

Luis Guillermo Rivera Stephens

Expediente: 746651

Luis Alberto González Escamilla

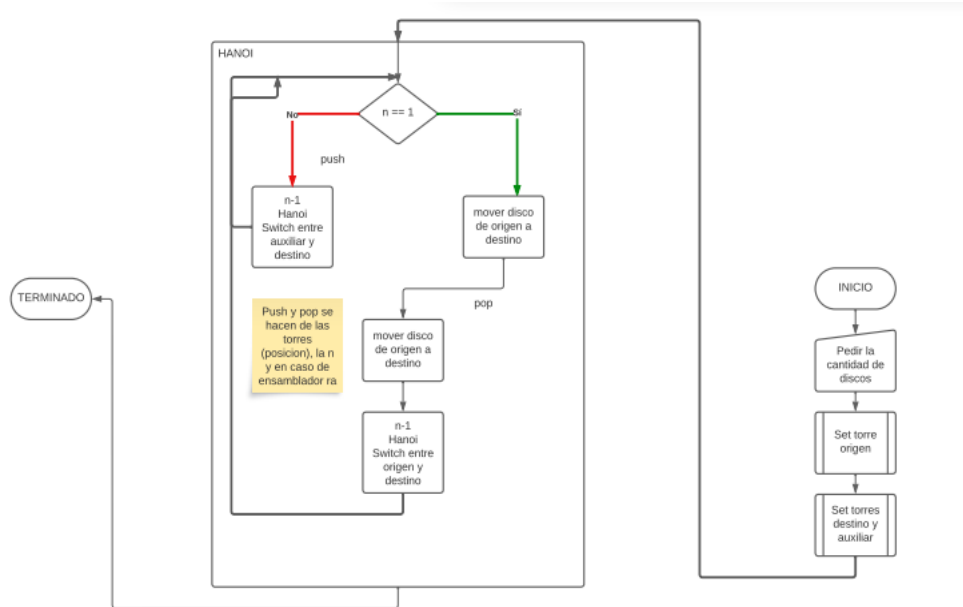
Expediente: “”



ITESO, Universidad  
Jesuita de Guadalajara

TORRES DE HANOI EN ENSAMBLADOR  
PRÁCTICA 1

# DIAGRAMA DE FLUJO



## DECISIONES AL REALIZAR EL PROGRAMA

Optamos por realizar por nuestra cuenta el algoritmo en C para entenderlo a fondo y que fuera más fácil traducir el código a ensamblador, pues sería más complicado partir de cero sin entender del todo cómo funciona el código.

Uno de los problemas que tuvimos a la hora de realizar el programa en ensamblador fue que intentamos manejar las torres como pilas, para que fuera más fácil solo meter y sacar los datos, y así asegurarnos de que solo pudiera sacar el ultimo disco.

Pero, tuvimos complicaciones a la hora de calcular la última posición del stack, entonces (por recomendación del profe) mover los discos por a su posición final sin irlos acumulando, solo verificamos que no hiciera un push de un disco más grande a una torre donde se encontraba un disco más pequeño.

Una observación que tuvimos fue que al hacer el jal hacia la función de mover discos teníamos que guardar PC + 4 en otro registro para que no afectara el valor de ra ni los valores del stack. Y no se puede hacer con un label porque no entraría a la segunda llamada a función de mover disco porque regresaría a la primera y haría de nuevo la recursión.

## Una simulación en el RARS para 3 discos (La simulación son impresiones de pantalla de data segment del RARS) .

Execute

Text Segment

Bkpt	Address	Code	Basic	Source
<input checked="" type="checkbox"/>	0x00400054	0x09800fef	jal x31,152	33: jal t6, moverDiscos
<input type="checkbox"/>	0x00400058	0x000080e7	jalr x1,x1,0	34: jalr ra
<input type="checkbox"/>	0x0040005c	0xfec10113	addi x2,x2,-20	38: addi sp, sp, -20
<input type="checkbox"/>	0x00400060	0x00a12823	sw x10,16(x2)	39: sw a0, 16(sp)
<input type="checkbox"/>	0x00400064	0x00b12623	sw x11,12(x2)	40: sw a1, 12(sp)
<input type="checkbox"/>	0x00400068	0x00c12423	sw x12,8(x2)	41: sw a2, 8(sp)
<input type="checkbox"/>	0x0040006c	0x00d12223	sw x13,4(x2)	42: sw a3, 4(sp)
<input type="checkbox"/>	0x00400070	0x00112023	sw x1,0(x2)	43: sw ra, 0(sp)
<input type="checkbox"/>	0x00400074	0xffff50513	addi x10,x10,-1	46: addi a0, a0, -1
<input type="checkbox"/>	0x00400078	0x00c00e33	add x28,x0,x12	48: add t3, zero, a2 #aux

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	3	2	1	0	0	0	0	0
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Messages Run I/O

Text Segment

Bkpt	Address	Code	Basic	Source
<input checked="" type="checkbox"/>	0x004000a0	0x04c00fef	jal x31,76	61: jal t6, moverDiscos
<input type="checkbox"/>	0x004000a4	0xfec10113	addi x2,x2,-20	64: addi sp, sp, -20
<input type="checkbox"/>	0x004000a8	0x00a12823	sw x10,16(x2)	65: sw a0, 16(sp)
<input type="checkbox"/>	0x004000ac	0x00b12623	sw x11,12(x2)	66: sw a1, 12(sp)
<input type="checkbox"/>	0x004000b0	0x00c12423	sw x12,8(x2)	67: sw a2, 8(sp)
<input type="checkbox"/>	0x004000b4	0x00d12223	sw x13,4(x2)	68: sw a3, 4(sp)
<input type="checkbox"/>	0x004000b8	0x00112023	sw x1,0(x2)	69: sw ra, 0(sp)
<input type="checkbox"/>	0x004000bc	0xffff50513	addi x10,x10,-1	72: addi a0, a0, -1
<input type="checkbox"/>	0x004000c0	0x00b00e33	add x28,x0,x11	74: add t3, zero, a1 #aux
<input type="checkbox"/>	0x004000c4	0x00c005b3	add x11,x0,x12	75: add a1, zero, a2 #torr

