Assignment 2

divide.bin photo

```
0011000000000000 ; Start at x3000
1010 011 000001110 ; Loading R3 with the address that holds x4000 (Current Address x3000)
1010 100 000001110 ; Loading R4 with the address that holds x4001 (Current Address x3001)
0101 110 110 1 00000 ; Assigning 0 to R6 (Current Address x3002)
1001 100 100 111111; Not R4 to be used in the subtraction (Current Address x3003)
0001 100 100 1 00001; R4 2's Complement needed for the subtraction (Current Address x3004)
0001 011 011 0 00 100 ; R3 <- R3-R4 (now R3 + (-R4)) (Current Address x3005)
0000 100 000000010; Break to address x3009 if R3 < 0 (negative) (Current Address x3006)
0001 110 110 1 00001 ; R6 <- R6+1 (Current Address x3007)
0000 001 111111100 ; Break to Address x3004 where we get the 2's complement for R4 (Current Address x3008)
0001 100 100 1 11111 ; Add -1 to R4 (will get R4 back to the way it was before doing the step in x3005) (Current Address x3009)
1001 100 100 111111 ; Not R4 to return it to the way it was before doing changing it for the step at x3005 (Current Address x300A)
0001 011 011 0 00 100; R3 <- R3+R4 (Current Address x300B)
1011 110 000000100; Store R6 indirectly in x5000 (Current Address x300C)
1011 110 000000100; Store R3 indirectly in x5001 (Current Address x300D)
1111 0000 00011001; HALT
0100000000000000 ; x4000
0100000000000001 ; x4001
0101000000000000 ; x5000
0101000000000001 ; x5001
```

divide.bin text

0011000000000000 ; Start at x3000

1010 011 000001110 ; Loading R3 with the address that holds x4000 (Current Address x3000)

1010 100 000001110 ; Loading R4 with the address that holds x4001 (Current Address x3001)

0101 110 110 1 00000 ; Assigning 0 to R6 (Current Address x3002)

1001 100 100 111111; Not R4 to be used in the subtraction (Current Address x3003)

0001 100 100 1 00001; R4 2's Complement needed for the subtraction (Current Address x3004)

0001 011 011 0 00 100 ; R3 <- R3-R4 (now R3 + (-R4)) (Current Address x3005)

0000 100 000000010; Break to address x3009 if R3 < 0 (negative) (Current Address x3006)

0001 110 110 1 00001 ; R6 <- R6+1 (Current Address x3007)

0000 001 111111100 ; Break to Address x3004 where we get the 2's complement for R4 (Current Address x3008)

0001 100 100 1 11111 ; Add -1 to R4 (will get R4 back to the way it was before doing the step in x3005) (Current Address x3009)

1001 100 100 111111 ; Not R4 to return it to the way it was before doing changing it for the step at x3005 (Current Address x300A)

0001 011 011 0 00 100; R3 <- R3+R4 (Current Address x300B)

1011 110 000000100; Store R6 indirectly in x5000 (Current Address x300C)

1011 110 000000100; Store R3 indirectly in x5001 (Current Address x300D)

