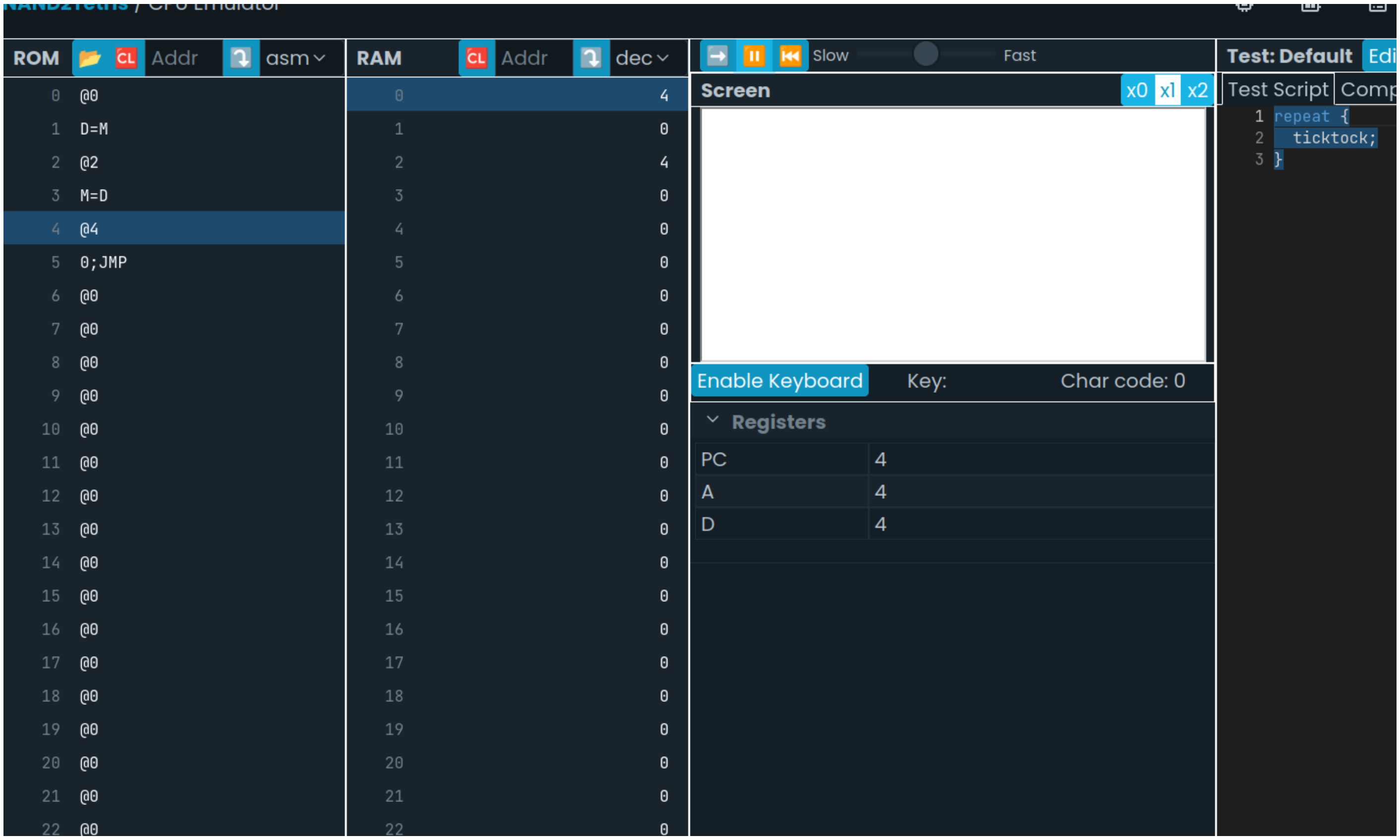
Nand2Tetris

CHETAN [NLN]

August 2024

1 Q0



ROM:

```
0  @0
1  D=M
2  @2
3  M=D
4  @4
5  0;JMP
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
```

RAM:

0	4
1	0
2	4
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

Registers:

PC	4
A	4
D	4

Test Script:

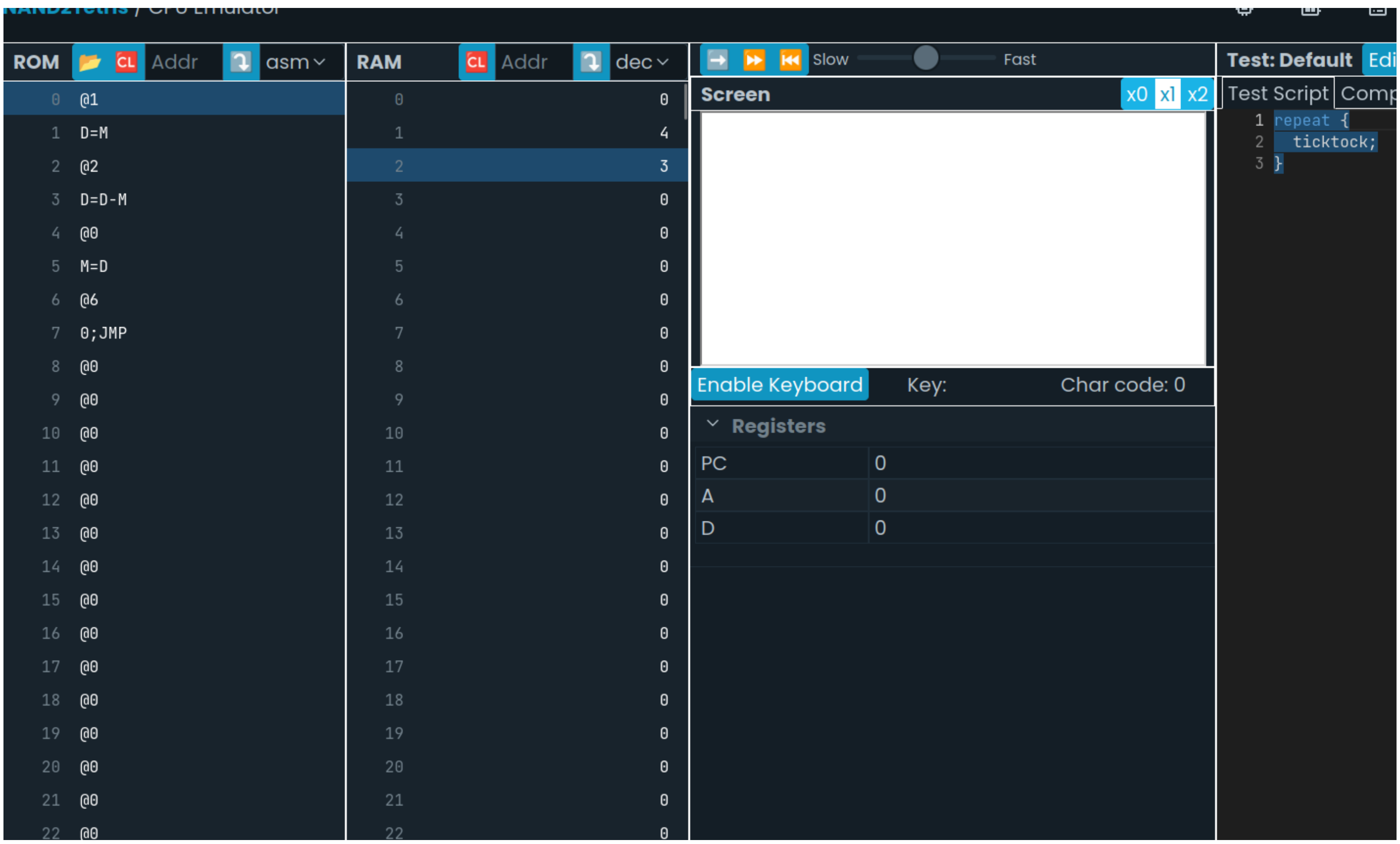
```
repeat {
  ticktock;
}
```

Figure 1: Output of Q0

CODE FOR Q0

```
//Q0

@0
D=M
@2
M=D

(END)
@END
0;JMP
```

2 Q1



ROM:

```
0   @1
1   D=M
2   @2
3   D=D-M
4   @0
5   M=D
6   @6
7   0;JMP
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
```

