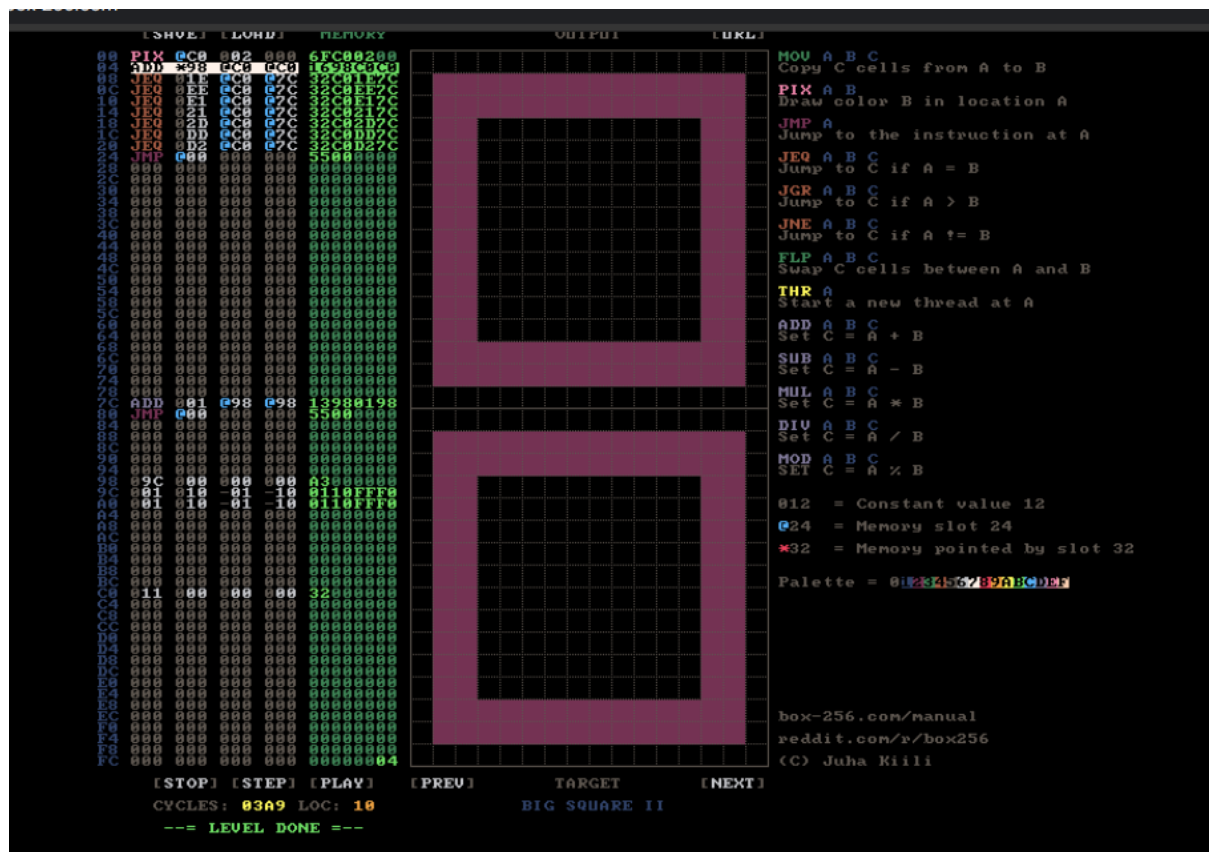


Código[illegible]

```

000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
011 000 000 000
000 000 000 000
000 000 000 000

