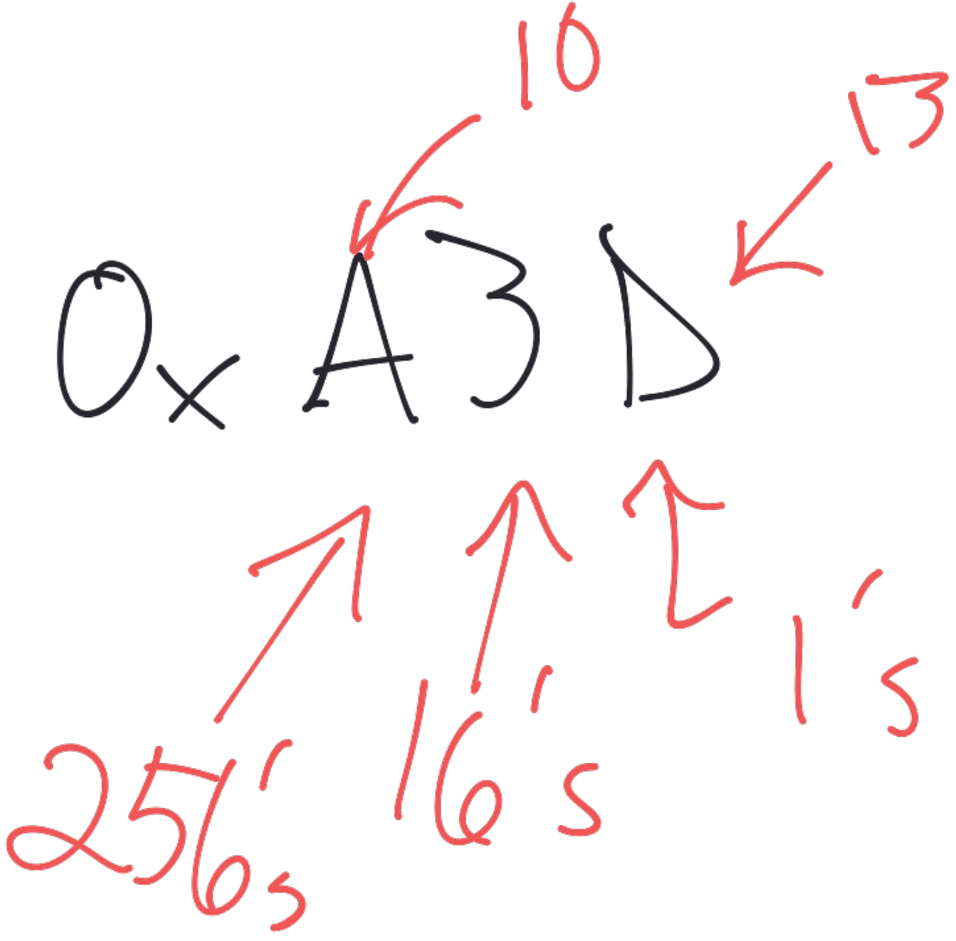
CS208

Wednesday

12 Jan 2022



10×256

$+ 3 \times 16$

$+ 13$

$2560 + 48$

$+3$

$= 2621$

0456 octal

64's 8's 1's

4×64 256

+ 5×8 + 40

+ 6 + 6

302

0b1011101

64 32 16 8 4 2 1

= 64 + 0 + 16 + 8 + 4 + 0 + 1

= 93

$$
\begin{array}{r}
37 \\
-32 \\
\hline
5
\end{array}
$$

$$
\begin{array}{r|l}
256 & 0 \\
16 & 2 \\
1 & 5
\end{array}
$$

$$0 \times 25$$

419
-256
163
-160
3

256 1
16 10 —
1 3

0x1A3

$419 = 0x1A3$

$0b\ 0001\ 1010\ 0011$

0 6 4 3

4-bit ints

⋮

1110	−2
1111	−1
0000	0
0001	+1
0010	+2
0011	+3
0100	

⋮

1101 ← signed magnitude

−5

1101 ← 3 2's Complement

"What do I add to this # to get 0?"

1 0011 ← 3

0000

0 1 1 1 7

0 1 1 0

0 1 0 1

0 1 0 0

0 0 1 1

0 0 1 0

0 0 0 1

0 0 0 0

1 1 1 1 -1

1 1 1 0

1 1 0 1 -3

1 1 0 0

1 0 1 1

5 + -3

0 1 0 1 5

1 1 0 1 -3

0 0 1 0

0x73

16×7+3

=115

01110011

+64+32+16 21 = 115

0x83

10000011

- 01111101

00000010

0x7D = 7×16 + 13

0111 1101 =

- of 0x83

=125

→ -125

0x83 ⟶ 10000011

What do we add to get 0?

