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`

1	0	1	0
0	1	0	1
1	1	1	1

Hexadecimal: 0xF

b) `0x7 & 0xE`

0	1	1	1
1	1	1	0
0	1	1	0

Hexadecimal: 0x6

c) `0xC & 0x4`

1	1	0	0
0	1	0	0
0	1	0	0

Hexadecimal: 0x4

d) `0xA ^ 0x7`

1	0	1	0
0	1	1	1
1	1	0	1

Hexadecimal: 0xD

e) `0x5B & 0xF0`

0	1	0	1	1	0	1	1
1	1	1	1	0	0	0	0
0	1	0	1	0	0	0	0

Hexadecimal: 0x50

f) `0xC3 | 0x0F`

1	1	0	0	0	0	1	1
0	0	0	0	1	1	1	1
1	1	0	0	1	1	1	1

Hexadecimal: 0xCF

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	0 0 0 0 0 1 0 1	0 0 0 0 1 0 1 0	0xA
0x8 >> 2	0 0 0 0 1 0 0 0	0 0 0 0 0 0 1 0	0x2
0x5 >> 2	0 0 0 0 0 1 0 1	0 0 0 0 0 0 0 1	0x1
0x1C >> 2	0 0 0 1 1 1 0 0	0 0 0 0 0 1 1 1	0x7
0x1B >> 2 << 1	0 0 0 1 1 0 1 1	0 0 0 0 1 1 0 0	0xC

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.

```
n & 0x1
```

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

```
(n&1)^1   or   ~n & 1
```

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.

```
n & 0xF
```

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.

```
(n>>x) & 0x1
```

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

```
n ^ (1<<x)
```