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	3	2	0	0	0	0	0	0
0x10010020	1	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Messages Run I/O

**Edit Execute**

**Text Segment**

Bkpt	Address	Code	Basic	Source
<input checked="" type="checkbox"/>	0x00400054	0x09800fef	jal x31,152	33: jal t6, moverDiscos
<input type="checkbox"/>	0x00400058	0x000080e7	jalr x1,x1,0	34: jalr ra
<input type="checkbox"/>	0x0040005c	0xfec10113	addi x2,x2,-20	38: addi sp, sp, -20
<input type="checkbox"/>	0x00400060	0x00a12823	sw x10,16(x2)	39: sw a0, 16(sp) ..
<input type="checkbox"/>	0x00400064	0x00b12623	sw x11,12(x2)	40: sw a1, 12(sp) ..
<input type="checkbox"/>	0x00400068	0x00c12423	sw x12,8(x2)	41: sw a2, 8(sp) ..
<input type="checkbox"/>	0x0040006c	0x00d12223	sw x13,4(x2)	42: sw a3, 4(sp) ..
<input type="checkbox"/>	0x00400070	0x00112023	sw x1,0(x2)	43: sw ra, 0(sp) ..
<input type="checkbox"/>	0x00400074	0xffff50513	addi x10,x10,-1	46: addi a0, a0, -1 ..
<input type="checkbox"/>	0x00400078	0x00c00e33	add x28,x0,x12	48: add t3, zero, a2 #aux ..

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	3	0	0	0	2	0	0	0
0x10010020	1	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0

0x10010000 (.data) ☒ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

**Messages Run I/O**

**Edit Execute**

**Text Segment**

Bkpt	Address	Code	Basic	Source
<input checked="" type="checkbox"/>	0x00400054	0x09800fef	jal x31,152	33: jal t6, moverDiscos
<input type="checkbox"/>	0x00400058	0x000080e7	jalr x1,x1,0	34: jalr ra
<input type="checkbox"/>	0x0040005c	0xfec10113	addi x2,x2,-20	38: addi sp, sp, -20
<input type="checkbox"/>	0x00400060	0x00a12823	sw x10,16(x2)	39: sw a0, 16(sp) ..
<input type="checkbox"/>	0x00400064	0x00b12623	sw x11,12(x2)	40: sw a1, 12(sp) ..
<input type="checkbox"/>	0x00400068	0x00c12423	sw x12,8(x2)	41: sw a2, 8(sp) ..
<input type="checkbox"/>	0x0040006c	0x00d12223	sw x13,4(x2)	42: sw a3, 4(sp) ..
<input type="checkbox"/>	0x00400070	0x00112023	sw x1,0(x2)	43: sw ra, 0(sp) ..
<input type="checkbox"/>	0x00400074	0xffff50513	addi x10,x10,-1	46: addi a0, a0, -1 ..
<input type="checkbox"/>	0x00400078	0x00c00e33	add x28,x0,x12	48: add t3, zero, a2 #aux ..

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	3	0	0	0	2	0	0	0
0x10010020	1	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0

0x10010000 (.data) ☒ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

**Messages Run I/O**

**Edit Execute**

**Text Segment**

Bkpt	Address	Code	Basic	Source
<input checked="" type="checkbox"/>	0x004000a0	0x04c00fef	jal x31,76	61: jal t6, moverDiscos
<input type="checkbox"/>	0x004000a4	0xfec10113	addi x2,x2,-20	64: addi sp, sp, -20
<input type="checkbox"/>	0x004000a8	0x00a12823	sw x10,16(x2)	65: sw a0, 16(sp) ..
<input type="checkbox"/>	0x004000ac	0x00b12623	sw x11,12(x2)	66: sw a1, 12(sp) ..
<input type="checkbox"/>	0x004000b0	0x00c12423	sw x12,8(x2)	67: sw a2, 8(sp) ..
<input type="checkbox"/>	0x004000b4	0x00d12223	sw x13,4(x2)	68: sw a3, 4(sp) ..
<input type="checkbox"/>	0x004000b8	0x00112023	sw x1,0(x2)	69: sw ra, 0(sp) ..
<input type="checkbox"/>	0x004000bc	0xffff50513	addi x10,x10,-1	72: addi a0, a0, -1 ..
<input type="checkbox"/>	0x004000c0	0x00b00e33	add x28,x0,x11	74: add t3, zero, a1 #aux ..
<input type="checkbox"/>	0x004000c4	0x00c005b3	add x11,x0,x12	75: add a1, zero, a2 #torr..