RAM:

```
0    0
1    4
2    3
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
```

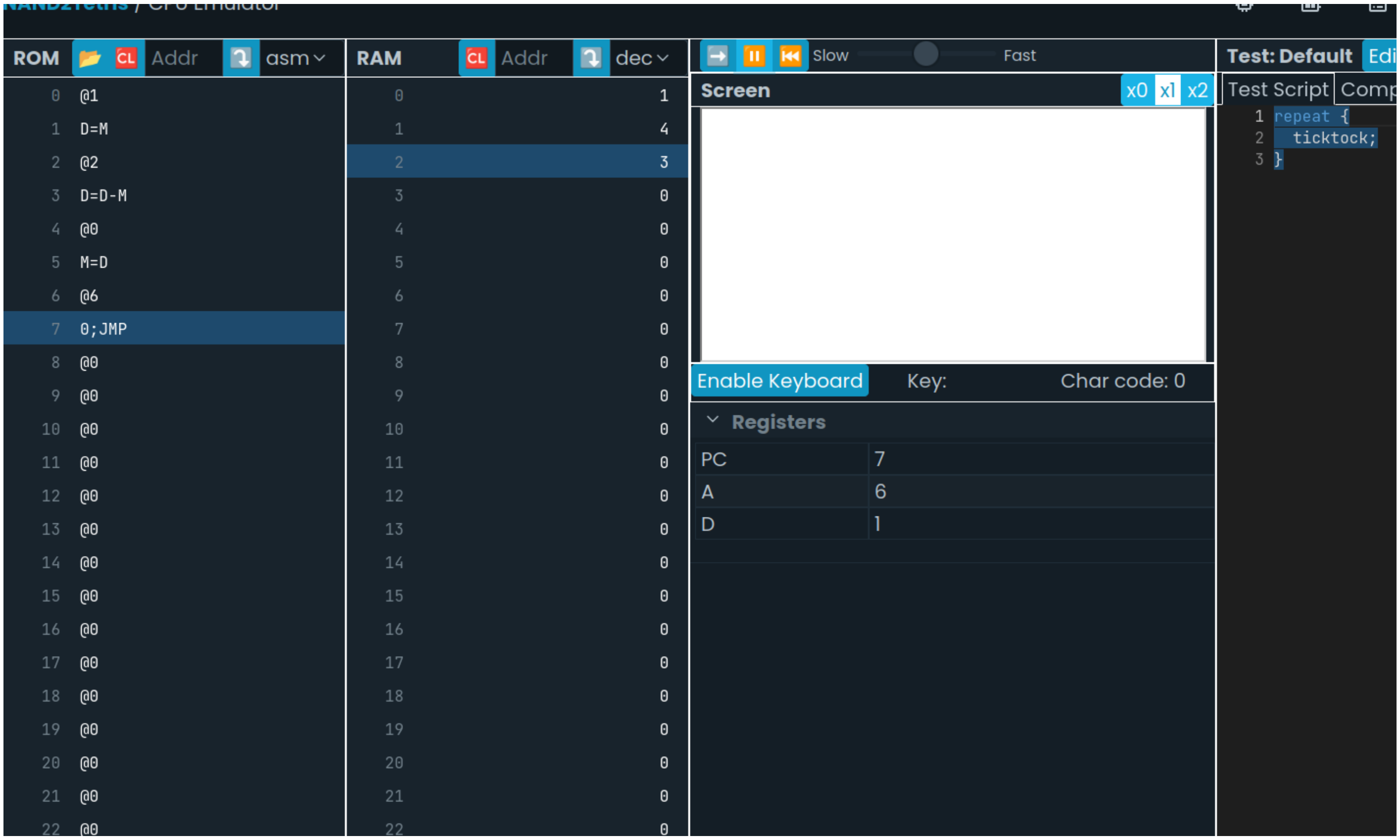
Registers:

PC	0
A	0
D	0

Test Script:

```
repeat {
  ticktock;
}
```

Figure 2: Output of Q1 before subtracting



```
0   @1
1   D=M
2   @2
3   D=D-M
4   @0
5   M=D
6   @6
7   0;JMP
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
```

