

# Nand2Tetris

CHETAN [NLN]

August 2024

## 1 Q0

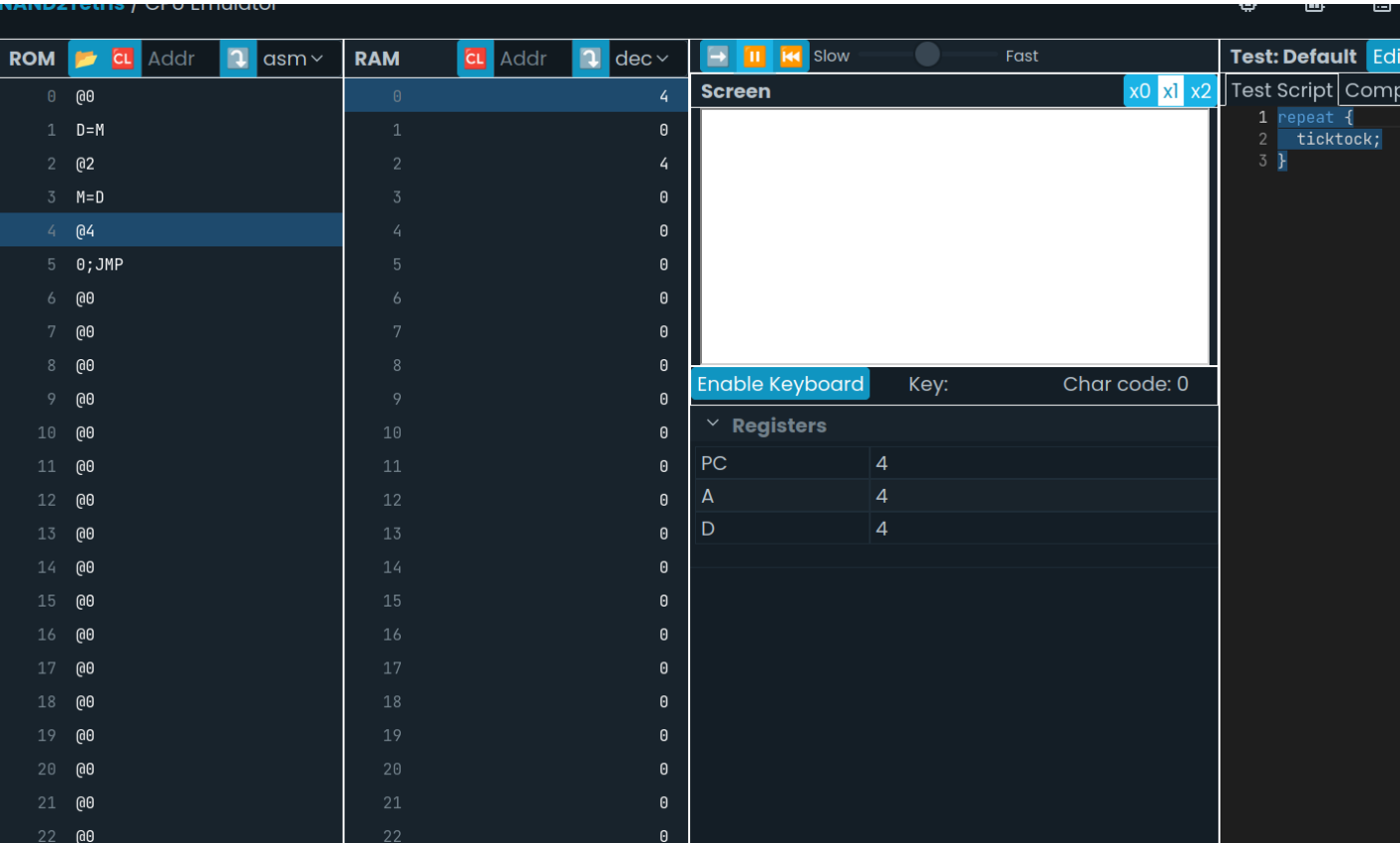


Figure 1: Output of Q0

### CODE FOR Q0

```
1 //Q0
2
3 @0
4 D=M
5 @2
6 M=D
7
8 (END)
9 @END
10 0; JMP
```

2 Q1

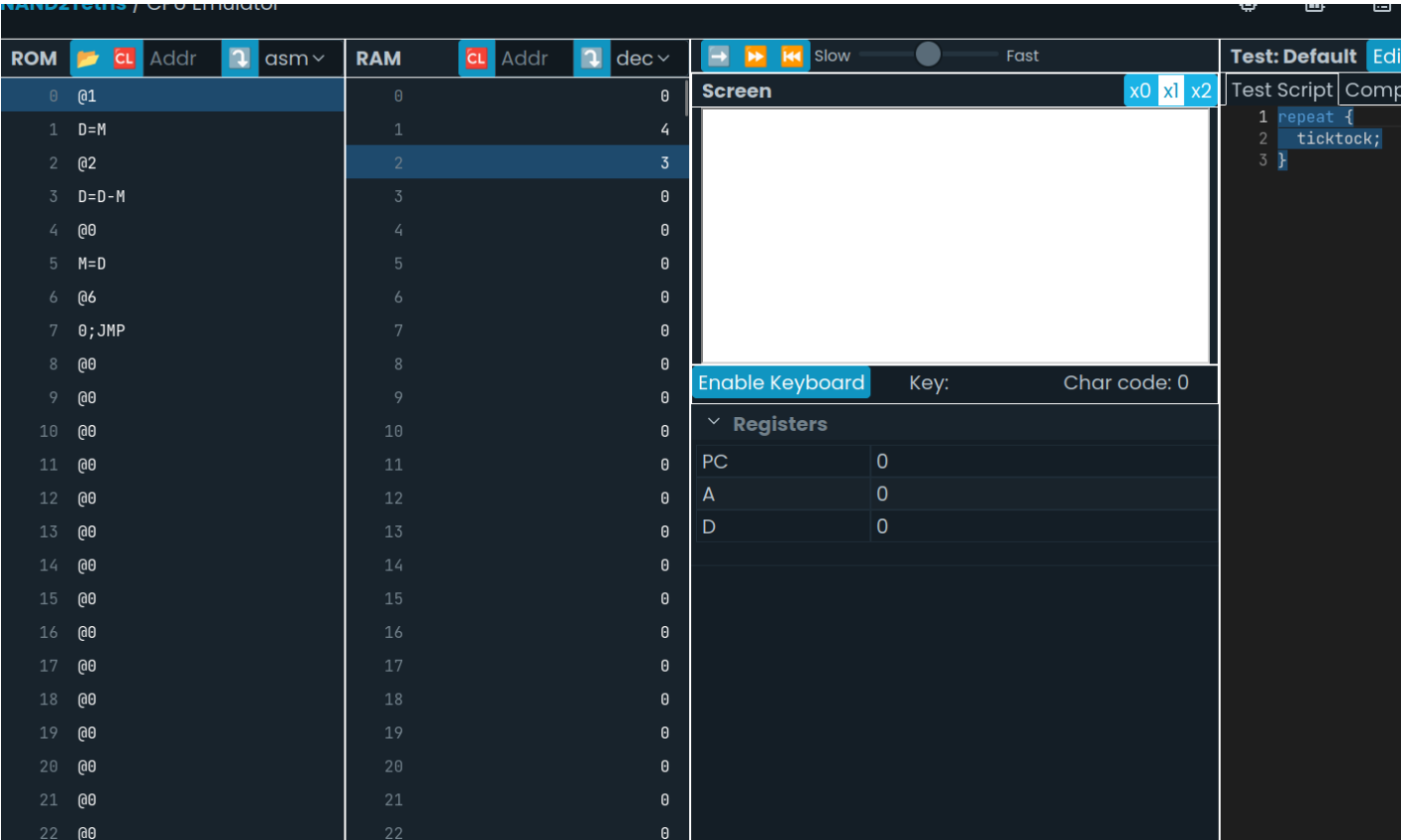


Figure 2: Output of Q1 before subtracting

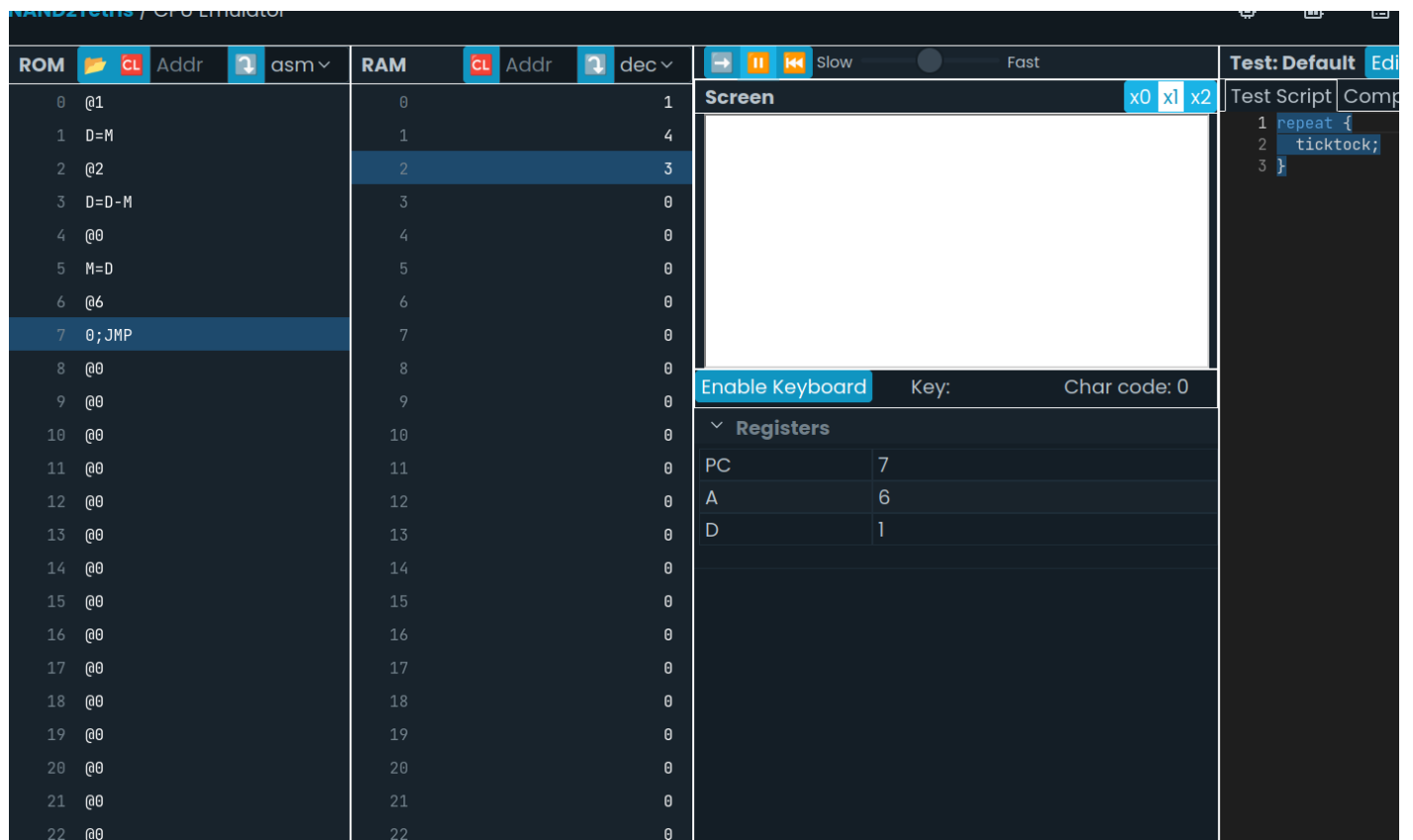


Figure 3: Output of Q1 after subtracting

## CODE FOR Q1

```