125 (dec)

So 0x83 ~ −125

0x83 1000 0011

Step 1
Complement

0111 1100

Step 2
add 1

0111 1101

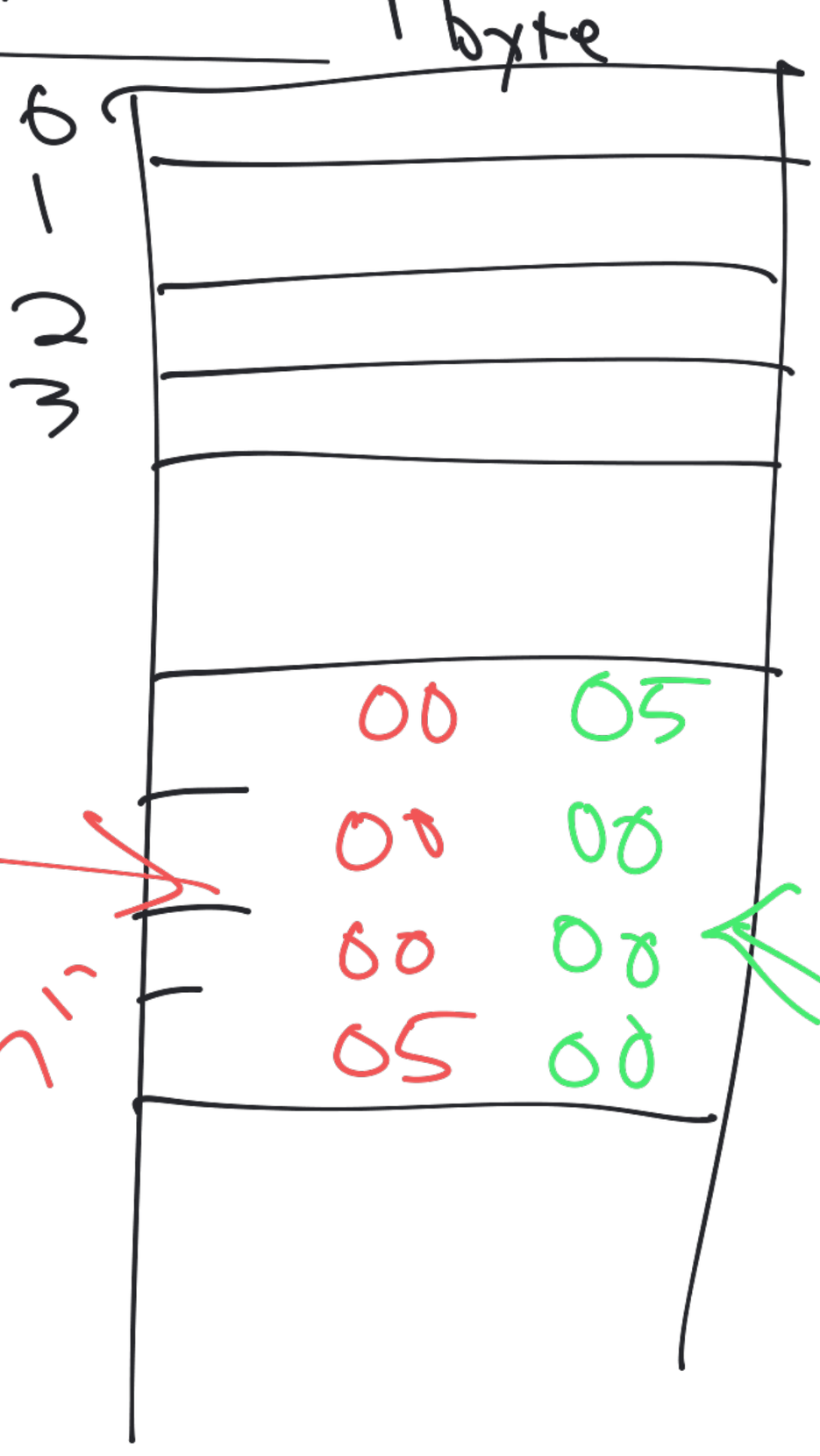
7 D

Step3
Answer =
neg of this

-125

Byte order

memory



"big endian"

I want to store 5 as a 32-bit 2's comp. integer here

0x00000005

"little endian"