5$	10011	0000	0000	0000	0000	0101
10	ADDE R3, R0, #0	$R_3 = R_0 + 0$	00101	0011	0010	0000	0000	0000
11	B Salto F	$PC = 6$	10011	0000	0000	0000	0000	0110
12	ADDE R3, R1, #0	$R_3 = R_1 + 0$	00010	010011	0001	0000	0000	0000
13	B Salto Final	$PC = 6$	10011	0000	0000	0000	0000	0110

6 Septiembre 2021

## Programa 4

Iniciarizar un array de 20 localidades de memoria de datos con numeros pares comenzando por el 4.

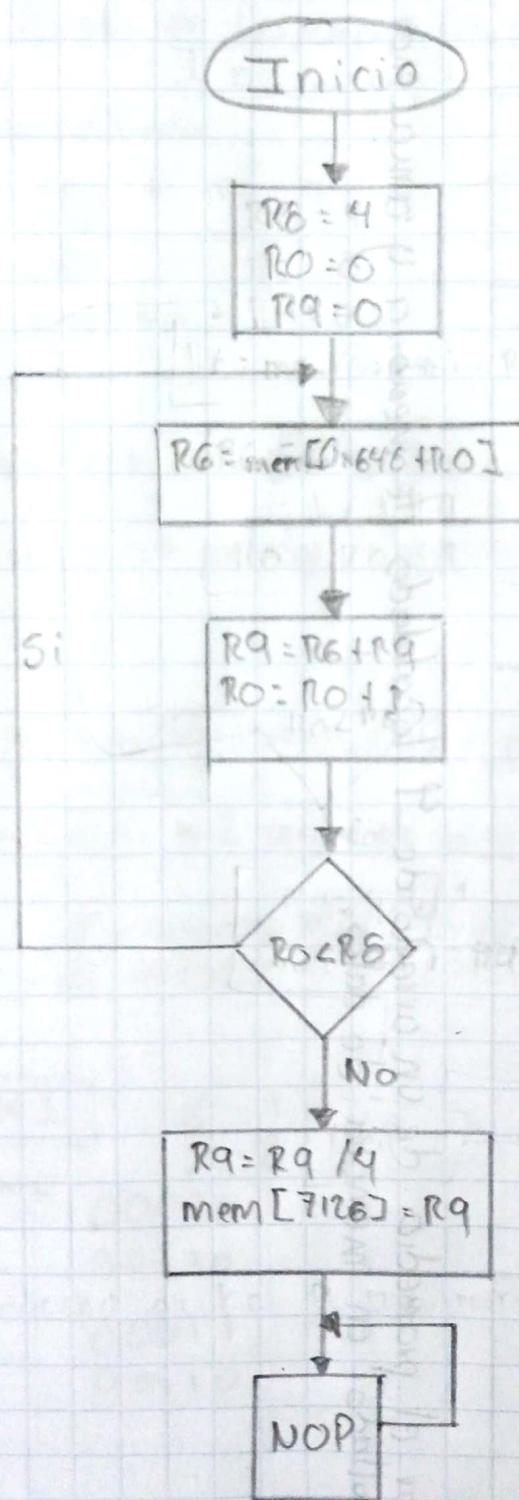


Memoria de programa que inicializa un array de 20 localidades de memoria con números pares, comenzando por el 4. (Programa 4)

PC	Instrucción	Significado	24...20	19...16	15...12	11...8	7...4	3...0
0	LI R7, #4	R7=4	00001	01110000000000000100				
1	LI R2, #20	R2=20	00001	001000000000000000010100				
2	LI RS, #0	RS=0	00001	010100000000000000000000				
3	SW, R7, 0x65(R5)	mem[0x65+RS] = R7	00100	011101000000010100101				
4	ADDI, R7, R7, #2	R7=R7+2	00101	01110111000000000000010				
5	ADDI, RS, RS, #1	RS=RS+1	00101	01010101010000000000001				
6	BLTI R2, RS	if (RS < R2)	01111	00101010111111111101				
	(aqui)	PC = 6 + (-3)						
7	NOP		10110	00000000000000000000000				
8	B FIN	PC = 7	10011	000000000000000000000111				

## Programa 5

Calcular el promedio de un arreglo de 4 localidades de memoria y almacenar el resultado en memoria de datos.