```



• Ejercicio 3

Código

```

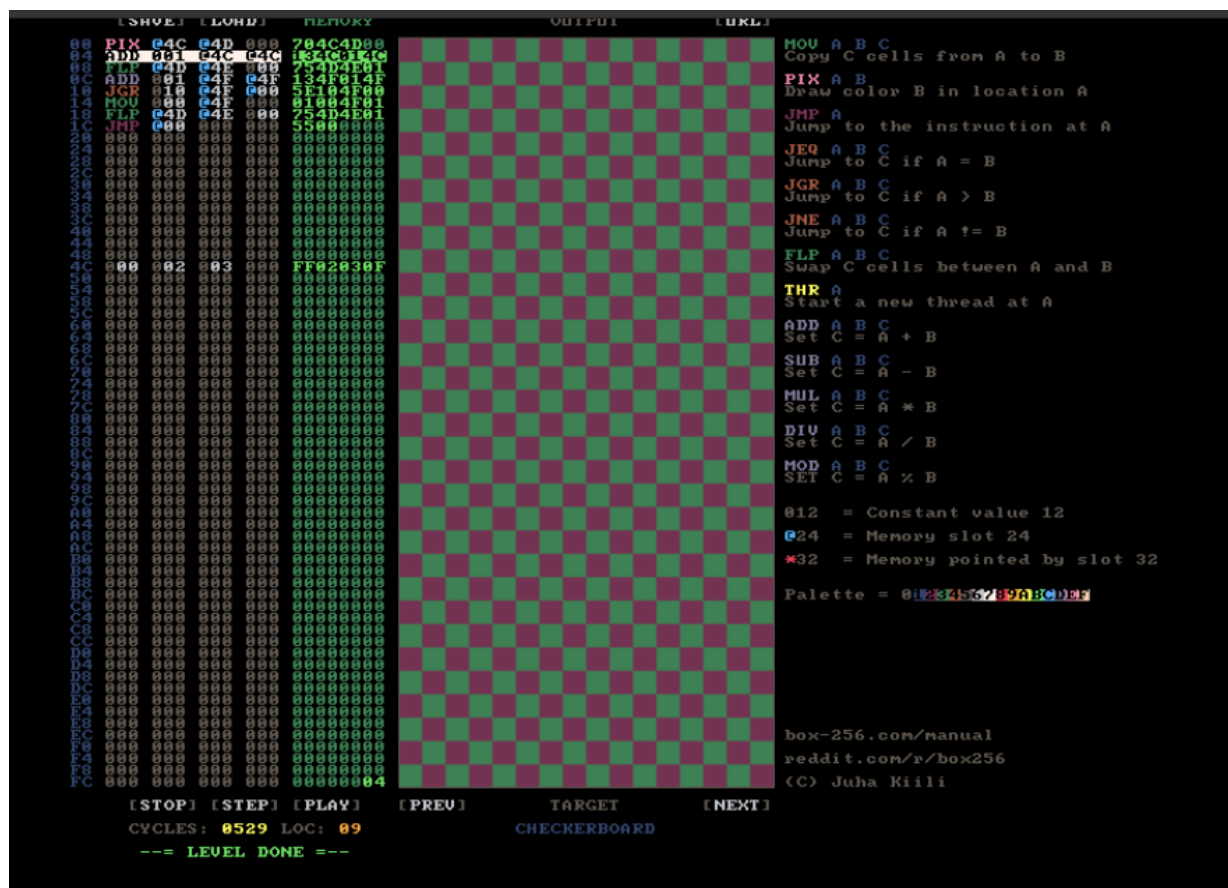
PIX @4C @4D 000
ADD 001 @4C @4C
FLP @4D @4E 000
ADD 001 @4F @4F
JGR 010 @4F @00
MOV 000 @4F 000
FLP @4D @4E 000
JMP @00 000 000
000 000 000 000
000 000 000 000

```

```

000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 002 003 000
000 000 000 000
000 000 000 000

```



• Ejercicio 4

Código

```

PIX @4C @4D 000
ADD 001 @4C @4C
ADD 001 @4F @4F
JGR 002 @4F @00
FLP @4D @4E 000
MOV 000 @4F 000
ADD 001 @50 @50

```

```

JGR 010 @50 @00
FLP @4D @4E 000
MOV 000 @50 000
JMP @00 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 001 00C 000
000 000 000 000
000 000 000 000

```

The screenshot displays the Box256 emulator interface. The top section shows assembly code with columns for instruction, address, and data. The middle section features a large checkerboard pattern. The right side contains a list of instructions and their descriptions. The bottom section shows the current state of the emulator, including the number of cycles, the location of the next instruction, and the name of the checkerboard pattern.