1 //Q1
2
3 @1
4 D=M
5 @2
6 D=D-M
7 @0
8 M=D
9 (END)
10 @END
11 0; JMP
```

3 Q2

ROM

CL

Addr

asm

RAM

CL

Addr

dec

Slow

Fast

Test: Default

Edi

0	@1	0	5
1	D=M	1	7
2	@2	2	0
3	M=D	3	0
4	@0	4	0
5	D=M	5	0
6	@1	6	0
7	M=D	7	0
8	@2	8	0
9	D=M	9	0
10	@0	10	0
11	M=D	11	0
12	@2	12	0
13	M=0	13	0
14	@14	14	0
15	0;JMP	15	0
16	@0	16	0
17	@0	17	0
18	@0	18	0
19	@0	19	0
20	@0	20	0
21	@0	21	0
22	@0	22	0

Screen

x0x1x2

Enable Keyboard

Key:

Char code: 0

Registers

PC

0

A

0

D

0

Test Script

Comp

1 repeat {

2 ticktock;

3 }

Figure 4: Output of Q2 before swapping

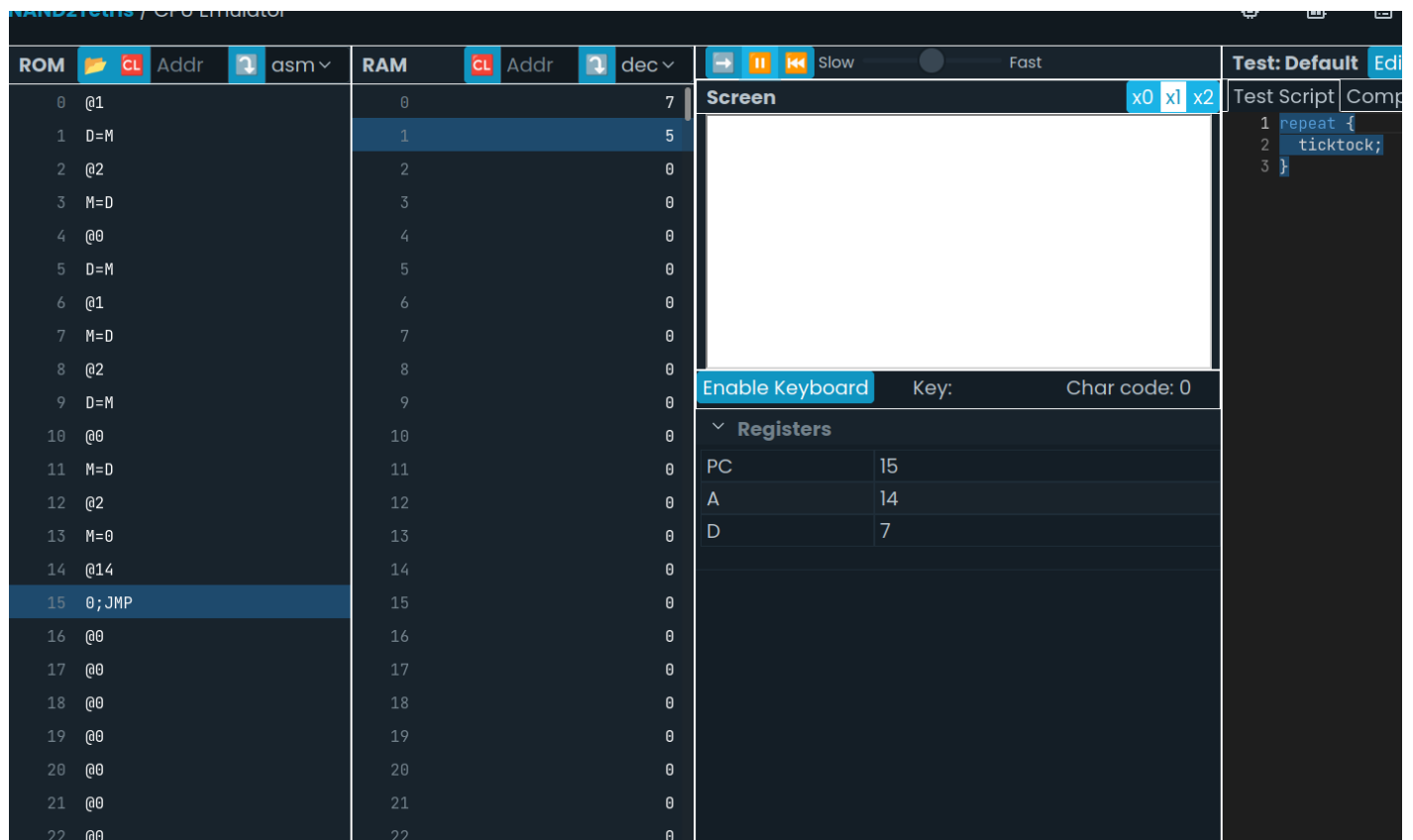


Figure 5: Output of Q2 after swapping

## CODE FOR Q2

```
1 //Q2
2
3 @1
4 D=M
5 @2
6 M=D
7 @0
8 D=M
9 @1
10 M=D
11 @2
12 D=M
13 @0
14 M=D
15 @2
16 M=0
17 (END)
18 @END
19 0; JMP
```