RAM	
0	1
1	4
2	3
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

Registers	
PC	7
A	6
D	1

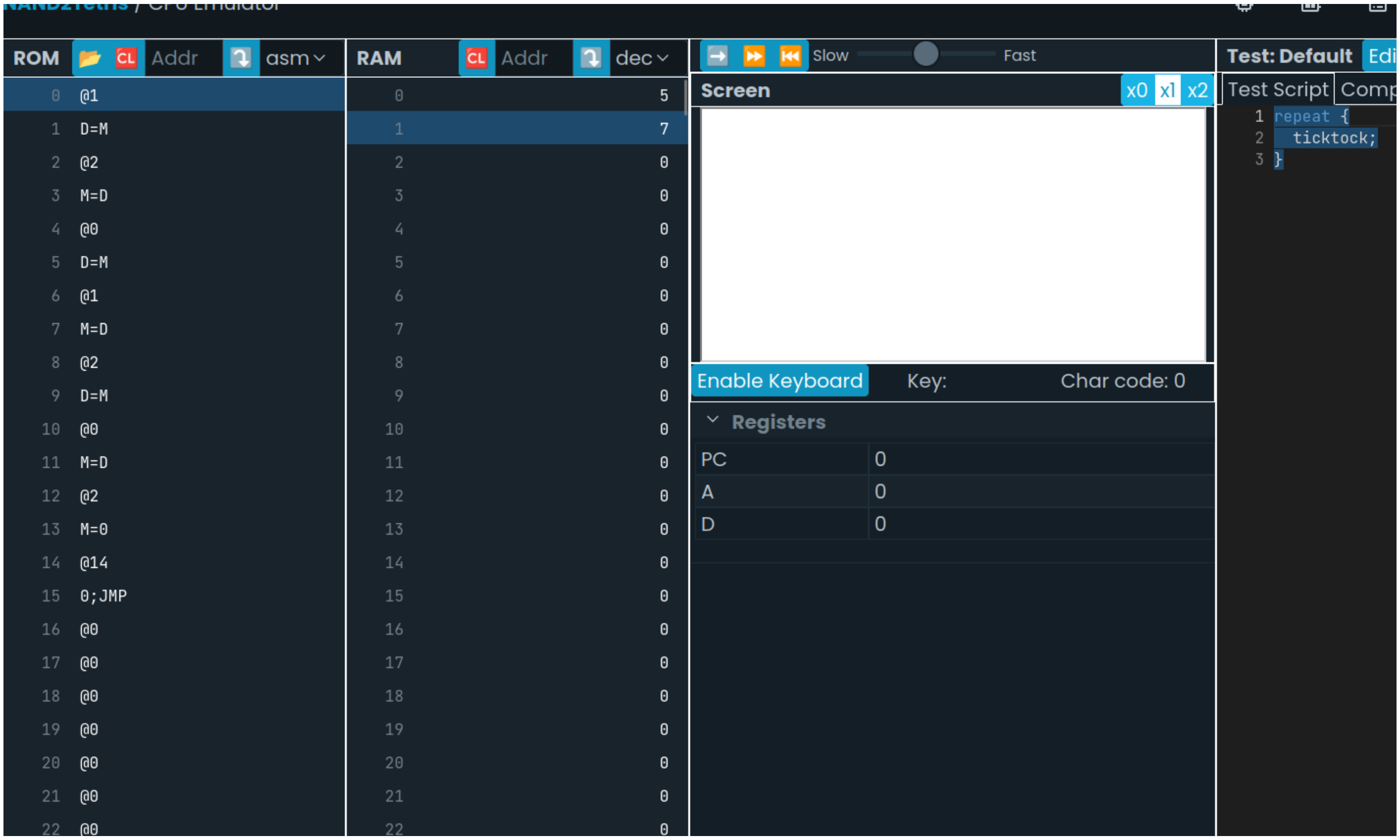
```
repeat {
  ticktock;
}
```

Figure 3: Output of Q1 after subtracting

CODE FOR Q1

```
//Q1

@1
D=M
@2
D=D-M
@0
M=D
(END)
@END
0;JMP
```

3 Q2



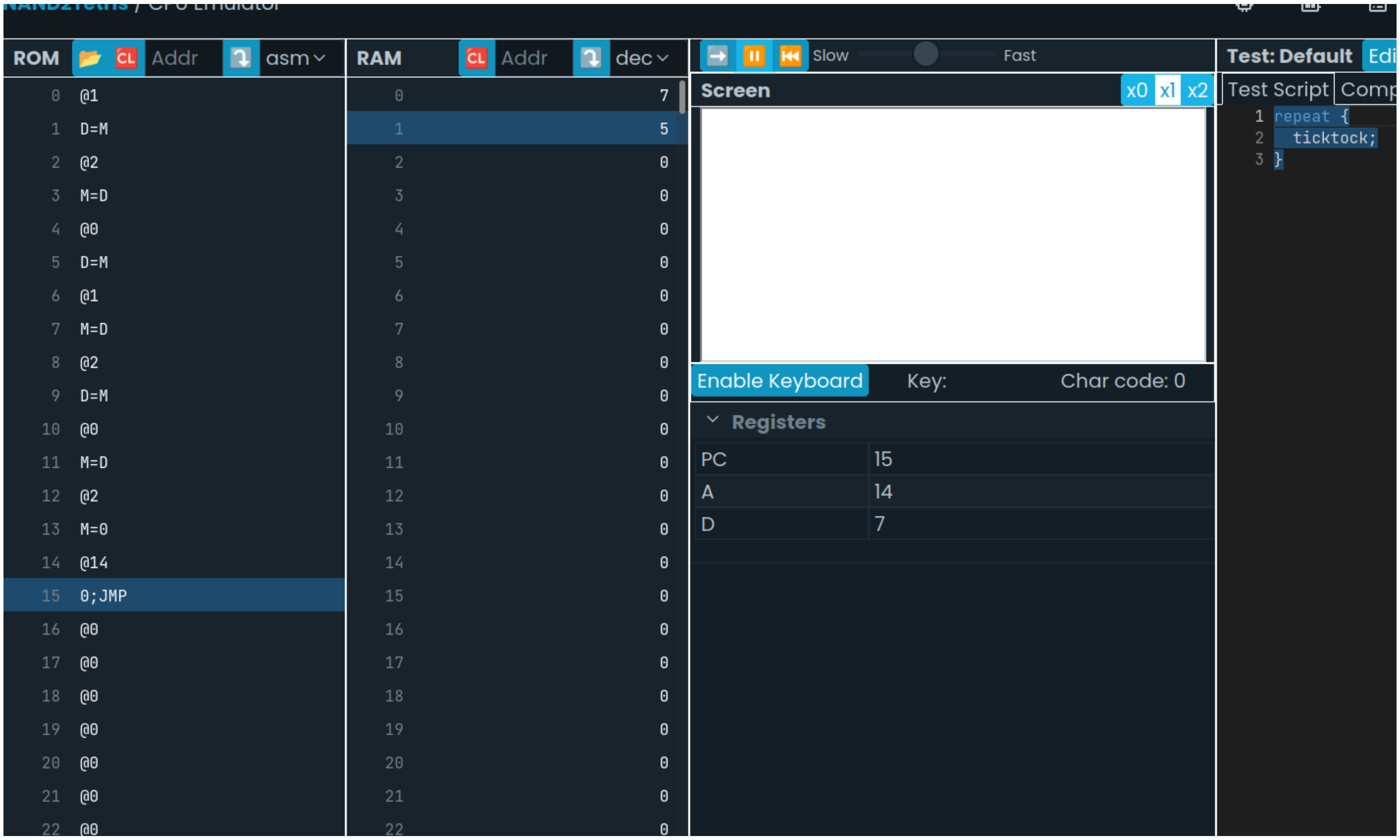
```
0   @1
1   D=M
2   @2
3   M=D
4   @0
5   D=M
6   @1
7   M=D
8   @2
9   D=M
10  @0
11  M=D
12  @2
13  M=0
14  @14
15  0;JMP
16  @0
17  @0
18  @0
19  @0
20  @0
21  @0
22  @0
```

```
0    5
1    7
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
```