INSTR	ADDR	DATA	DESCRIPTION		
PIX	04C	04D	000	704C4D00	MOV A B C
ADD	001	04C	04C	104C04C0	Copy C cells from A to B
ADD	001	04C	04C	104C04C0	PIX A B
JGR	002	04F	000	5E024F00	Draw color B in location A
FLP	04D	04E	000	754D4E01	JMP A
MOV	000	04F	000	01004F01	Jump to the instruction at A
ADD	001	050	000	13500150	JEQ A B C
JGR	010	050	000	5E105000	Jump to C if A = B
FLP	04D	04E	000	754D4E01	JGR A B C
MOV	000	050	000	01005001	Jump to C if A > B
JMP	000	000	000	00000000	JNE A B C
JMP	000	000	000	00000000	Jump to C if A != B
JMP	000	000	000	00000000	FLP A B C
JMP	000	000	000	00000000	Swap C cells between A and B
JMP	000	000	000	00000000	THR A
JMP	000	000	000	00000000	Start a new thread at A
JMP	000	000	000	00000000	ADD A B C
JMP	000	000	000	00000000	Set C = A + B
JMP	000	000	000	00000000	SUB A B C
JMP	000	000	000	00000000	Set C = A - B
JMP	000	000	000	00000000	MUL A B C
JMP	000	000	000	00000000	Set C = A * B
JMP	000	000	000	00000000	DIV A B C
JMP	000	000	000	00000000	Set C = A / B
JMP	000	000	000	00000000	MOD A B C
JMP	000	000	000	00000000	SET C = A % B
JMP	000	000	000	00000000	012 = Constant value 12
JMP	000	000	000	00000000	024 = Memory slot 24
JMP	000	000	000	00000000	*32 = Memory pointed by slot 32
JMP	000	000	000	00000000	Palette = 0123456789ABCDEF
JMP	000	000	000	00000000	box-256.com/manual
JMP	000	000	000	00000000	reddit.com/r/box256
JMP	000	000	000	00000000	(C) Juha Kiili

[STOP] [STEP] [PLAY] [PREV] TARGET [NEXT]
 CYCLES: 060E LOC: 0C CHECKERBOARD II
 -- LEVEL DONE --

- **Ejercicio 5**

Código

PIX @7C @7D 000
ADD 001 @7C @7C
ADD 001 @7E @7E
JGR 004 @7E @00
MOV 000 @7E 000
ADD 00C @7C @7C
ADD 001 @7F @7F
JGR 004 @7F @00
ADD 004 @7C @7C
MOV 000 @7F 000
ADD 001 @7D @7D
ADD @80 001 @80
JGR 004 @80 @00
MOV 000 @80 000
MOV 011 @7C 000
PIX @7C 000 000
ADD 001 @7C @7C
PIX @7C 000 000
ADD 00F @7C @7C
ADD 001 @7E @7E
JGR 002 @7E @3C
MOV 000 @7E 000
ADD 024 @7C @7C
JMP @3C 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 008 000 000
000 000 000 000
000 000 000 000