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	3	0	0	0	2	1	0	0
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0

0x10010000 (.data) ☒ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

**Messages Run I/O**

**Edit Execute**

**Text Segment**

Bkpt	Address	Code	Basic	Source
<input checked="" type="checkbox"/>	0x00400054	0x09800fef	jal x31,152	33: jal t6, moverDiscos
<input type="checkbox"/>	0x00400058	0x000080e7	jalr x1,x1,0	34: jalr ra
<input type="checkbox"/>	0x0040005c	0xfec10113	addi x2,x2,-20	38: addi sp, sp, -20
<input type="checkbox"/>	0x00400060	0x00a12823	sw x10,16(x2)	39: sw a0, 16(sp) ..
<input type="checkbox"/>	0x00400064	0x00b12623	sw x11,12(x2)	40: sw a1, 12(sp) ..
<input type="checkbox"/>	0x00400068	0x00c12423	sw x12,8(x2)	41: sw a2, 8(sp) ..
<input type="checkbox"/>	0x0040006c	0x00d12223	sw x13,4(x2)	42: sw a3, 4(sp) ..
<input type="checkbox"/>	0x00400070	0x00112023	sw x1,0(x2)	43: sw ra, 0(sp) ..
<input type="checkbox"/>	0x00400074	0xffff50513	addi x10,x10,-1	46: addi a0, a0, -1 ..
<input type="checkbox"/>	0x00400078	0x00c00e33	add x28,x0,x12	48: add t3, zero, a2 #aux ..

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0	0	0	0	2	1	3	0
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0

0x10010000 (.data) ☒ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

**Messages Run I/O**

**Edit Execute**

**Text Segment**

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400078	0x00c00e33	add x28,x0,x12	48: add t3, zero, a2 #aux ..
<input type="checkbox"/>	0x0040007c	0x00d00633	add x12,x0,x13	49: add a2, zero, a3 #torr..
<input type="checkbox"/>	0x00400080	0x01c006b3	add x13,x0,x28	50: add a3, zero, t3 #torr..
<input type="checkbox"/>	0x00400084	0xfcdff0ef	jal x1,-52	52: jal hanoi
<input type="checkbox"/>	0x00400088	0x00012083	lw x1,0(x2)	55: lw ra,0(sp)
<input type="checkbox"/>	0x0040008c	0x00412683	lw x13,4(x2)	56: lw a3, 4(sp)
<input type="checkbox"/>	0x00400090	0x00812603	lw x12,8(x2)	57: lw a2, 8(sp)
<input type="checkbox"/>	0x00400094	0x00c12583	lw x11,12(x2)	58: lw a1, 12(sp)
<input type="checkbox"/>	0x00400098	0x01012503	lw x10,16(x2)	59: lw a0, 16(sp)
<input type="checkbox"/>	0x0040009c	0x01410113	addi x2,x2,20	60: addi sp,sp,20
<input checked="" type="checkbox"/>	0x004000a0	0x04c00fef	jal x31,76	61: jal t6, moverDiscos

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0	0	1	0	2	0	3	0
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0

0x10010000 (.data) ☒ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

**Messages Run I/O**

**Edit Execute**

**Text Segment**

Bkpt	Address	Code	Basic	Source
<input checked="" type="checkbox"/>	0x00400054	0x09800fef	jal x31,152	33: jal t6, moverDiscos
<input type="checkbox"/>	0x00400058	0x000080e7	jalr x1,x1,0	34: jalr ra
<input type="checkbox"/>	0x0040005c	0xfec10113	addi x2,x2,-20	38: addi sp, sp, -20
<input type="checkbox"/>	0x00400060	0x00a12823	sw x10,16(x2)	39: sw a0, 16(sp) ..
<input type="checkbox"/>	0x00400064	0x00b12623	sw x11,12(x2)	40: sw a1, 12(sp) ..
<input type="checkbox"/>	0x00400068	0x00c12423	sw x12,8(x2)	41: sw a2, 8(sp) ..
<input type="checkbox"/>	0x0040006c	0x00d12223	sw x13,4(x2)	42: sw a3, 4(sp) ..
<input type="checkbox"/>	0x00400070	0x00112023	sw x1,0(x2)	43: sw ra, 0(sp) ..
<input type="checkbox"/>	0x00400074	0xffff50513	addi x10,x10,-1	46: addi a0, a0, -1 ..
<input type="checkbox"/>	0x00400078	0x00c00e33	add x28,x0,x12	48: add t3, zero, a2 #aux ..

**Data Segment**

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0	0	1	0	0	0	3	2
0x10010020	0	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0

0x10010000 (.data) ☒ Hexadecimal Addresses ☐ Hexadecimal Values ☐ ASCII

**Messages Run I/O**

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
<input checked="" type="checkbox"/>	0x00400054	0x09800fef	jal x31,152	33: jal t6, moverDiscos
<input type="checkbox"/>	0x00400058	0x000080e7	jalr x1,x1,0	34: jalr ra
<input type="checkbox"/>	0x0040005c	0xfec10113	addi x2,x2,-20	38: addi sp, sp, -20
<input type="checkbox"/>	0x00400060	0x00a12823	sw x10,16(x2)	39: sw a0, 16(sp)
<input type="checkbox"/>	0x00400064	0x00b12623	sw x11,12(x2)	40: sw a1, 12(sp)
<input type="checkbox"/>	0x00400068	0x00c12423	sw x12,8(x2)	41: sw a2, 8(sp)
<input type="checkbox"/>	0x0040006c	0x00d12223	sw x13,4(x2)	42: sw a3, 4(sp)
<input type="checkbox"/>	0x00400070	0x00112023	sw x1,0(x2)	43: sw ra, 0(sp)
<input type="checkbox"/>	0x00400074	0xffff50513	addi x10,x10,-1	46: addi a0, a0, -1
<input type="checkbox"/>	0x00400078	0x00c00e33	add x28,x0,x12	48: add t3, zero, a2 #aux

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0	0	0	0	0	0	3	2
0x10010020	1	0	0	0	0	0	0	0
0x10010040	0	0	0	0	0	0	0	0
0x10010060	0	0	0	0	0	0	0	0
0x10010080	0	0	0	0	0	0	0	0
0x100100a0	0	0	0	0	0	0	0	0
0x100100c0	0	0	0	0	0	0	0	0
0x100100e0	0	0	0	0	0	0	0	0