4 Q3

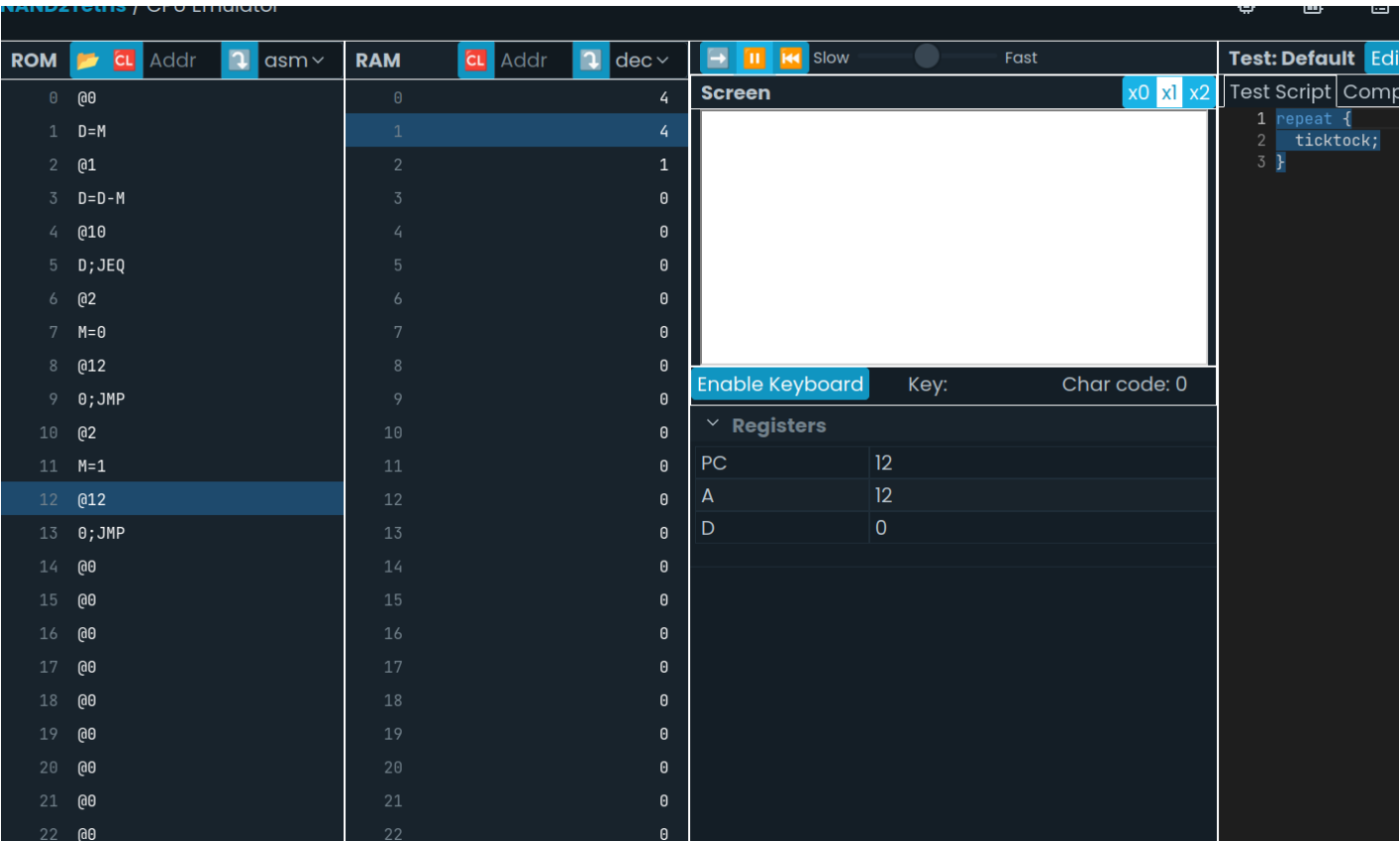


Figure 6: Output of Q3 if equal value



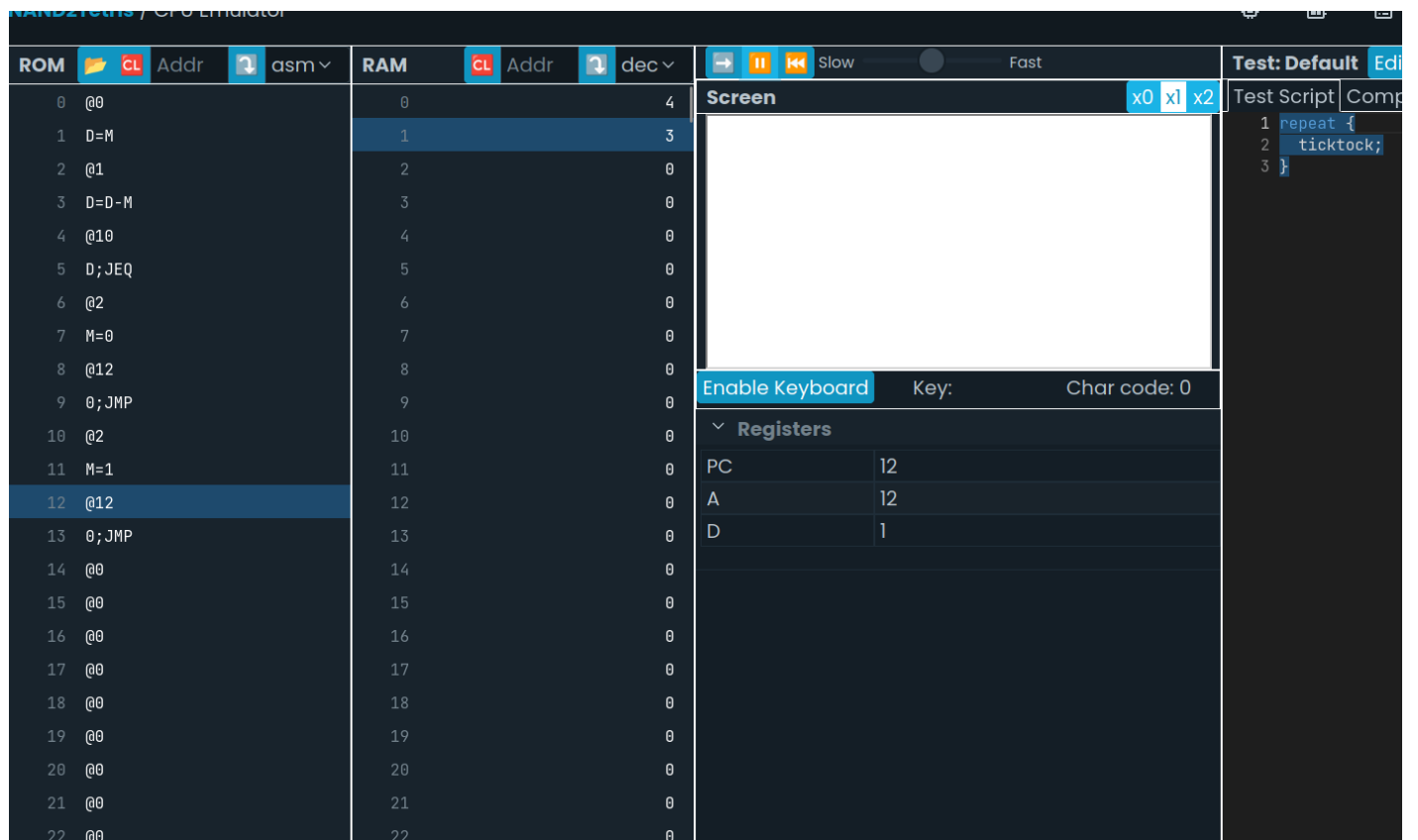


Figure 7: Output of Q3 if not equal

### CODE FOR Q3

```
1 // Q3
2
3 @0
4 D=M
5 @1
6 D=D-M
7 @TRUE
8 D;JEQ
9 @2
10 M=0
11 @END
12 O;JMP
13 (TRUE)
14 @2
15 M=1
16 (END)
17 @END
18 O;JMP
```

## 5 Q4

The screenshot displays the NAND2TTLs / CRO Emulator interface. It features a dark-themed layout with several panels:

- ROM Panel:** A list of memory addresses and their corresponding assembly instructions.
 

Addr	asm
0	@16
1	M=0
2	@16
3	D=M
4	@5
5	D=D-A
6	@14
7	D;JEQ
8	@0
9	M=M+1
10	@16
11	M=M+1
12	@2
13	0;JMP
14	@0
15	D=M
16	@1
17	M=D
18	@18
19	0;JMP
20	@0
21	@0
22	@0
- RAM Panel:** A list of memory addresses and their corresponding decimal values.
 

Addr	dec
0	5
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
- Screen Panel:** A large white rectangular area for visual output.
- Registers Panel:** A section titled "Registers" showing the current values of the PC, A, and D registers.
 

Register	Value
PC	0
A	0
D	0
- Test Script Panel:** A section titled "Test: Default" containing a script for testing the emulator.
 

```

1 repeat {
2   ticktock;
3 }
      
```

At the top of the interface, there are controls for execution speed (Slow to Fast) and a keyboard enable toggle.