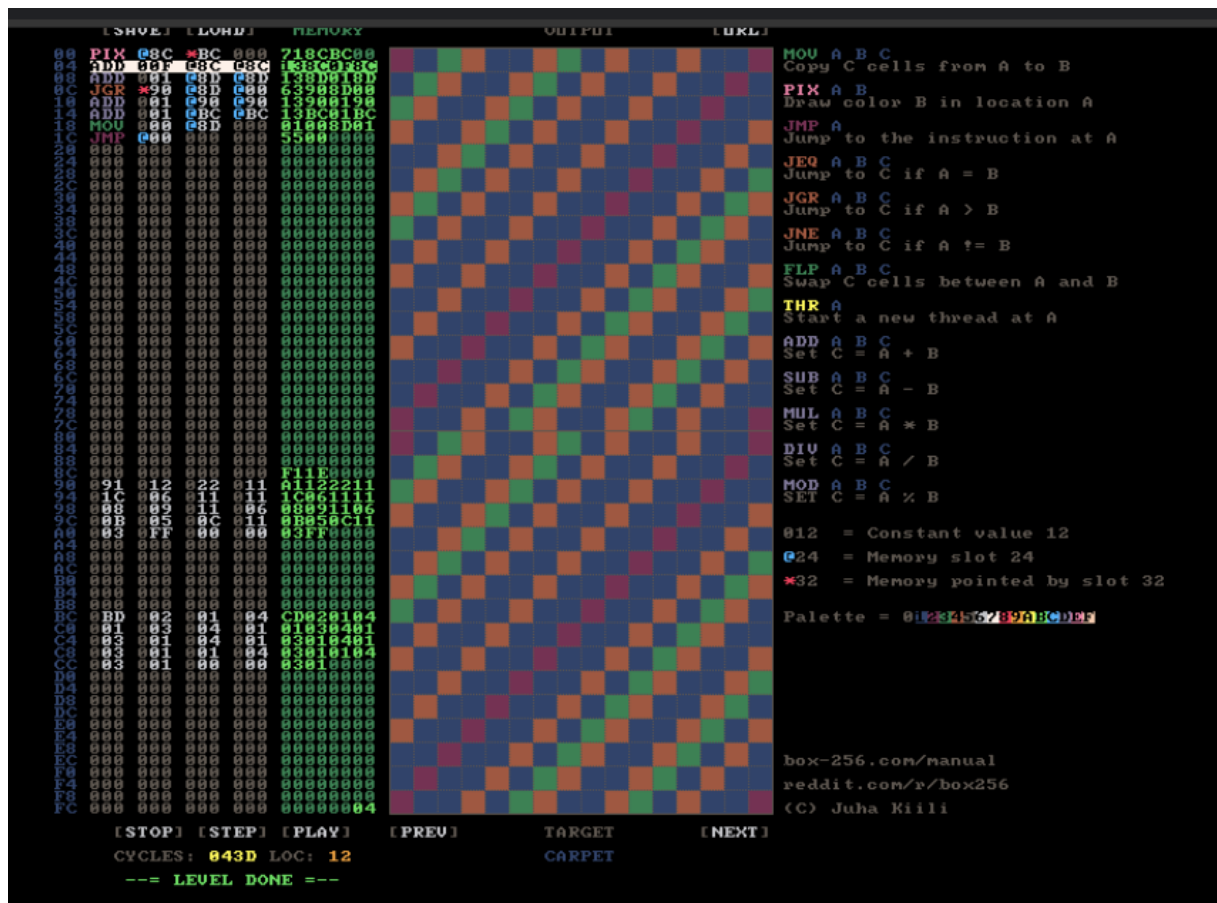
- **Ejercicio 6**

Código

PIX @8C @8D 000
ADD 001 @8C @8C
ADD 001 @8E @8E

```
JGR @80 @8E @00  
ADD 008 @8C @8C  
MOV 000 @8E 000  
ADD 001 @8F @8F  
JGR @80 @8F @00  
MOV 000 @8F 000  
ADD @80 @8C @8C  
DIV @80 002 @80  
ADD 001 @8D @8D  
ADD @80 @12 @12  
JMP @00 000 000  
  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
008 000 000 000  
000 000 000 000  
000 000 000 000  
000 008 000 000  
000 000 000 000  
000 000 000 000
```


000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
091 012 022 011
01C 006 011 011
008 009 011 006
00B 005 00C 011
003 0FF 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
0BD 002 001 004
001 003 004 001
003 001 004 001
003 001 001 004
003 001 000 000
000 000 000 000
000 000 000 000