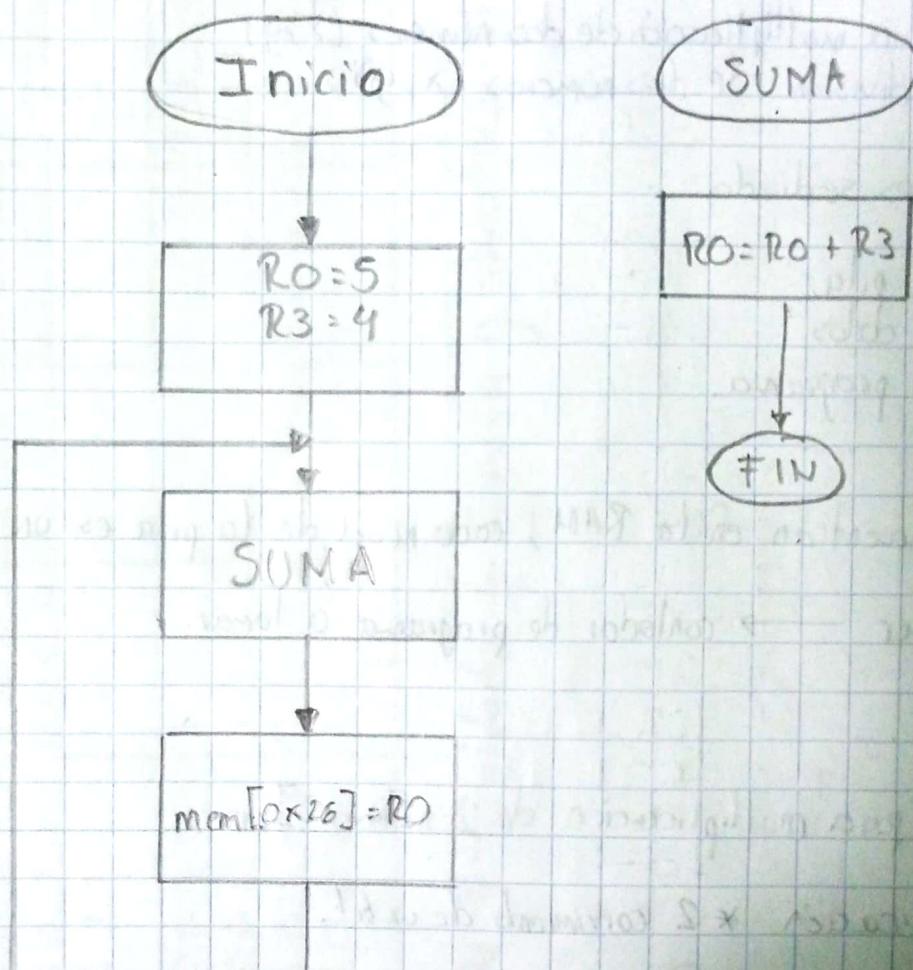
# Memoria de programa del Programa 5.

PC	Instrucción	Significado	24...20	19...16	15...12	11...8	7...4	3...0
0	LI R8, #4	R8=4	000001	100000000000000000000100				
1	LI R0, #0	R0=0	000001	00000000000000000000000000000000				
2	LI R9, #0	R9=0	000001	10010000000000000000000000000000				
3	LW R6, 0x648(R0)	R6=[0x648+R0]	101110110000000001100100	10000000000000000000000000000000	1000			
4	ADD R9, R9, R6	R9 = R9 + R6	000000100110010110	00000000000000000000000000000000	0000			
5	ADDI R0, R0, #1	R0=R0+1	00101000000000000000000000000001					
6	BLTI R8, R0, aquí	R0 < R8 PC=6+(-3)	01111100000000111111111101					
7	SRL R9, R9, 2	R9=R9>>2	0000001001100100000000101010					
8	SWI R9, 7126	mem[7126]=R9	000111001011100010010010					
9	NOP		10110000000000000000000000000000					
10	B Fin	PC=9	10011000000000000000000000000001					

Calcular el promedio de un arreglo de 4 localidades de memoria y almacenar el resultado en memoria de datos.

