Signed & Unsigned 4-bit Subtraction:

Unsigned subtraction: flip bits and add 1

```
13 -  1 =
```

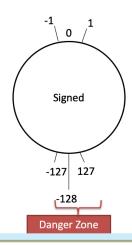Signed subtraction: flip bits and add 1

```
-3 - 1 =
```

A. 1100 & 1100
B. 1100 & 1010
C. 1010 & 1010
D. 1001 & 1100

If we add a positive number and a negative number, will we have overflow? (Assume they are the same # of bits)

A. Always

B. Sometimes

C. Never

-1 0 1

Signed

-127 127

-128

Danger Zone

How many of these <u>unsigned</u> operations have overflowed?

Interpret these <u>as 4-bit unsigned values</u> (valid range 0 to 15):

                                          carry-in carry-out
                                             ↓   ↓
Addition (carry-in = 0)

    9 + 11 =   1001 + 1011 + 0 =   1   0100
    9 + 6 =   1001 + 0110 + 0 =   0   1111
    3 + 6 =   0011 + 0110 + 0 =   0   1001

                        (-3)
Subtraction (carry-in = 1)

    6 − 3 =   0110 + 1100 + 1   = 1   0011
    3 − 6 =   0011 + 1001 + 1   = 0   1101

                        (-6)

A. 1
B. 2
C. 3
D. 4
E. 5

Try some 4-bit examples:

bit-wise operations:
- 0101 & 1101
- 0101 | 1101

Logical (unsigned) bit shift:
- 1010 << 2
- 1010 >> 2

Arithmetic (signed) bit shift:
- 1010 << 2
- 1010 >> 2