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

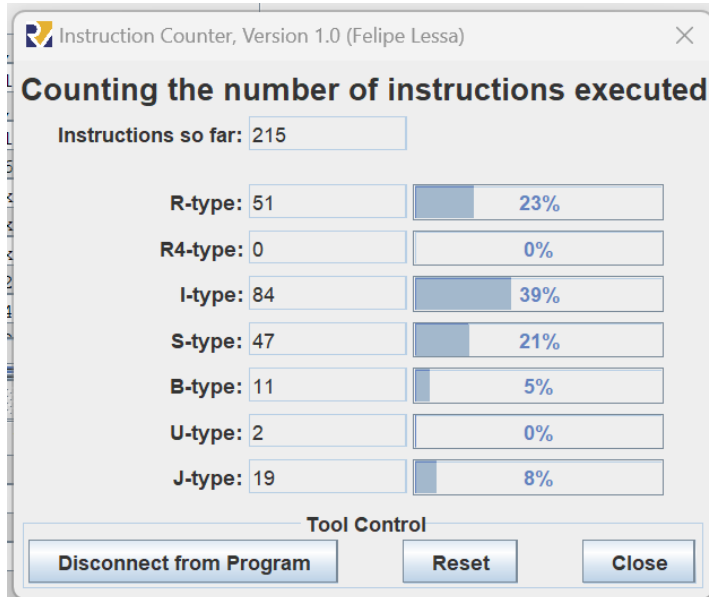
Messages Run I/O

## Análisis del comportamiento del stack para el caso de 3 discos.

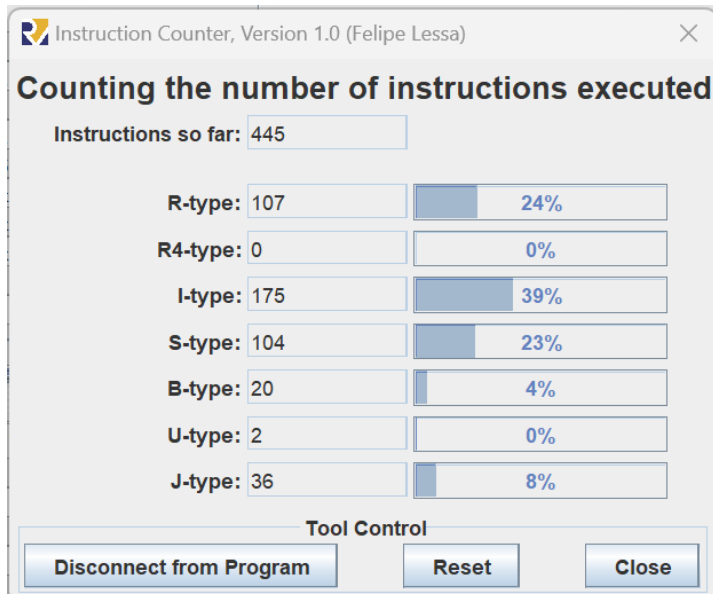
```
#push
addi sp, sp, -20
sw a0, 16(sp)      #push de n
sw a1, 12(sp)      #push de torre origen
sw a2, 8(sp)       #push de torre auxiliar
sw a3, 4(sp)       #push de torre destino
sw ra, 0(sp)       #push de PC+4
#####
#modificacion de datos
addi a0, a0, -1    #n-1
#swap torres
add t3, zero, a2 #aux = torre auxiliar
add a2, zero, a3 #torre auxiliar = torre destino
add a3, zero, t3 #torre destino = aux
#recursion
jal hanoi
#####
#pop
lw ra, 0(sp)       #pop de PC+4
lw a3, 4(sp)       #pop de torre destino
lw a2, 8(sp)       #pop de torre auxiliar
lw a1, 12(sp)      #pop de torre origen
lw a0, 16(sp)      #pop de n
addi sp, sp, 20
jal t6, moverDiscos
```