1111 0000 00011001; HALT

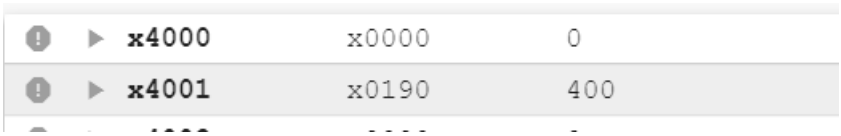
0100000000000000 ; x4000

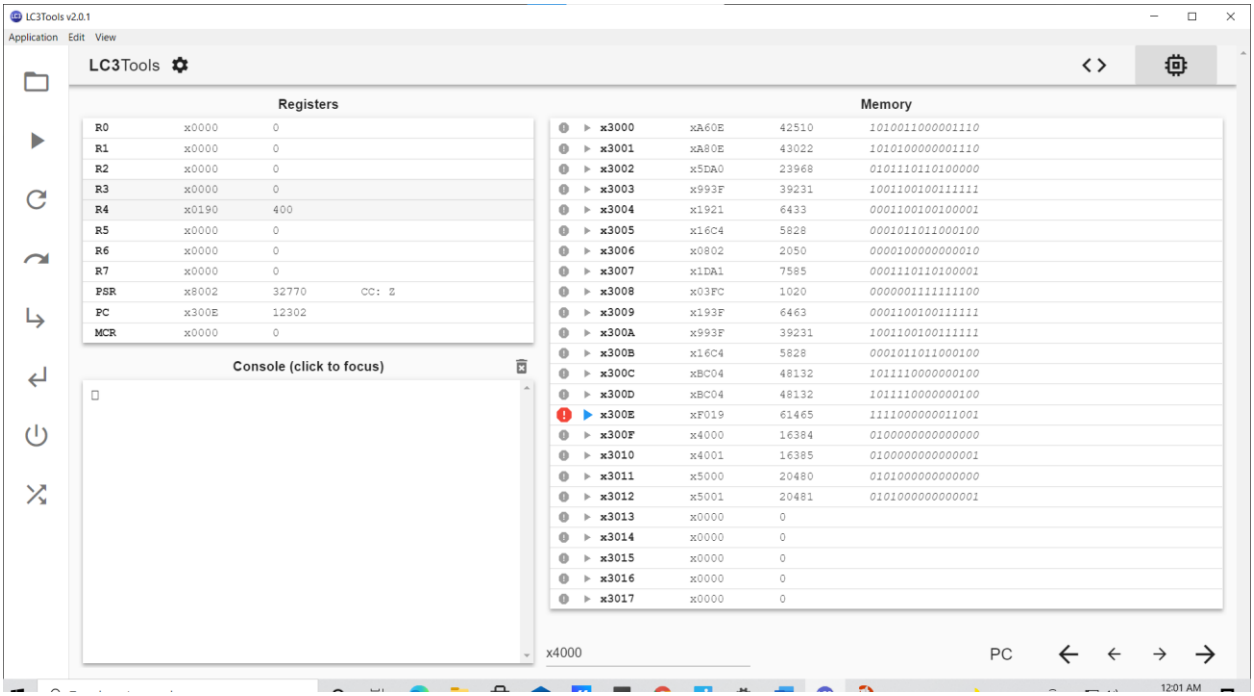
0100000000000001 ; x4001

0101000000000000 ; x5000

0101000000000001 ; x5001

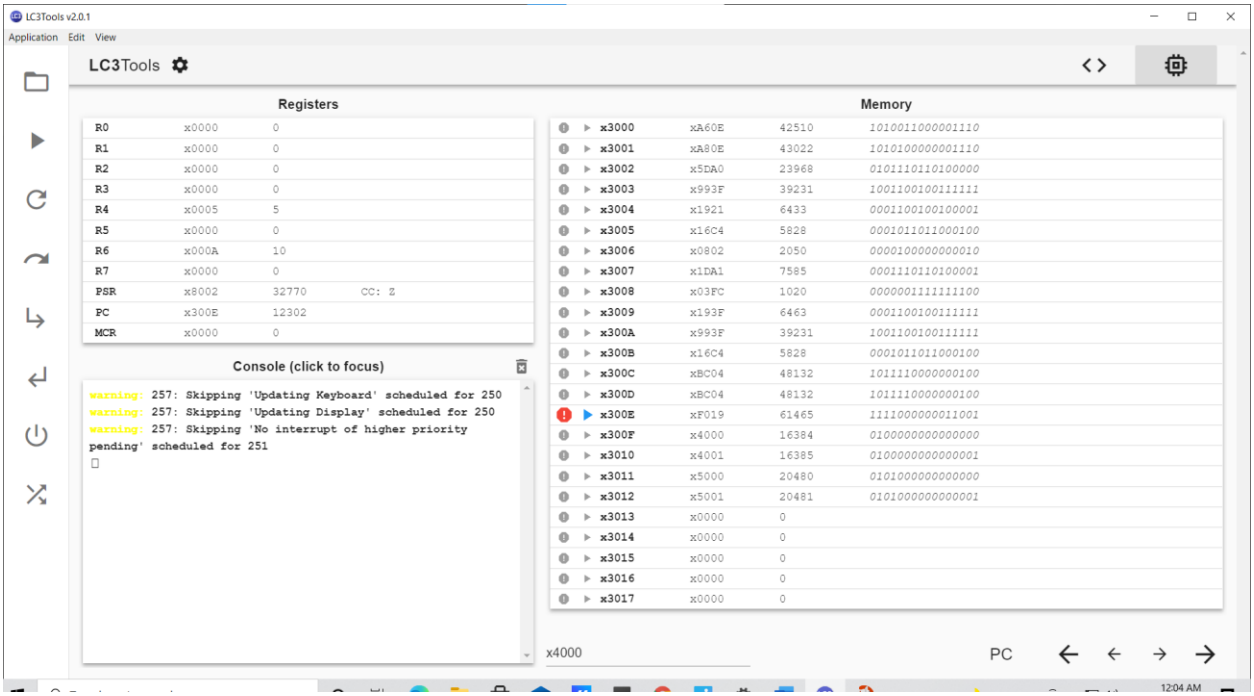
Registers when quotient is 0





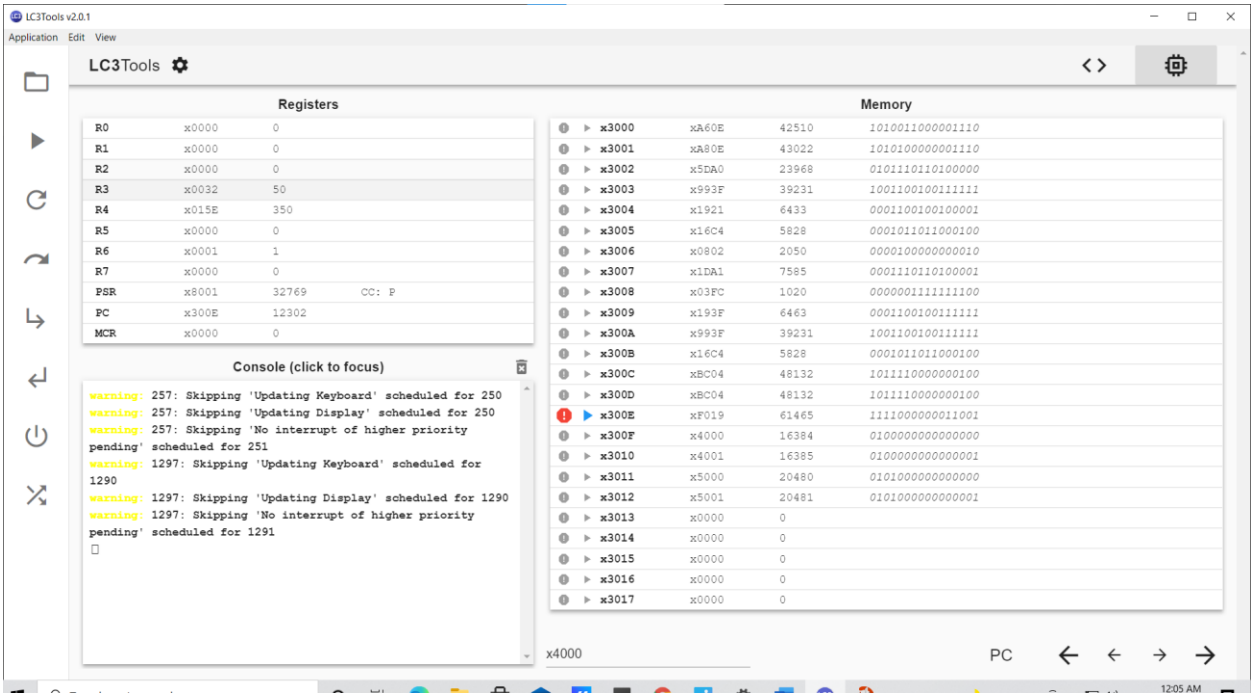
Registers when remainder is 0

x4000	x0032	50
x4001	x0005	5



Registers when the remainder is positive

x4000	x0190	400
x4001	x015E	350


LC3Tools v2.0.1
Application Edit View
LC3Tools
Registers
R0 x0000 0
R1 x0000 0
R2 x0000 0
R3 x0032 50
R4 x015E 350
R5 x0000 0
R6 x0001 1
R7 x0000 0
PSR x8001 32769 CC: P
PC x300E 12302
MCR x0000 0
Console (click to focus)
warning: 257: Skipping 'Updating Keyboard' scheduled for 250
warning: 257: Skipping 'Updating Display' scheduled for 250
warning: 257: Skipping 'No interrupt of higher priority pending' scheduled for 251
warning: 1297: Skipping 'Updating Keyboard' scheduled for 1290
warning: 1297: Skipping 'Updating Display' scheduled for 1290
warning: 1297: Skipping 'No interrupt of higher priority pending' scheduled for 1291
Memory
x3000 xA60E 42510 1010011000001110
x3001 xA80E 43022 1010100000001110
x3002 x5DA0 23968 0101110110100000
x3003 x993F 39231 1001100100111111
x3004 x1921 6433 0001100100100001
x3005 x16C4 5828 0001011011000100
x3006 x0802 2050 0000100000000010
x3007 x1DA1 7585 0001110110100001
x3008 x03FC 1020 0000001111111100
x3009 x193F 6463 0001100100111111
x300A x993F 39231 1001100100111111
x300B x16C4 5828 0001011011000100
x300C xBC04 48132 1011110000000100
x300D xBC04 48132 1011110000000100
x300E xF019 61465 1111000000011001
x300F x4000 16384 0100000000000000
x3010 x4001 16385 0100000000000001
x3011 x5000 20480 0101000000000000
x3012 x5001 20481 0101000000000001
x3013 x0000 0
x3014 x0000 0
x3015 x0000 0
x3016 x0000 0
x3017 x0000 0
x4000
PC