Figure 8: Output of Q4 before additions

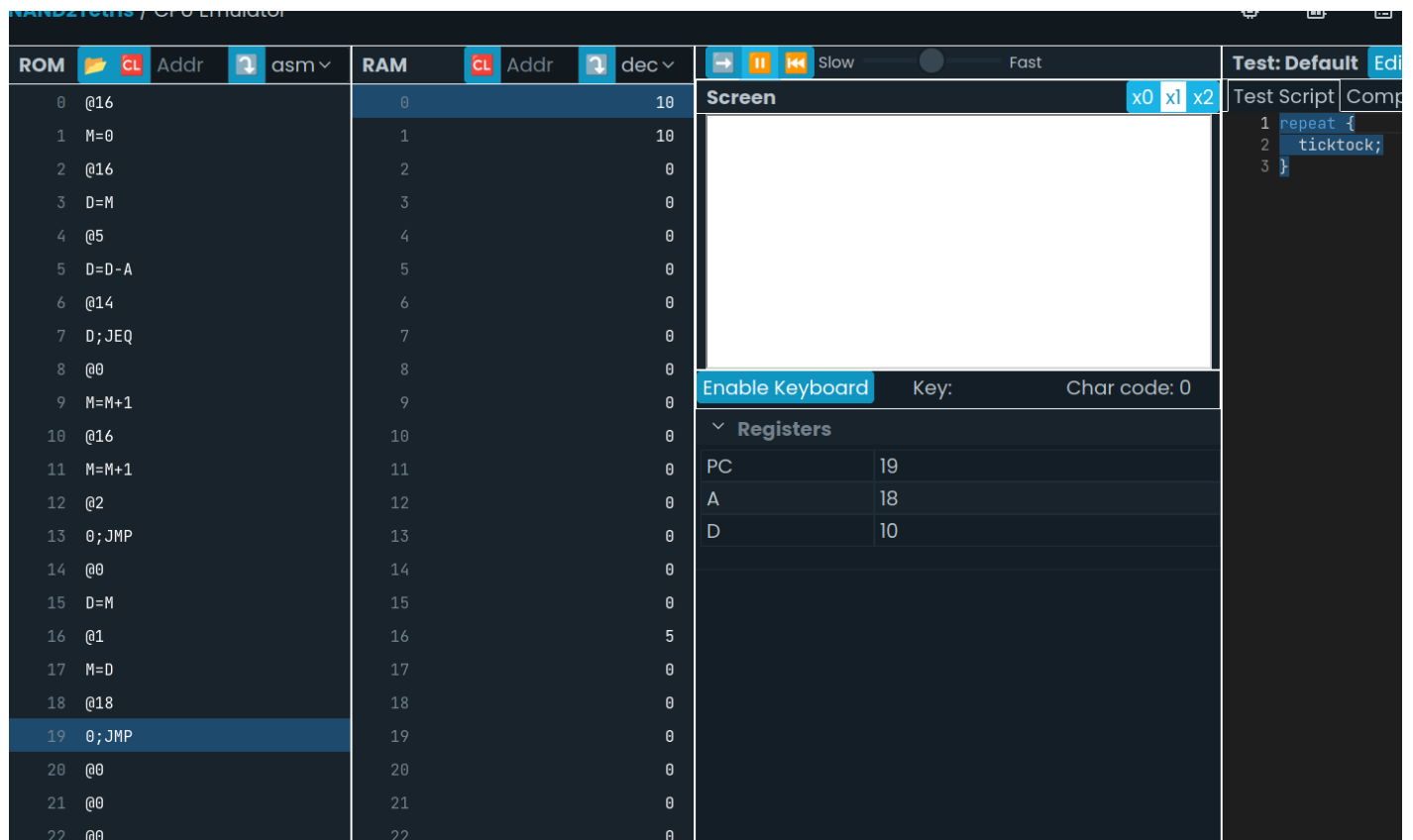


Figure 9: Output of Q4 after additions

## CODE FOR Q4

```
1 //Q4
2
3 @i
4 M=0
5 (LOOP)
6
7 @i
8 D=M
9 @5
10 D=D-A
11 @END
12 D;JEQ
13
14 @0
15 M=M+1
16
17 @i
18 M=M+1
19
20 @LOOP
21 O;JMP
22 (END)
23 @0
24 D=M
25 @1
26 M=D
27 (FINISH)
28 @FINISH
29 O;JMP
```