**Incluir en el instruction count (IC) y especificar el porcentaje de instrucciones de tipo R, I y J para 8 discos.**

N = 3

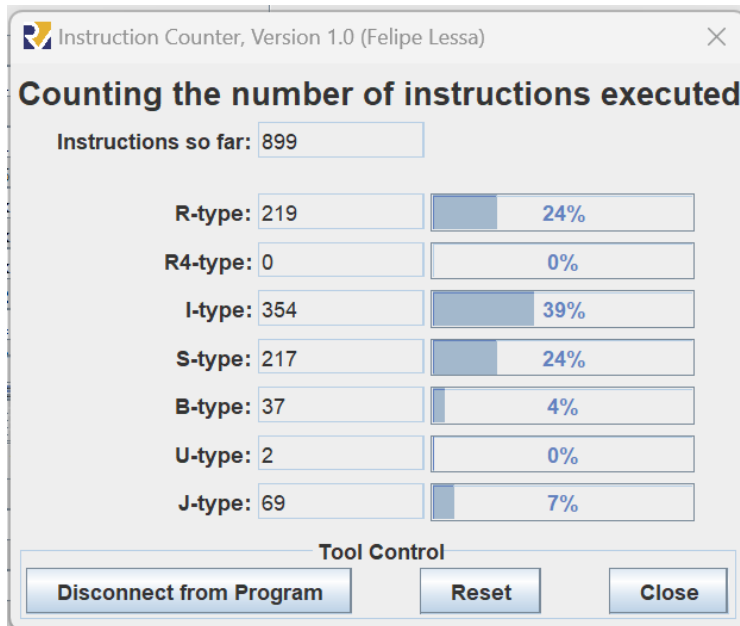


N = 4

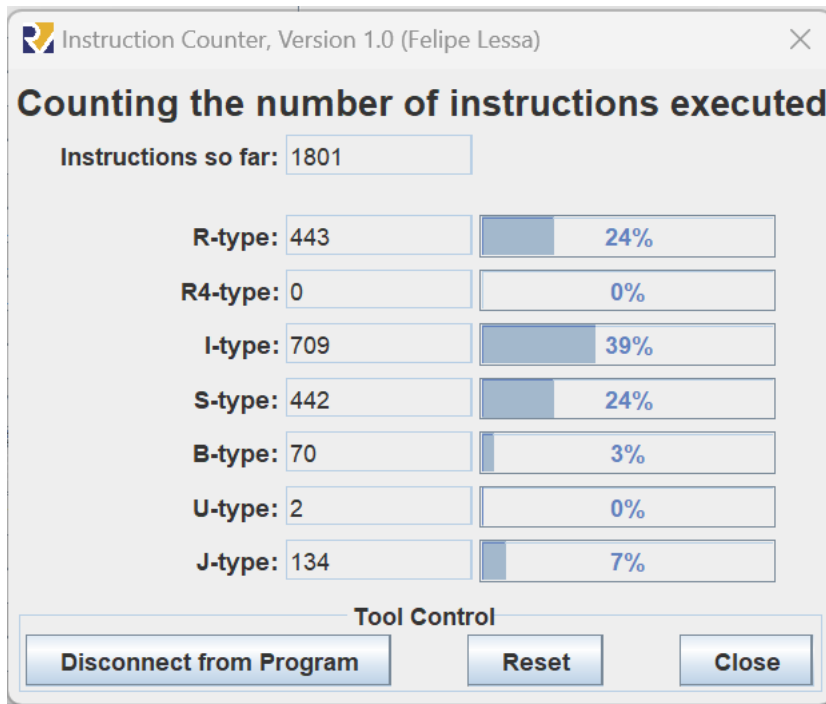




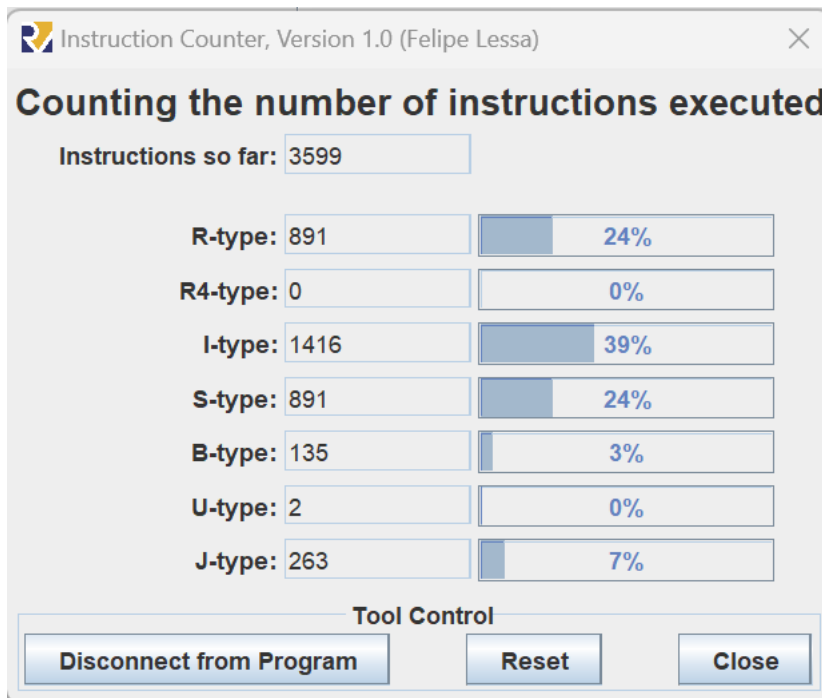
N = 5



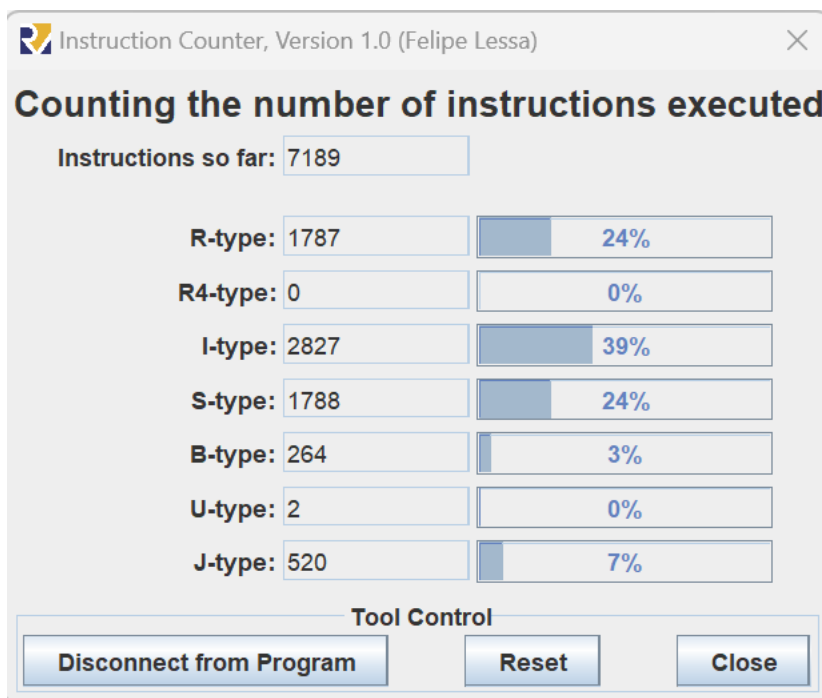
N = 6



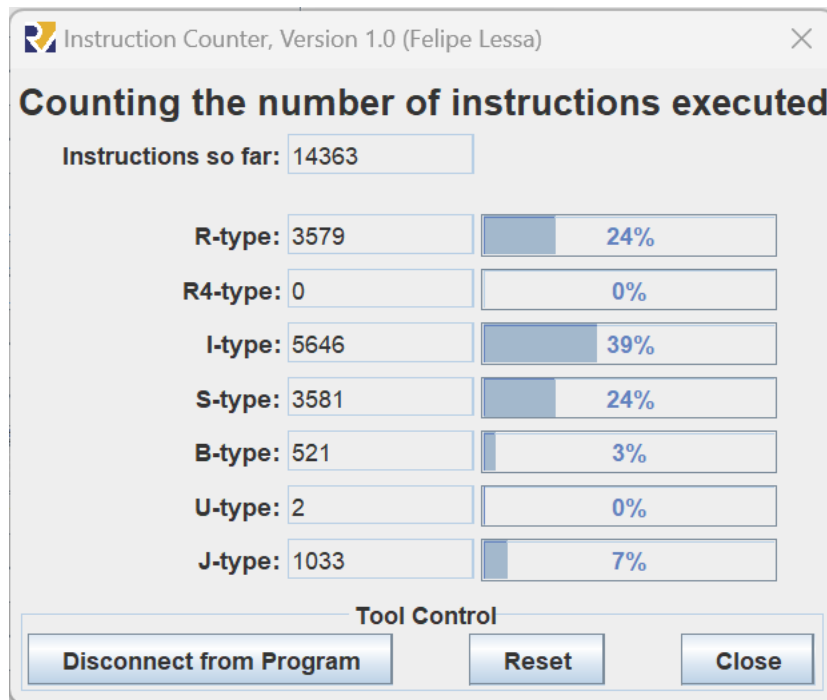
N = 7



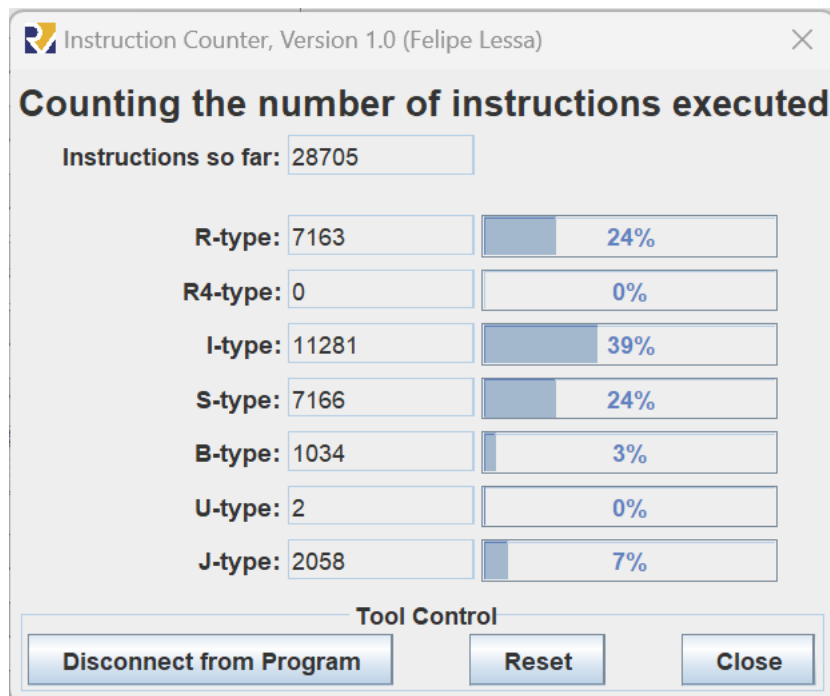
N = 8



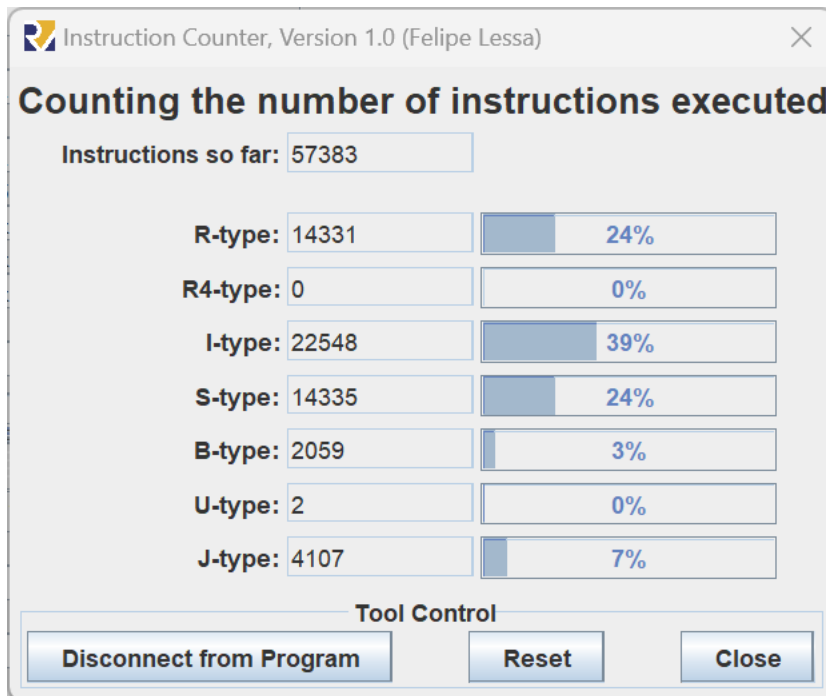