PC	0
A	0
D	0

```
repeat {
  ticktock;
}
```

Figure 4: Output of Q2 before swapping



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

RAM	
0	7
1	5
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

Registers	
PC	15
A	14
D	7

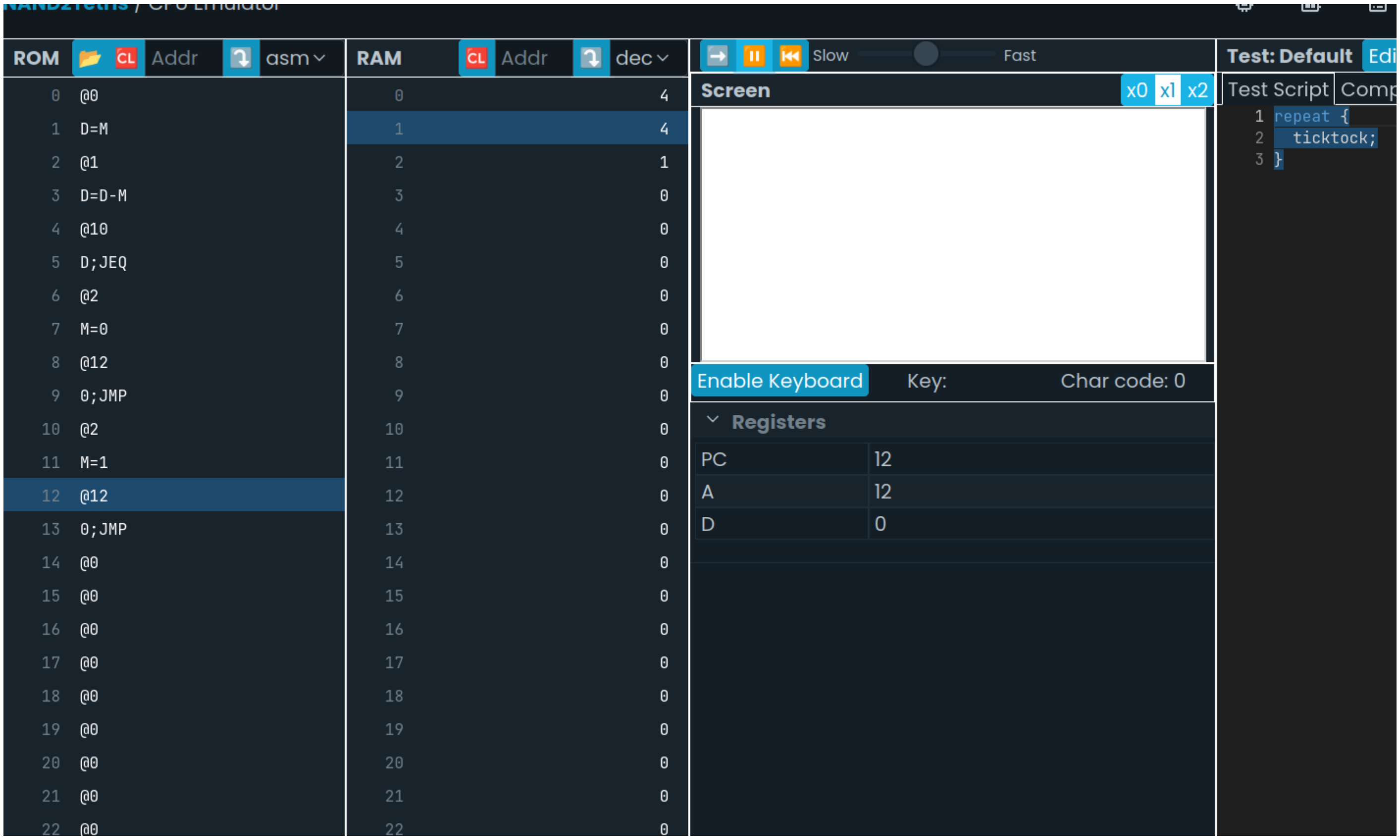
```
repeat {
  ticktock;
}
```

Figure 5: Output of Q2 after swapping

CODE FOR Q2

```
//Q2

@1
D=M
@2
M=D
@0
D=M
@1
M=D
@2
D=M
@0
M=D
@2
M=0
(END)
@END
0;JMP
```

4 Q3



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

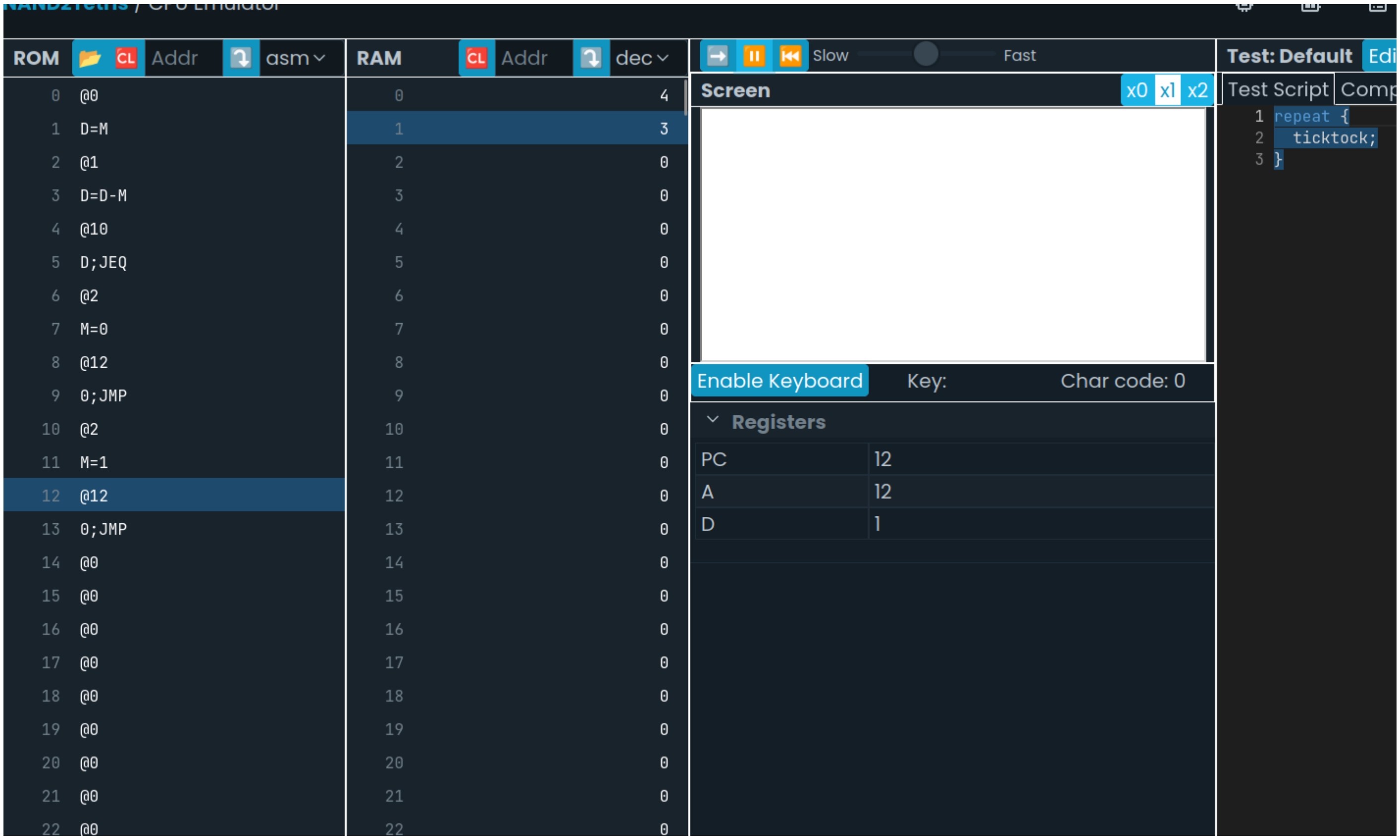
RAM	
0	4
1	4
2	1
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0

Char code: 0

Registers	
PC	12
A	12
D	0

```
repeat {
  ticktock;
}
```

Figure 6: Output of Q3 if equal value



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

```
RAM
 0   4
 1   3
 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
```