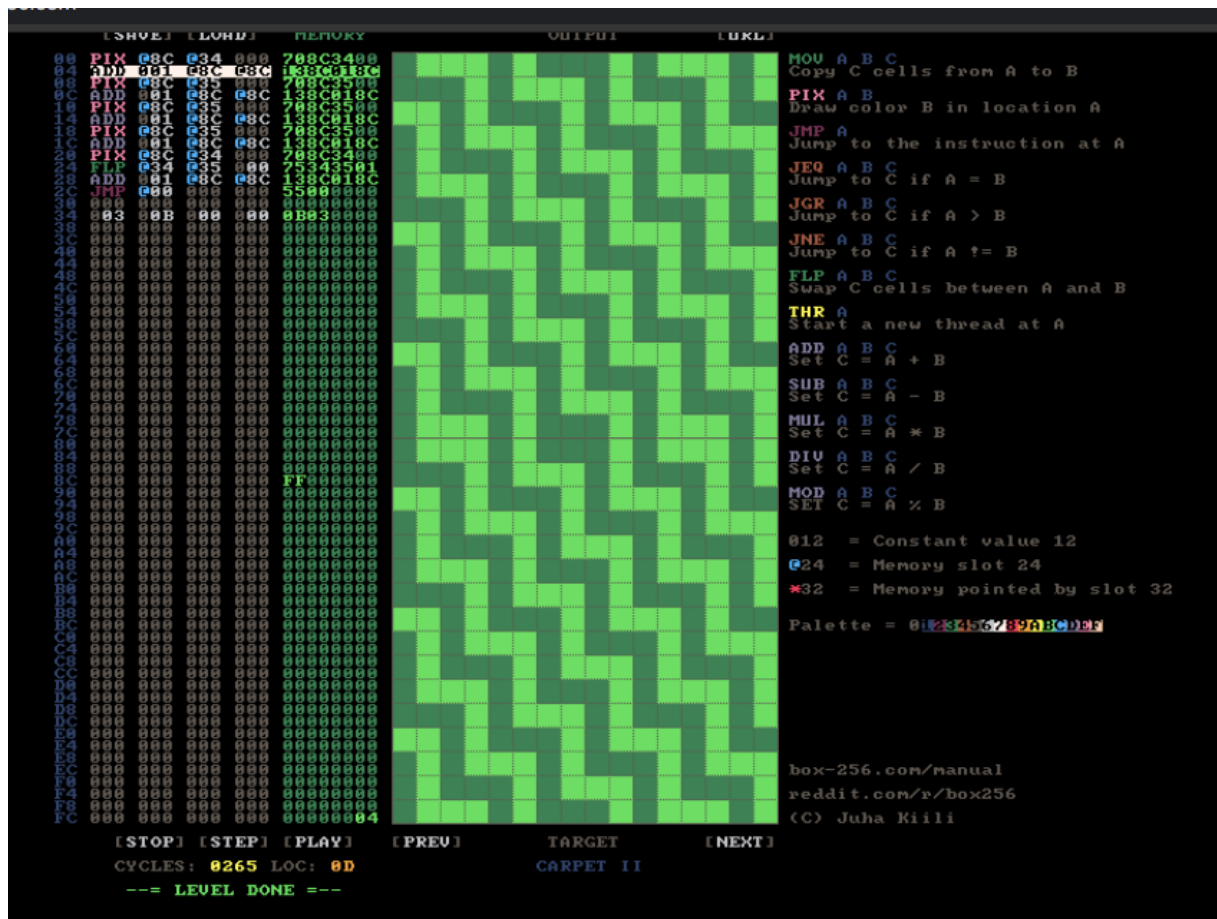
• Ejercicio 8

Código

```

PIX @8C @34 000
ADD 001 @8C @8C
PIX @8C @35 000
ADD 001 @8C @8C
PIX @8C @35 000
ADD 001 @8C @8C
PIX @8C @35 000
ADD 001 @8C @8C
JMP @00 000 000
000 000 000 000
003 00B 000 000
000 000 000 000
000 000 000 000

```



• Ejercicio 9

Código

```
PIX @8C 008 000
ADD *F0 @8C @8C
ADD 001 @8D @8D
JGR 003 @8D @00
MOV 000 @8D 000
ADD 001 @F0 @F0
ADD 001 @8E @8E
JGR 003 @8E @00
ADD *F4 @8C @8C
MOV 000 @8E 000
MOV 0F1 @F0 000
ADD 001 @F4 @F4
ADD 001 @8F @8F
JGR 003 @8F @00
MOV 000 @8F 000
MOV 0F5 @F4 000
MOV 080 @8C 000
ADD 001 @90 @90
JGR 002 @90 @00
```

```
MOV 000 @90 000  
MOV 088 @8C 000  
JMP @00 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
000 000 000 000  
0F1 011 -01 -10  
0F5 040 004 000  
000 000 000 000
```

