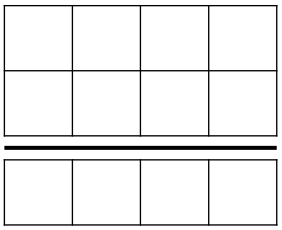
Bitwise Operators – Worksheet 1

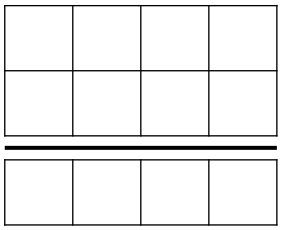
1. Convert the numbers to binary, perform the bitwise operation, then convert the final answer back to hexadecimal.

a) `0xA | 0x5`



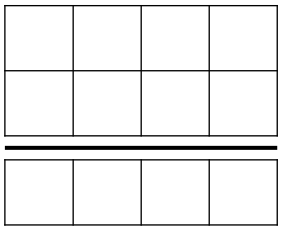
Hexadecimal: ☐

b) `0x7 & 0xE`



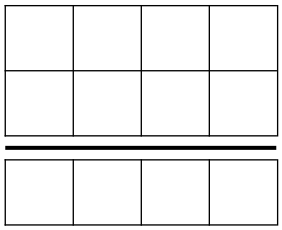
Hexadecimal: ☐

c) `0xC & 0x4`



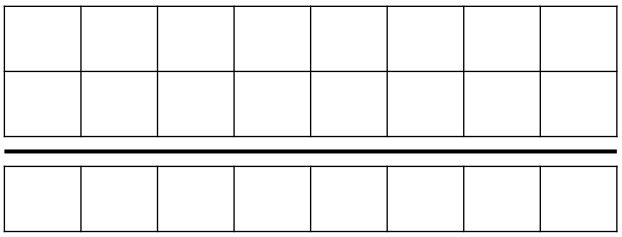
Hexadecimal: ☐

d) `0xA ^ 0x7`



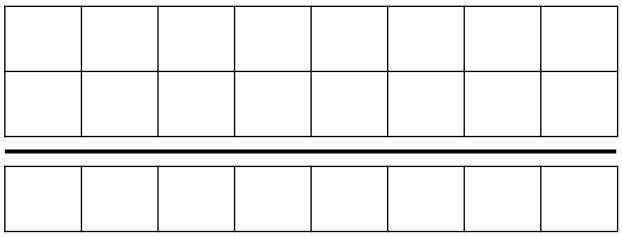
Hexadecimal: ☐

e) `0x5B & 0xF0`



Hexadecimal: ☐

f) `0xC3 | 0x0F`



Hexadecimal: ☐

2. Convert the initial value to binary, perform the bit shift operation, then convert the result back into hexadecimal.

	initial binary	resulting binary	hexadecimal
0x5 << 1	☐	☐	☐
0x8 >> 2	☐	☐	☐
0x5 >> 2	☐	☐	☐
0x1C >> 2	☐	☐	☐
0x1B >> 2 << 1	☐	☐	☐

3. For each description, write expression in the corresponding box to the right.

a) Given an integer, n, write a bitwise expression that returns one if the number is odd, and zero if the number is even. ☐

b) Given an integer, n, write a bitwise expression that returns zero if the number is odd, and one if the number is even. ☐

c) If all four of the least-significant (right-most) bits of an integer are zeros, the number is divisible by 16. Write a bitwise expression that will be zero if all four of the least-significant bits of a number, `n`, are zero, and non-zero if there are any 1's in the four right-most bits. ☐

d) Given two integers, `x` and `n`, write an expression that returns one if the x^{th} bit of `n` is set, and zero if it is not. ☐

e) Given two integers, `x` and `n`, write an expression that returns the number `n` with the x^{th} bit inverted. ☐