13 Septiembre 2021.

## Programa 6.

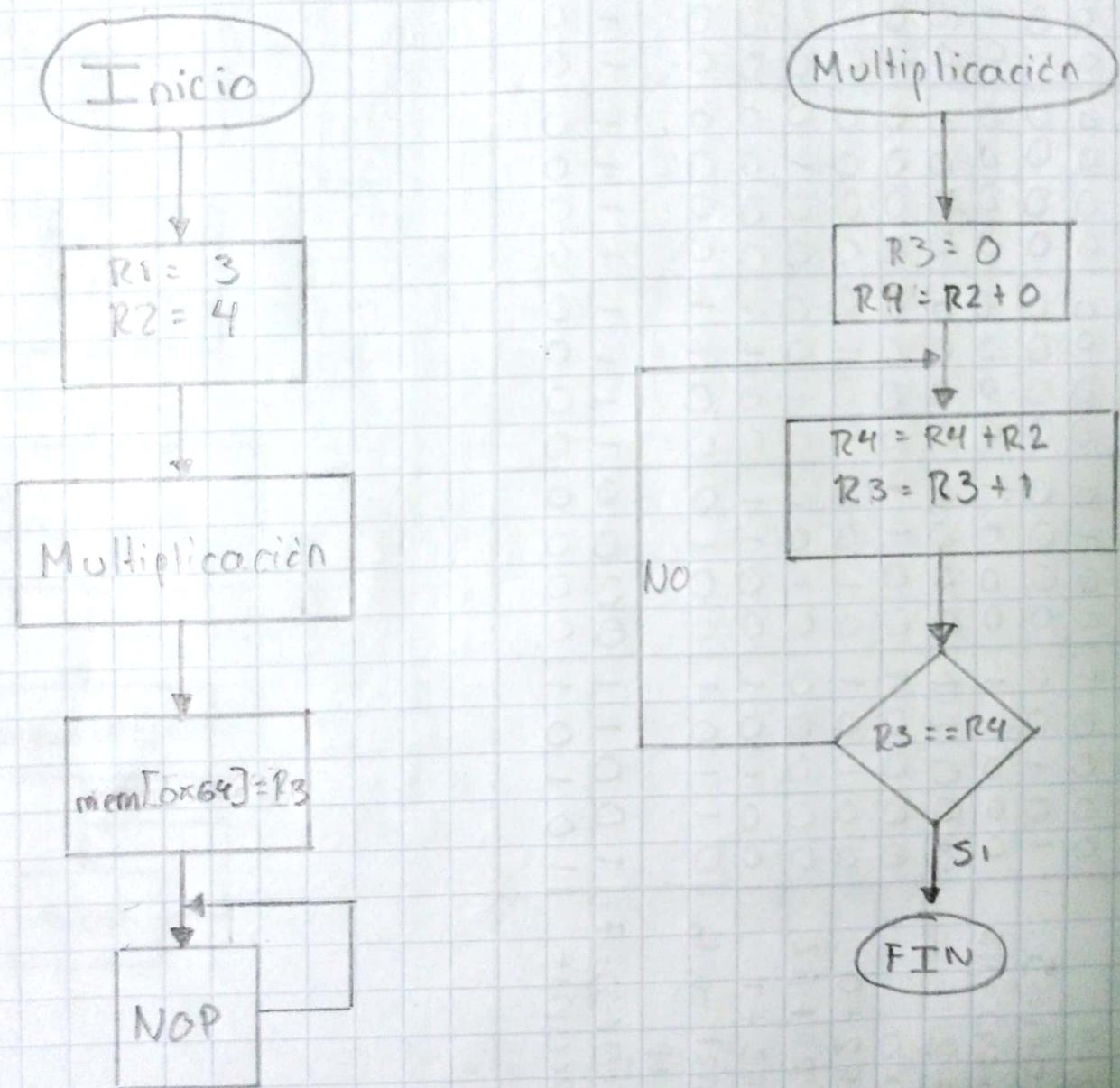


## Memoria de programa del Programa 6. (Subrutina Suma).

PC	Instrucción	Significado	24...20	19...16	15...12	11...8	7...4	3...0
0	L1 R0, #5	R0 = 5	00001	0000	0000	0000	0000	0101
1	L1 R3, #4	R3 = 4	00001	0011	0000	0000	0000	0100
2	CALL #5	SP++ PC(SP) = 5	10100	0000	0000	0000	0000	0101
3	SWI R0, 0x26	mem[0x26] = R0	00011	0000	0000	0000	00100110	
4	B Ciclo	PC = 2	10011	0000	0000	0000	0000	0010
5	ADD R0, R0, R3	R0 = R0 + R3	00000	0000	0000	0011	0000	0000
6	RET	SP-- PC(SP)++	10101	0000	0000	0000	0000	0000

1 area.

Subrutina para multiplicación de dos números ( $x, y$ ).