000 000 000 000

```
[SHOE] [LOVE] MEMORY OUTPUT [LXLE]
00 PIX 03C 000 000 6F8C0800 MOV A B C
04 ADD 030 000 000 1A700800 Copy C cells from A to B
08 ADD 001 000 000 138D0100 PIX A B
0C JGR 003 000 000 5E038D00 Draw color B in location A
10 MOV 000 000 000 01008D01 JMP A
14 ADD 001 000 000 13790170 Jump to the instruction at A
18 ADD 001 000 000 138E018E JEQ A B C
1C JGR 003 000 000 5E038E00 Jump to C if A = B
20 ADD *F4 000 000 16F48C8C JGR A B C
24 MOV 000 000 000 01008E01 Jump to C if A > B
28 MOV 0F1 000 000 01F1F001 JNE A B C
2C ADD 001 0F4 0F4 13F401F4 Jump to C if A != B
30 ADD 001 00F 00F 138F018F FLP A B C
34 JGR 003 00F 000 5E038F00 Swap C cells between A and B
38 MOV 000 00F 000 01008F01 THR A
3C MOV 0F5 0F4 000 01F5F401 Start a new thread at A
40 MOV 000 00C 000 01508C01 ADD A B C
44 ADD 001 000 000 01508C01 Set C = A + B
48 JGR 002 000 000 5E029000 SUB A B C
4C MOV 000 000 000 01009001 Set C = A - B
50 MOV 000 00C 000 01508C01 MUL A B C
54 JMP 000 000 000 55000000 Set C = A * B
58 000 000 000 000 00000000 DIV A B C
5C 000 000 000 000 00000000 Set C = A / B
60 000 000 000 000 00000000 MOD A B C
64 000 000 000 000 00000000 SET C = A % B
68 000 000 000 000 00000000
6C 000 000 000 000 00000000
70 000 000 000 000 00000000
74 000 000 000 000 00000000
78 000 000 000 000 00000000
7C 000 000 000 000 00000000
80 000 000 000 000 00000000
84 000 000 000 000 00000000
88 000 000 000 000 00000000
8C 000 000 000 000 DC020202
90 000 000 000 000 00000000
94 000 000 000 000 00000000
98 000 000 000 000 00000000
A0 000 000 000 000 00000000
A4 000 000 000 000 00000000
A8 000 000 000 000 00000000
AC 000 000 000 000 00000000
B0 000 000 000 000 00000000
B4 000 000 000 000 00000000
B8 000 000 000 000 00000000
BC 000 000 000 000 00000000
C0 000 000 000 000 00000000
C4 000 000 000 000 00000000
C8 000 000 000 000 00000000
CC 000 000 000 000 00000000
D0 000 000 000 000 00000000
D4 000 000 000 000 00000000
D8 000 000 000 000 00000000
DC 000 000 000 000 00000000
E0 000 000 000 000 00000000
E4 000 000 000 000 00000000
E8 000 000 000 000 00000000
EC 000 000 000 000 00000000
F0 0F1 011 -01 -10 F311FF0
F4 0F5 400 004 00 F7400400
F8 000 000 000 000 00000000
FC 000 000 000 000 00000000

MOV A B C
Copy C cells from A to B
PIX A B
Draw color B in location A
JMP A
Jump to the instruction at A
JEQ A B C
Jump to C if A = B
JGR A B C
Jump to C if A > B
JNE A B C
Jump to C if A != B
FLP A B C
Swap C cells between A and B
THR A
Start a new thread at A
ADD A B C
Set C = A + B
SUB A B C
Set C = A - B
MUL A B C
Set C = A * B
DIV A B C
Set C = A / B
MOD A B C
SET C = A % B

012 = Constant value 12
024 = Memory slot 24
*32 = Memory pointed by slot 32

Palette = 0123456789ABCDEF

box-256.com/manual
reddit.com/r/box256
(C) Juha Kiili

[STOP] [STEP] [PLAY] [PREV] TARGET [NEXT]
CYCLES: 01E6 LOC: 18
--= LEVEL DONE ==
```

Ejercicio 10

Código

```
PIX @F0 001 000
ADD 010 @F0 @F0
ADD 001 @F1 @F1
JGR 010 @F1 @00
MOV 000 @F1 000
ADD @0D -01 @0D
MOV 000 @F0 000
ADD 011 @F0 @F0
ADD 011 @19 @19
ADD 001 @F2 @F2
JGR 010 @F2 @00
MOV 000 @F2 000
MOV *E8 @F0 000
PIX @F0 00C 000
ADD 001 @F0 @F0
ADD 001 @F1 @F1
JGR 007 @F1 @34
MOV 000 @F1 000
```