Registers	
PC	12
A	12
D	1

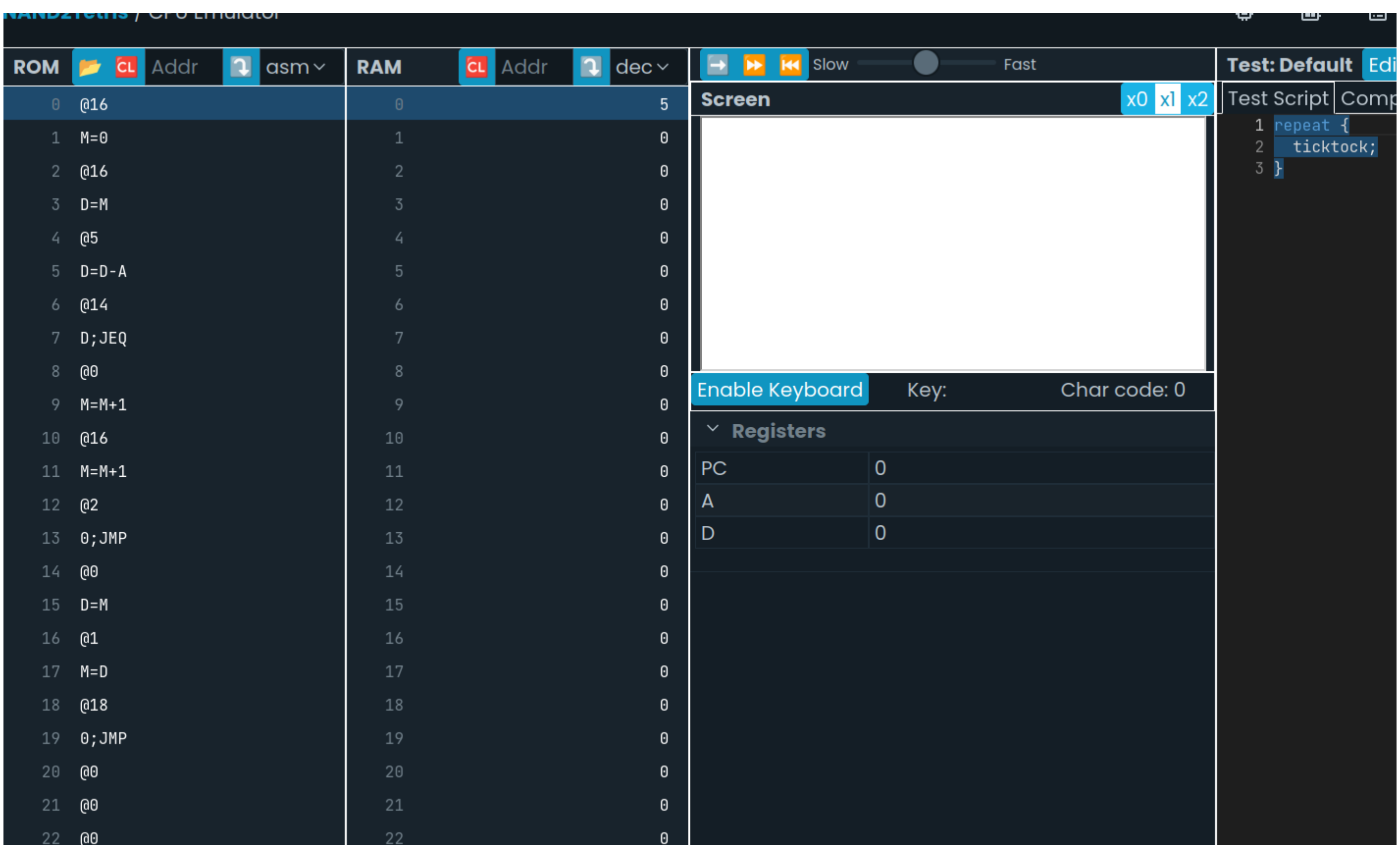
```
repeat {
  ticktock;
}
```

Figure 7: Output of Q3 if not equal

CODE FOR Q3

```
// Q3

@0
D=M
@1
D=D-M
@TRUE
D;JEQ
@2
M=0
@END
0;JMP
(TRUE)
@2
M=1
(END)
@END
0;JMP
```

5 Q4



```
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
```

```
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
```

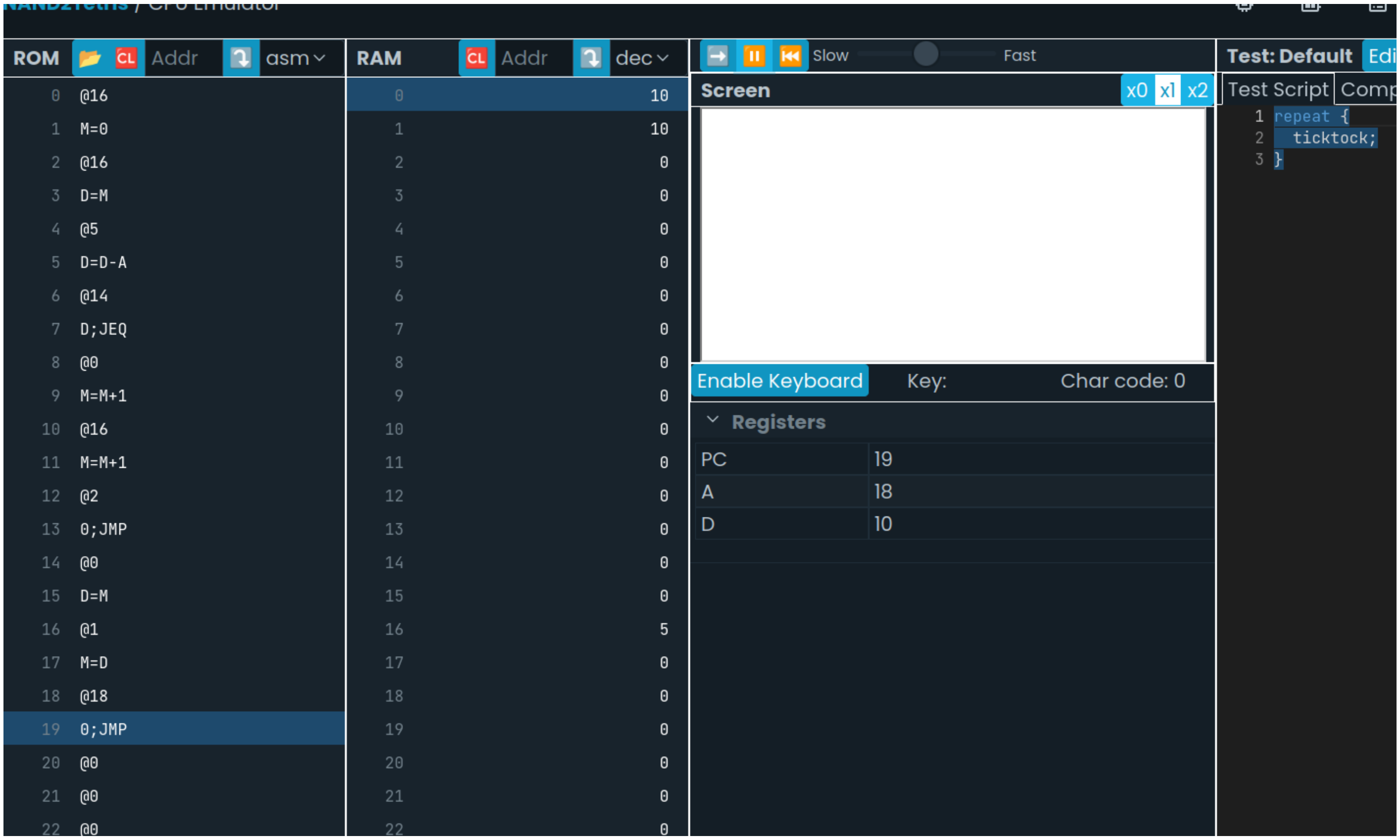
Key: Char code: 0

Registers

PC	0
A	0
D	0

```
repeat {
  ticktock;
}
```

Figure 8: Output of Q4 before additions



ROM:

```
 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:

```
 0  10
 1  10
 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   5
17   0
18   0
19   0
20   0
21   0
22   0
```

Registers:

```
PC  19
A   18
D   10
```

Test Script:

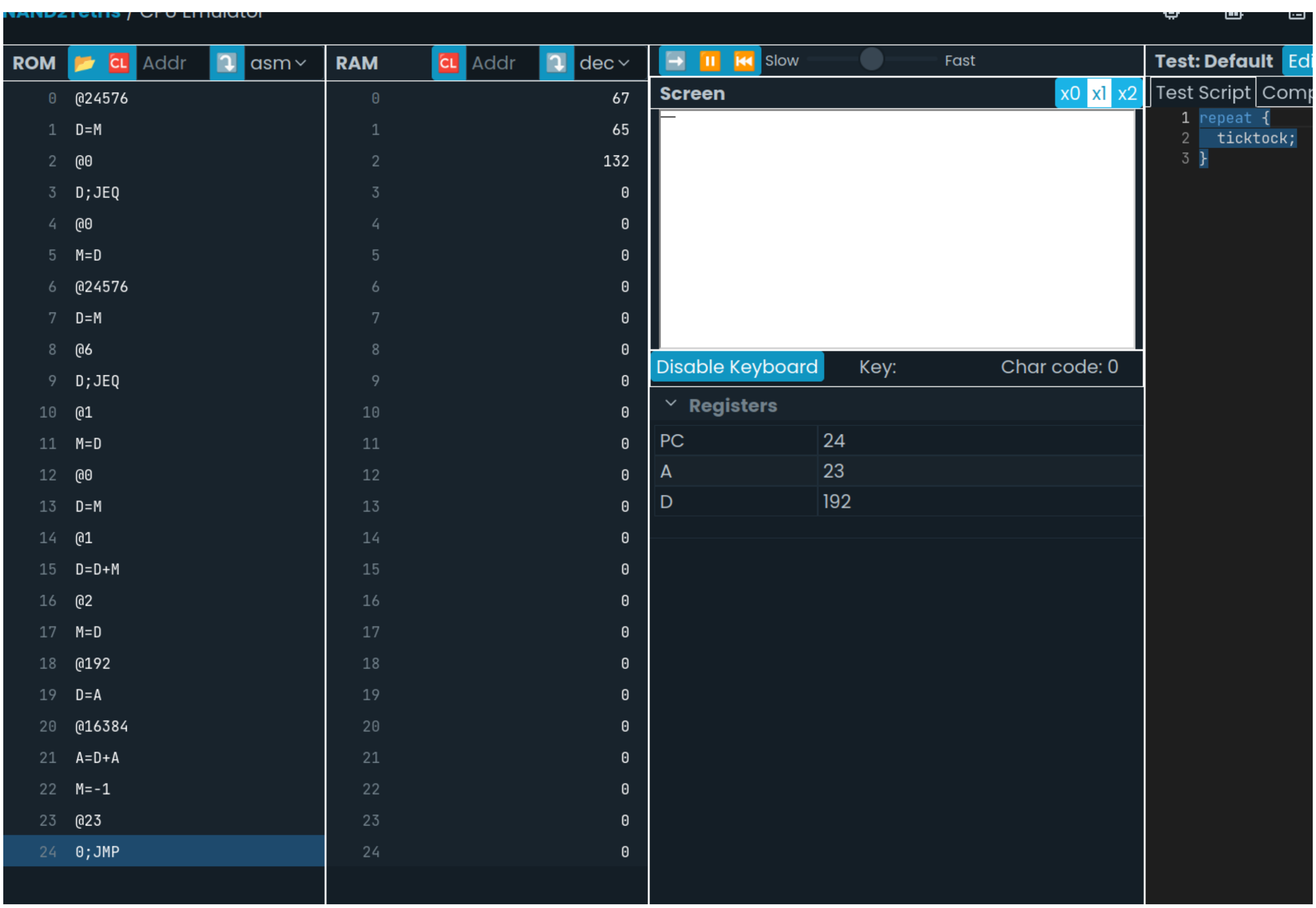
```
repeat {
  ticktock;
}
```

Figure 9: Output of Q4 after additions

CODE FOR Q4

```
//Q4

@i
M=0
(LOOP)

@i
D=M
@5
D=D-A
@END
D;JEQ

@0
M=M+1

@i
M=M+1

@LOOP
0;JMP
(END)
@0
D=M
@1
M=D
(FINISH)
@FINISH
0;JMP
```

6 Q5



```
0   @24576
1   D=M
2   @0
3   D;JEQ
4   @0
5   M=D
6   @24576
7   D=M
8   @6
9   D;JEQ
10  @1
11  M=D
12  @0
13  D=M
14  @1
15  D=D+M
16  @2
17  M=D
18  @192
19  D=A
20  @16384
21  A=D+A
22  M=-1
23  @23
24  0;JMP
```

0	67
1	65
2	132
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

Key:

Char code: 0

PC	24
A	23
D	192

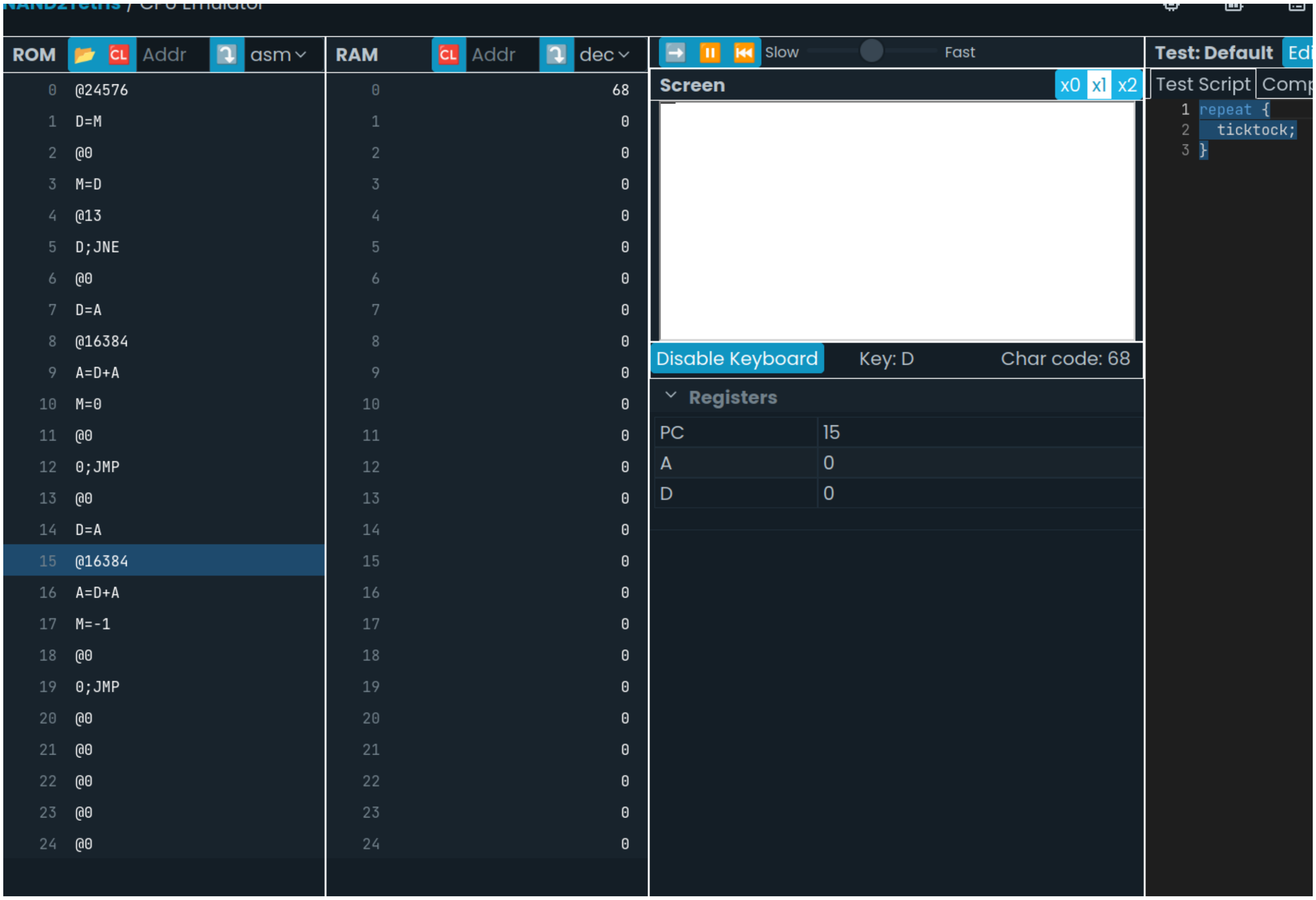
```
repeat {
  ticktock;
}
```

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

CODE FOR Q5