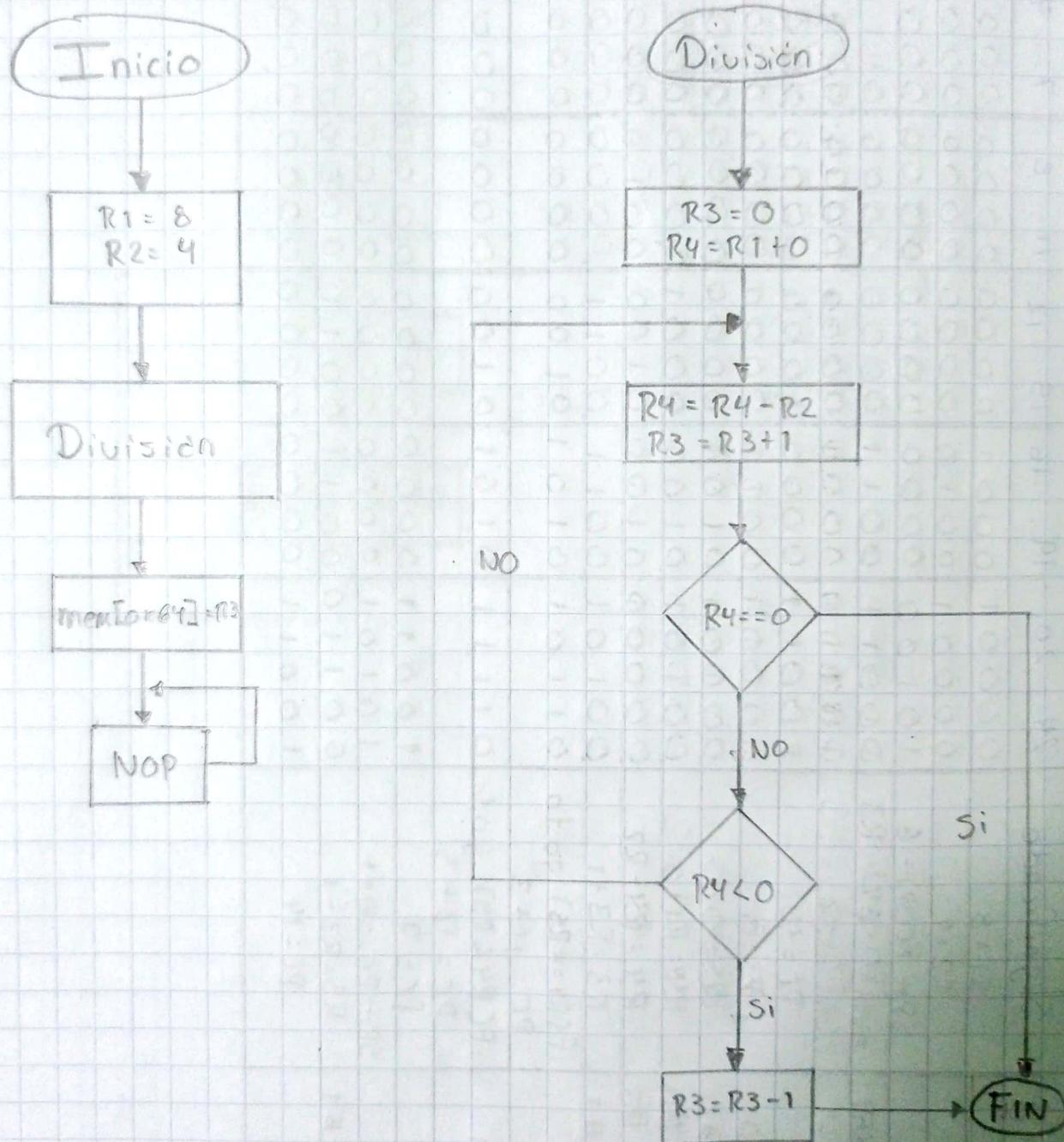


# Memoria de programa de Subrutina para multiplicación de dos números (x,y).

PC	Instrucción	Significado	24...20	19...16	15...12	11...8	7...4	3...0
0	LI R1, #3	R1 = 3	00001	0001	0000	0000	0000	0011
1	LI R2, #4	R2 = 4	00001	0010	0000	0000	0000	0100
2	CALL #5	SP++ PC(SP) = 5	10100	0000	0000	0000	0000	0101
3	SWI R1, 0x64	mem[0x64] = R3	00011	0011	0000	0000	01100100	
4	B ciclo	PC = 2	10011	0000	0000	0000	0000	0010
5	LI R3, #0	R3 = 0	00001	0011	0000	0000	0000	0010
6	ADDI R4, RL, #0	R4 = R2 + 0	00101	0100	0010	0000	0000	0000
7	ADD R4, R4, R2	R4 = R4 + R2	00000	01000100	0010010000000000			
8	ADDI R3, R3, #1	R3 = R3 + 1	00101	0011	0011	0000	0000	0001
9	BEQJ R2, R3	if(R3 == R2) goto PC = 11 PC = 11	01101	00100011	0000000000001011			
10	B repite	PC = 10 + (-3) = 7	10011	000011111111111111111101				
11	Ret	SP-- PC = PC(SP)++	10101	00000000000000000000000000				

# Tarea.

Subrutina de división de dos números ( $x, y$ ).