[illegible]

```
000 000 000 000
000 000 000 000
```



• Ejercicio 11

Código

```

PIX @F0 00A 000
ADD 001 @F0 @F0
JGR 0FF @F0 @00
MOV *B8 @F0 000
PIX @F0 000 000
ADD 001 @F0 @F0
ADD 001 @F1 @F1
JGR *B0 @F1 @10
ADD 001 @B0 @B0
ADD 001 @B8 @B8
MOV 000 @F1 000
ADD 001 @F2 @F2
JGR 006 @F2 @0C
ADD -01 @B0 @B0
MOV 0FF @22 000
MOV 000 @F2 000

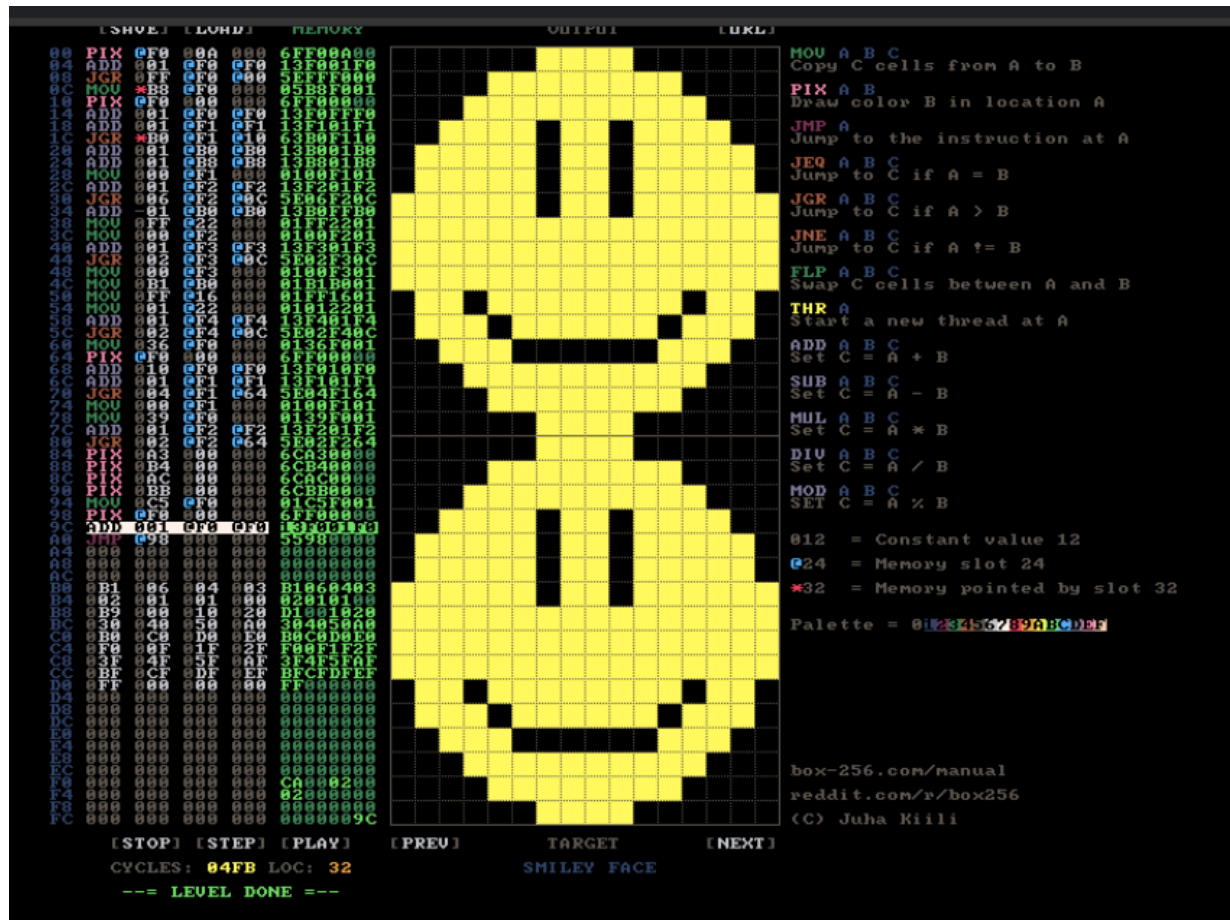
```

[illegible]


```

000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000

```



- Ejercicio 12

Código

```

MOV *DD @F8 000
PIX @F8 *AE 000
ADD 001 @F8 @F8
ADD 001 @F9 @F9
JGR *C2 @F9 @04
MOV 000 @F9 000
ADD 001 @C2 @C2
ADD 001 @DD @DD
ADD 001 @FA @FA
JGR *B8 @FA @00
MOV 000 @FA 000
ADD 001 @B8 @B8
ADD 001 @AE @AE
ADD 001 @FB @FB
JGR 009 @FB @00