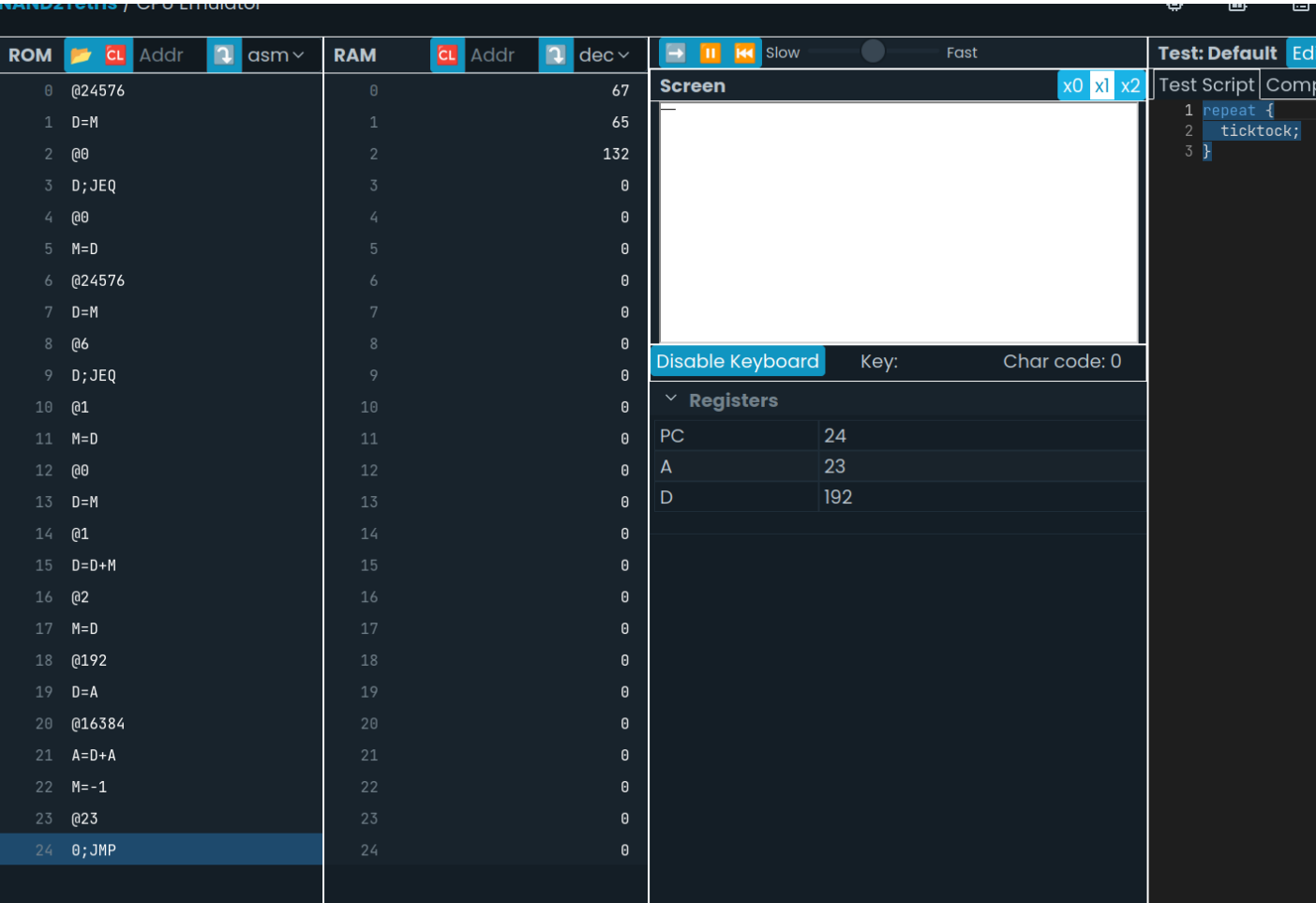


Figure 10: Output of Q5 sum of R0 and R1 and screen black some pixels

## CODE FOR Q5

```
1 // Q5
2 (LOOP1)
3 @KBD
4 D=M
5 @LOOP1
6 D;JEQ
7 @0
8 M=D
9
10
11 (LOOP2)
12 @KBD
13 D=M
14 @LOOP2
15 D;JEQ
16 @1
17 M=D
18 @0
19 D=M
20 @1
21 D=D+M
22 @2
23 M=D
24 @192
25 D=A
26 @SCREEN
27 A=D+A
28 M=-1
29 (END)
30 @END
31 O;JMP
```

## 7 Q6

The screenshot displays the Q6 emulator interface with the following components:

- ROM Panel:**

Addr	asm
0	@24576
1	D=M
2	@0
3	M=D
4	@13
5	D;JNE
6	@0
7	D=A
8	@16384
9	A=D+A
10	M=0
11	@0
12	0;JMP
13	@0
14	D=A
15	@16384
16	A=D+A
17	M=-1
18	@0
19	0;JMP
20	@0
21	@0
22	@0
23	@0
24	@0
- RAM Panel:**

Addr	dec
0	68
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
- Registers Panel:**

Register	Value
PC	15
A	0
D	0
- Screen Panel:** A large black rectangular area representing the screen output.
- Test Script Panel:**

```

1 repeat {
2   ticktock;
3 }

```
- Control Panel:** Includes buttons for 'Disable Keyboard', 'Key: D', 'Char code: 68', and a speed slider from 'Slow' to 'Fast'.

Figure 11: Output of Q6 screen black on keyboard input



## CODE FOR Q6

```
1 //Q6
2
3 (LOOP)
4 @KBD
5 D=M
6 @0
7 M=D
8 // CHECKING KBD INPUT
9 @TRUE
10 D;JNE
11 //CODE TO MAKE SCREEN WHITE
12 @0
13 D=A
14 @SCREEN
15 A=D+A
16 M=0
17 @LOOP
18 O;JMP
19 (TRUE)
20 //CODE TO MAKE SCREEN BLACK
21 @0
22 D=A
23 @SCREEN
24 A=D+A
25 M=-1
26 @LOOP
27 O;JMP
```

## 8 Q7

The screenshot displays the Nand2Tetris CPU Emulator interface. On the left, the ROM and RAM memory windows are shown. The ROM window lists instructions from address 0 to 24, with instruction 16 highlighted. The RAM window shows addresses 0 to 24, with address 1 highlighted. The central panel features a blank screen, an 'Enable Keyboard' button, and a 'Registers' window showing PC=16, A=16, and D=0. The right panel displays a test script with a 'repeat' loop.

ROM	Addr	asm	RAM	Addr	dec
0	@16		0		5
1	M=0		1		7
2	@16		2		35
3	D=M		3		0
4	@0		4		0
5	D=D-M		5		0
6	@16		6		0
7	D;JEQ		7		0
8	@1		8		0
9	D=M		9		0
10	@2		10		0
11	M=D+M		11		0
12	@16		12		0
13	M=M+1		13		0
14	@2		14		0
15	0;JMP		15		0
16	@16		16		5
17	0;JMP		17		0
18	@0		18		0
19	@0		19		0
20	@0		20		0
21	@0		21		0
22	@0		22		0
23	@0		23		0
24	@0		24		0

Registers:

Register	Value
PC	16
A	16
D	0

Test Script:

```

1 repeat {
2   ticktock;
3 }
  
```

Figure 12: Output of Q7 multiplication of R0 and R1

## CODE FOR Q7

```
1 //Q7
2
3 @i
4 M=0
5
6 (LOOP)
7
8 @i
9 D=M
10 @R0
11 D=D-M
12 @END
13 D; JEQ
14
15 @R1
16 D=M
17 @R2
18 M=D+M
19
20 @i
21 M=M+1
22
23 @LOOP
24 0; JMP
25
26 (END)
27 @END
28 0; JMP
```

## 9 Q8

The screenshot shows the Q8 emulator interface. The top bar includes a title bar, a toolbar with icons for ROM, RAM, and execution, and a speed slider. The main area is divided into four panels:

- ROM Panel:** Displays memory addresses and assembly instructions. The current instruction at address 17 is `@16`.
- RAM Panel:** Displays memory addresses and their corresponding values. The current value at address 17 is 0.
- Screen Panel:** A large white rectangle representing the screen. It is currently blank.
- Registers Panel:** Displays the current values of the PC, A, and D registers. PC is 17, A is 17, and D is 68.