```
// Q5
(LOOP1)
@KBD
D=M
@LOOP1
D;JEQ
@0
M=D


(LOOP2)
@KBD
D=M
@LOOP2
D;JEQ
@1
M=D
@0
D=M
@1
D=D+M
@2
M=D
@192
D=A
@SCREEN
A=D+A
M=-1
(END)
@END
0;JMP
```

7 Q6



```
 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	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

Key: D Char code: 68

PC	15
A	0
D	0

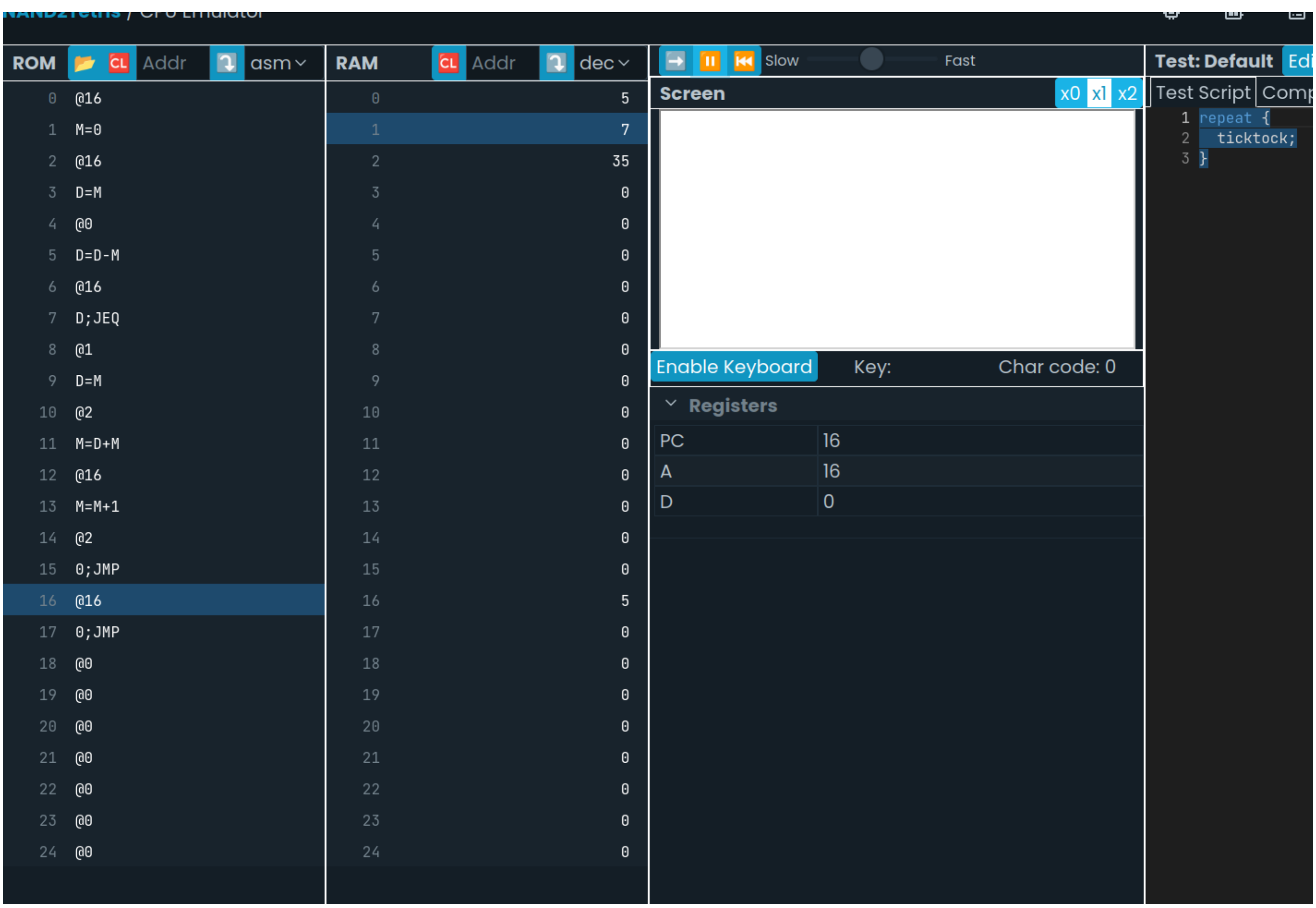
```
repeat {
  ticktock;
}
```

Figure 11: Output of Q6 screen black on keyboard input

CODE FOR Q6

```
//Q6

(LOOP)
@KBD
D=M
@0
M=D
// CHECKING KBD INPUT
@TRUE
D;JNE
//CODE TO MAKE SCREEN WHITE
@0
D=A
@SCREEN
A=D+A
M=0
@LOOP
0;JMP
(TRUE)
//CODE TO MAKE SCREEN BLACK
@0
D=A
@SCREEN
A=D+A
M=-1
@LOOP
0;JMP
```

8 Q7



ROM:

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

RAM:

```
 0     5
 1     7
 2    35
 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     5
17     0
18     0
19     0
20     0
21     0
22     0
23     0
24     0
```

Registers:

```
PC   16
A    16
D    0
```

Test Script:

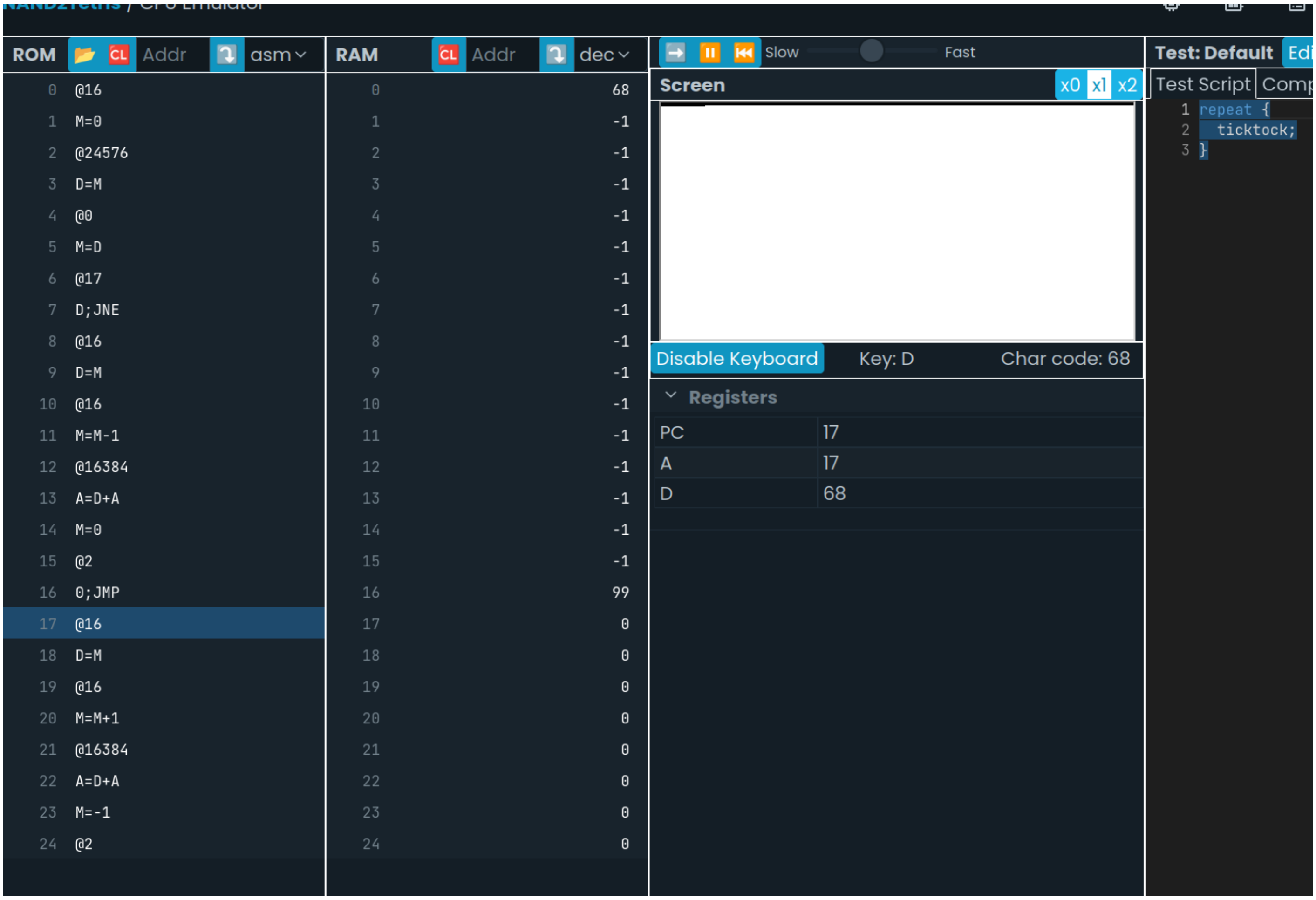
```
repeat {
  ticktock;
}
```

Figure 12: Output of Q7 multiplication of R0 and R1

CODE FOR Q7

```
//Q7

@i
M=0

(LOOP)

@i
D=M
@R0
D=D-M
@END
D;JEQ

@R1
D=M
@R2
M=D+M

@i
M=M+1

@LOOP
0;JMP

(END)
@END
0;JMP
```

9 Q8



ROM:

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

RAM:

0	68
1	-1
2	-1
3	-1
4	-1
5	-1
6	-1
7	-1
8	-1
9	-1
10	-1
11	-1
12	-1
13	-1
14	-1
15	-1
16	99
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0

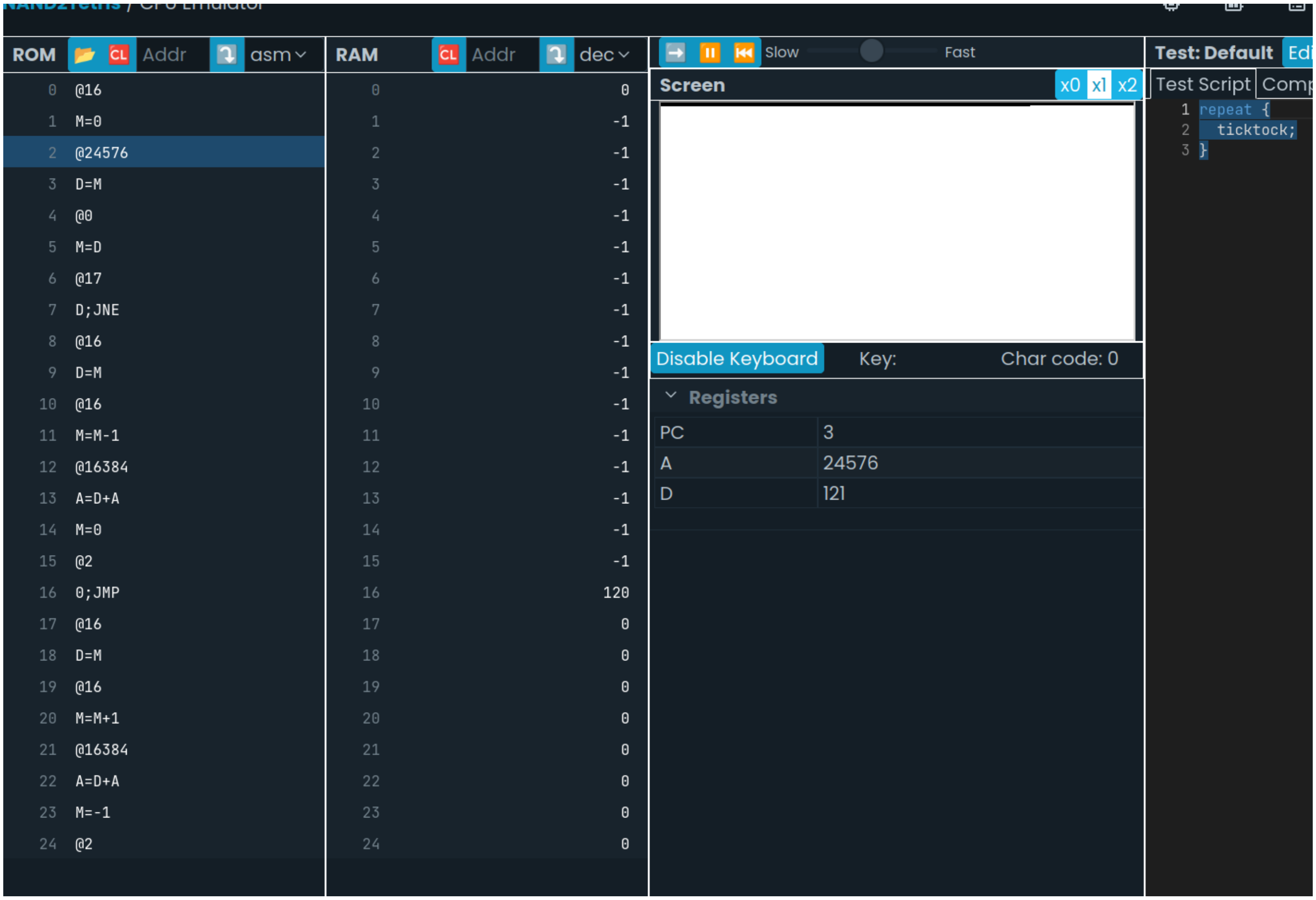
Key: D

Char code: 68

PC	17
A	17
D	68

```
repeat {
  ticktock;
}
```

Figure 13: Output of Q8 while screen is blackening



ROM:

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

RAM:

```
 0     0
 1    -1
 2    -1
 3    -1
 4    -1
 5    -1
 6    -1
 7    -1
 8    -1
 9    -1
10    -1
11    -1
12    -1
13    -1
14    -1
15    -1
16   120
17     0
18     0
19     0
20     0
21     0
22     0
23     0
24     0
```

Key:

Char code: 0

Registers

PC	3
A	24576
D	121

Test Script:

```
repeat {
  ticktock;
}
```

Figure 14: Output of Q8 while screen whitening

CODE FOR Q8

```
// Q8

@i
M=0

(LOOP)

@KBD
D=M
@0
M=D
// CHECKING KBD INPUT
@TRUE
D;JNE
//CODE TO MAKE SCREEN WHITE
@i
D=M
@i
M=M-1
@SCREEN
A=D+A
M=0
@LOOP
0;JMP
(TRUE)
//CODE TO MAKE SCREEN BLACK
@i
D=M
@i
M=M+1
@SCREEN
A=D+A
M=-1
@LOOP
0;JMP
```