```


PIX 035 004 000
PIX 033 004 000
PIX 043 004 000
PIX 054 004 000
PIX 04A 004 000
PIX 029 004 000
PIX 039 004 000
PIX 0A6 00A 000
PIX 0A9 00A 000
PIX 0A4 008 000
PIX 0AB 008 000
PIX 076 00C 000
PIX 086 00C 000
PIX 079 00C 000
PIX 089 00C 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 000 000
000 000 0AF 008
00F 008 00F 00C
004 000 00C 004
0B9 002 005 003
003 001 002 003
004 003 0C3 005
009 007 00A 00B
009 006 008 00A
00C 00C 00C 00C
008 00A 00C 002
004 004 008 006
006 004 002 003
004 0DE 005 014
024 033 043 054
065 074 083 092
0A2 0B2 0C2 0D4
0E3 0F2 0D7 0E6
0F6 0C4 0B5 0A5
096 045 024 059
000 000 000 000

PC	INSTR	OP	DATA	PC	INSTR	OP	DATA
00	MOV	*DD	05DDDF80	00	MOV	A B C	
04	PIX	*E8	71F8AE00	04	Copy	C cells from A to B	
08	ADD	*01	00000000	08	PIX	A B	
0C	ADD	*01	00000000	0C	Draw	color B in location A	
10	JGR	*C2	00000000	10	JMP	A	
14	MOV	*00	00000000	14	Jump	to the instruction at A	
18	ADD	*01	00000000	18	JEQ	A B C	
1C	ADD	*01	00000000	1C	Jump	to C if A = B	
20	ADD	*01	00000000	20	JGR	A B C	
24	ADD	*01	00000000	24	Jump	to C if A > B	
28	JGR	*B8	00000000	28	JNE	A B C	
2C	MOV	*00	00000000	2C	Jump	to C if A != B	
30	ADD	*01	00000000	30	FLP	A B C	
34	ADD	*01	00000000	34	Swap	C cells between A and B	
38	ADD	*01	00000000	38	THR	A	
3C	PIX	*35	00000000	3C	Start	a new thread at A	
40	PIX	*23	00000000	40	ADD	A B C	
44	PIX	*24	00000000	44	Set	C = A + B	
48	PIX	*54	00000000	48	SUB	A B C	
4C	PIX	*4A	00000000	4C	Set	C = A - B	
50	PIX	*29	00000000	50	MUL	A B C	
54	PIX	*06	00000000	54	Set	C = A * B	
58	PIX	*A9	00000000	58	DIU	A B C	
5C	PIX	*04	00000000	5C	Set	C = A / B	
60	PIX	*0B	00000000	60	MOD	A B C	
64	PIX	*76	00000000	64	SET	C = A % B	
68	PIX	*86	00000000	68	012	= Constant value 12	
6C	PIX	*77	00000000	6C	024	= Memory slot 24	
70	PIX	*89	00000000	70	*32	= Memory pointed by slot 32	
74	PIX	*89	00000000	74	Palette	= 0123456789ABCDEF	
78	000	000	000	78	box-256.com/manual		
7C	000	000	000	7C	reddit.com/r/box256		
80	000	000	000	80	(C) Juha Kiili		
84	000	000	000				
88	000	000	000				
8C	000	000	000				
90	000	000	000				
94	000	000	000				
98	000	000	000				
9C	000	000	000				
A0	000	000	000				
A4	000	000	000				
A8	000	000	000				
AC	00F	008	00F				
B0	004	000	00C				
B4	000	002	005				
B8	003	001	002				
BC	004	003	003				
C0	009	007	00A				
C4	009	006	008				
C8	00C	00C	00C				
CC	008	00A	00C				
D0	004	004	006				
D4	006	004	002				
D8	004	00C	014				
DC	024	033	054				
E0	065	074	083				
E4	065	074	083				
E8	065	074	083				
EC	065	074	083				
F0	065	074	083				
F4	065	074	083				
F8	065	074	083				
FC	065	074	083				

[STOP] [STEP] [PLAY] [PREV]

TARGET

[NEXT]

CYCLES: 03E8 LOC: 31

MARIO

--= LEVEL DONE ==