The right sidebar contains a 'Test: Default' button and a 'Test Script' section with a list of instructions:

```

1 repeat {
2   ticktock;
3 }
  
```

Figure 13: Output of Q8 while screen is blackening

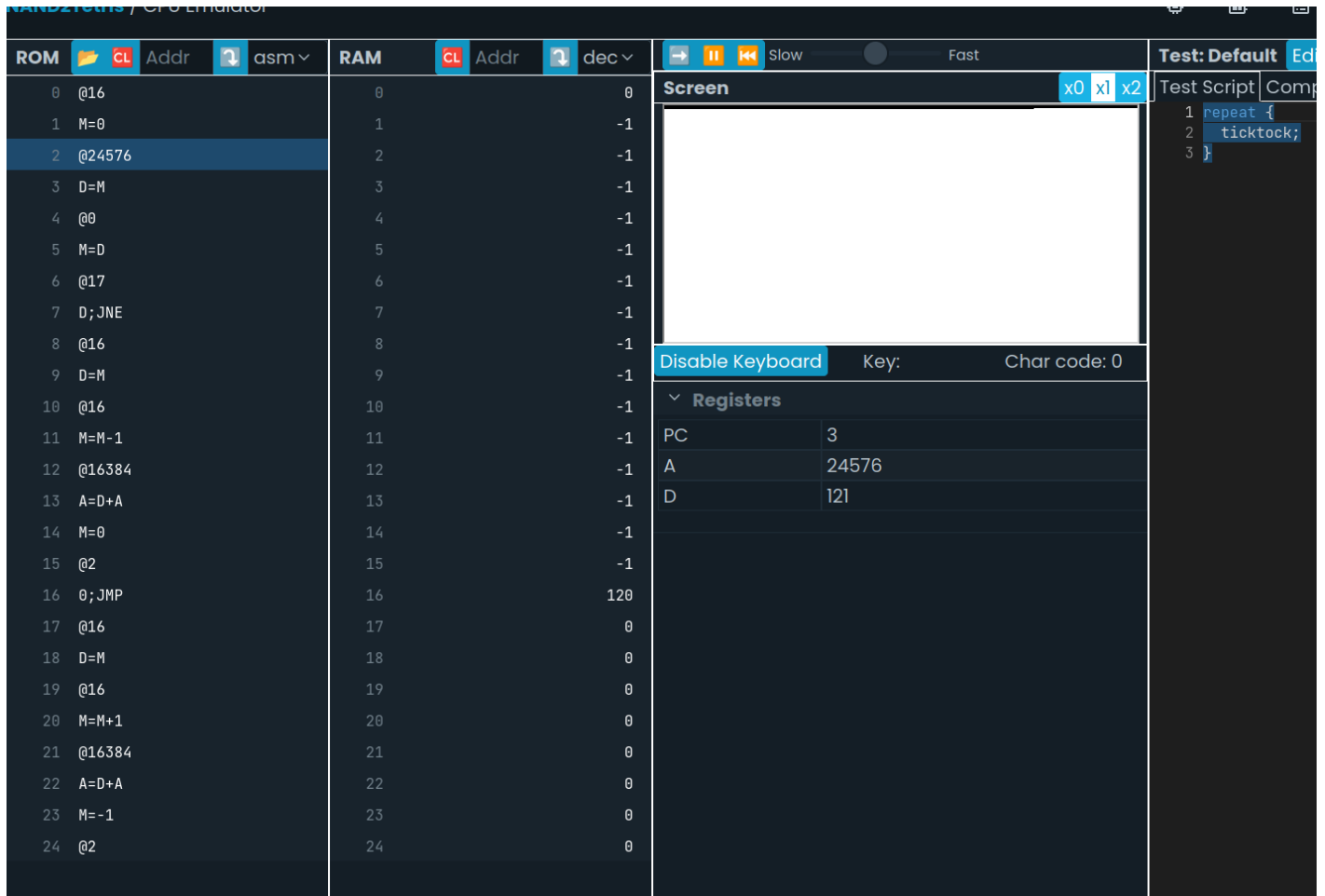


Figure 14: Output of Q8 while screen whitening

## CODE FOR Q8

```
1 // Q8
2
3 @i
4 M=0
5
6 (LOOP)
7
8 @KBD
9 D=M
10 @0
11 M=D
12 // CHECKING KBD INPUT
13 @TRUE
14 D;JNE
15 //CODE TO MAKE SCREEN WHITE
16 @i
17 D=M
18 @i
19 M=M-1
20 @SCREEN
21 A=D+A
22 M=0
23 @LOOP
24 O;JMP
25 (TRUE)
26 //CODE TO MAKE SCREEN BLACK
27 @i
28 D=M
29 @i
30 M=M+1
31 @SCREEN
32 A=D+A
33 M=-1
34 @LOOP
35 O;JMP
```