# Memoria de programa de Subrutina para división de dos números (x,y).

PC	Instrucción	Significado	24...20	19...16	15...12	11...8	7...4	3...0
0	LIT R1, #8	R1=8	00001	0001	0000	0000	0000	1000
1	LIT R2, #4	R2=4	00001	0010	0000	0000	0000	0100
2	CALL	SP+ PC(SP)= 6	10100	0000	0000	0000	0000	0110
3	SWI R3,0x64	mem[0x64]=R3	00011	0011	0000	0000	0110	0100
4	NOP		10110	0000	0000	0000	0000	0000
5	B FIN	PC=4	10011	0000	0000	0000	0000	0100
6	LIT R3, #0	R3=0	00001	0011	0000	0000	0000	0000
7	LIT RS, #0	RS=0	00001	0101	0000	0000	0000	0000
8	ADDI R4,R1,#0	R4=R1+0	00101	0100	0001	0000	0000	0000
9	SUB R4,R4,R2	R4=R4-R2	00000	0100	0100	0010	0000	0001
10	ADDI R3,R3,#1	R3=R3+1	00101	0011	0011	0000	0000	0001
11	BEGI RS,R4	iF(R4==RS) goto PC = 11+3	01101	0101	0100	0000	0000	0011
12	BLTE RS,R4	iF(R4<RS) goto PC = 12+3	01111	0101	0100	0000	0000	0011
13	B Ciclo	PC=9	10011	0000	0000	0000	0000	1101
14	RET	SP=PC(SP)+	10101	0000	0000	0000	0000	0000
15	SUBT R3,R3,#1	R3=R3-1	00110	0011	0011	0000	0000	0001
16	B ret	PC=14	10011	0000	0000	0000	0000	1110