N = 9



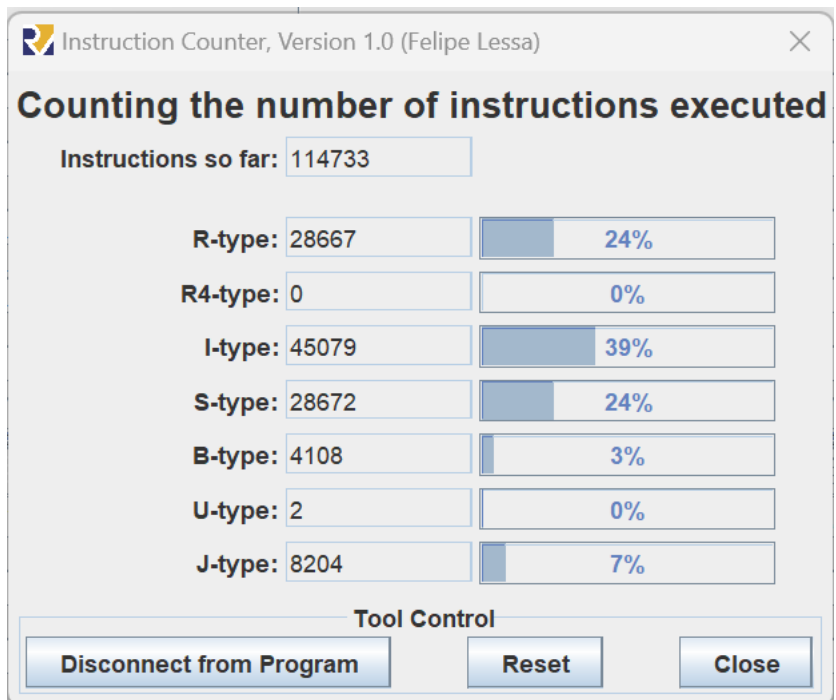
N = 10



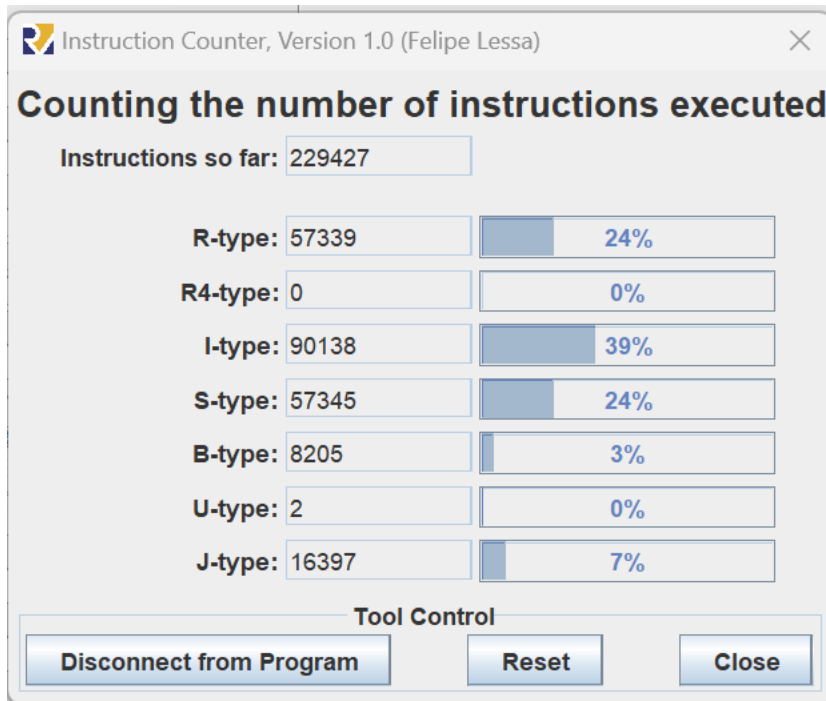
N = 11



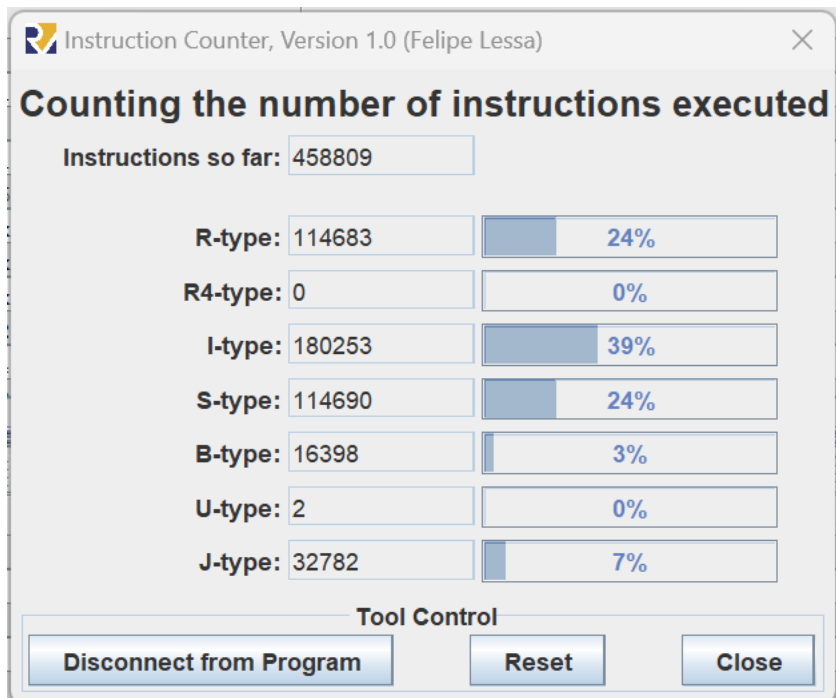
N = 12



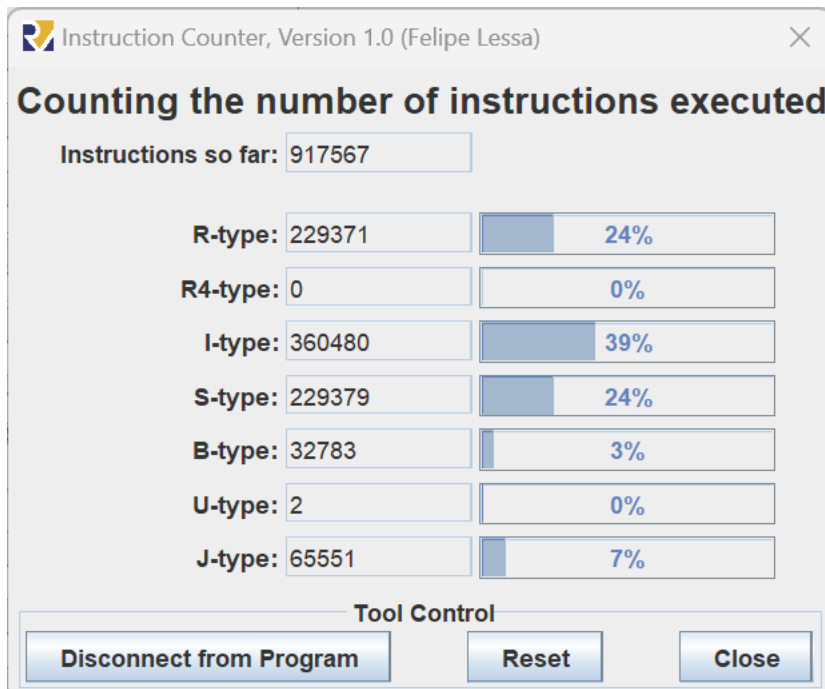
N = 13



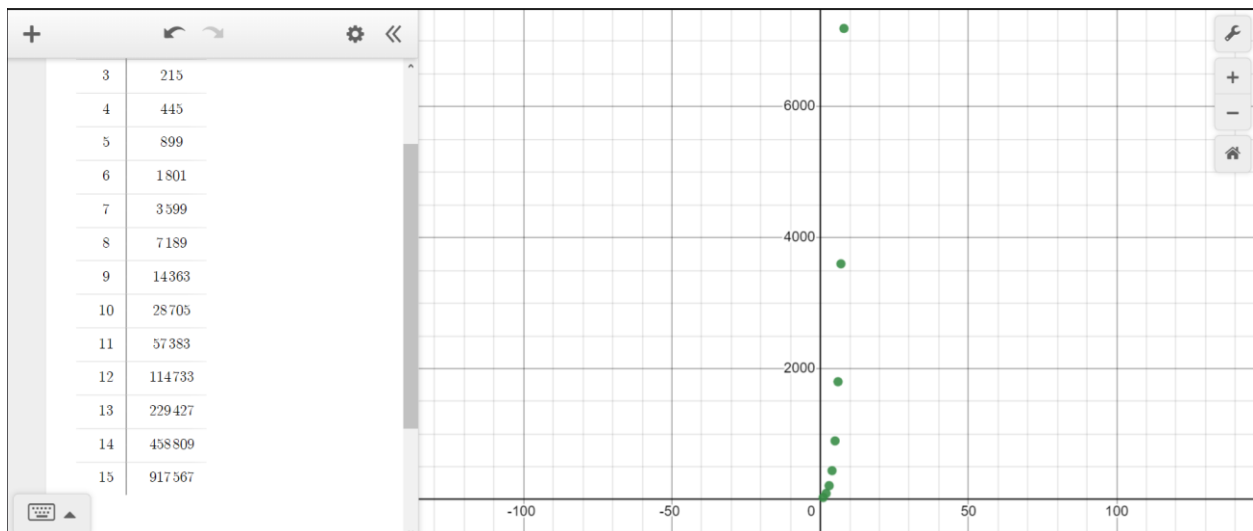
N = 14



N = 15



Una gráfica que muestre como se incrementa el IC para las torres de hanoi en los casos de 4 a 15 discos.



Podemos notar que es una función exponencial con el número de instrucciones, aunque tiene más partes la función, lamentablemente no pudimos encontrar la función con desmos.

## Conclusiones

Memo:

Puedo concluir de esta práctica que programar las Torres de Hanoi en ensamblador es más complicado de lo que pensaba, ya que no solo conllevaba el entender el código de las torres de hanoi en C lo cual tomo su dedicación, sino que también fue traducirlo a ensamblador, esto nos trajo múltiples problemas con el uso de la recursividad sin embargo puedo decir que esta práctica me ayudó mucho a aumentar mis habilidades en ensamblador.

Linguini:

Gracias a esta práctica me pude dar cuenta de lo complicado que es la recursividad siendo mi mayor problemática el entender el código de las torres de Hanoi en C lo cual me llevo un poco de tiempo hasta que logre comprender que las torres se van switcheando, pero una vez logre comprender el código el traducir este a ensamblador fue una tarea un poco más fácil.