CSE 370 Spring 2000
Solutions for Assignment 1
04/06/2000

1. Does not require a solution

2. Ditto

3.
a) $011010_2 = 2^4 + 2^3 + 2^1 = \mathbf{26_{10}}$
b) $70_8 = 7 \times 8^1 = \mathbf{56_{10}}$
c) $AB3_{16} = 10 \times 16^2 + 11 \times 16^1 + 3 \times 16^0 = \mathbf{2739_{10}}$
d) $110011.011_2 = 2^5 + 2^4 + 2^1 + 2^0 + 2^{-2} + 2^{-3} = \mathbf{51.375_{10}}$

4.

a)

$212_{10} = 2 \times 106_{10} + 0$
$106_{10} = 2 \times 53_{10} + 0$
$53_{10} = 2 \times 26_{10} + 1$
$26_{10} = 2 \times 13_{10} + 0$
$13_{10} = 2 \times 6_{10} + 1$
$6_{10} = 2 \times 3_{10} + 0$
$3_{10} = 11_2$ (common knowledge...)

Now, read the bits from bottom to top: 11010100

Thus, $\mathbf{212_{10} = 11010100_2}$

b)

Each digit in octal corresponds to three digits in binary. We have $4_{10} = 100_2$ and $7_{10} = 111_2$, so that $\mathbf{47_8 = 100111_2}$.

c)

The integer part is easy: $5_2 = 101_2$

Now, let’s do the fractional part:

$0.3125_{10} \times 2 = 0.625_{10}$ ==> 0
$0.625_{10} \times 2 = 1.25_{10}$ ==> 1
$0.25_{10} \times 2 = 0.5_{10}$ ==> 0
$0.5_{10} \times 2 = 1_{10}$ ==> 1

Now, read the bits from top to bottom, so:
$0.3125_{10} = 0.0101_2$

We can now add the integer and the fractional results, so:
$\mathbf{5.3125_{10} = 101.0101_2}$

d)

Each hexadecimal digit corresponds to 4 binary digits. We have $2_{16} = 0010_2$, $D_{16} = 1101_2$ and $9_{16} = 1001_2$. Thus, **$2D9_{16} = 1011011001_2$**

5.

a)

	1		
		5	1
+		3	5
	1	**0**	**6**

b)

	1	1	1	1	
		0	0	0	1
+		1	1	0	1
+	0	0	1	1	0
	1	**0**	**1**	**0**	**0**

c)

	1	1	0	1	0	1
-	0	1	1	0	1	0
	1	1		1		
	0	**1**	**1**	**0**	**1**	**1**

d)

			1	1	0
x			0	1	0
			0	0	0
+		1	1	0	
+	0	0	0		
	0	**1**	**1**	**0**	**0**

6.

a)

We have $28_{10} = 011100_2$
Thus, -28 is represented as **111100**

b)

We have $28_{10} = 011100_2$
Invert the bits: 100011
Add 1: **100100**

c)

For n-bits 2s complement, the range is -2^{n-1} to $2^{n-1}-1$. For n = 16, the range is -2^{15} to $2^{15}-1$, which is **–32768 to 32767.**

d)

1100 is negative, and the absolute value is 0100_2, which is 4_{10}. So the value we have is -4_{10}. We now need to represent this in 8-bit 2's complement. 4_{10} in 8-bit's is 00000100. Now, we need to invert, which gives 11111011 and add 1: **11111100**.

Note that if you follow the following rule, you get the same results:

"To increase the number of bits in 2’s complement, just pad the bit stream to the left by inserting copies of the msb (most significant bit) from the original number until you get the desired number of bits."

This is called sign extension, and that’s the way that computers convert 2’s complement numbers from smaller to larger number of bits. For example, if you have a 32-bit integer and you add it to a 64-bit integer, the 32-bit integer is first sign-extended using the above mechanism to 64-bits before the addition is done.

7.

5_{10} = 0101
Invert the bits: 1010
Add 1: 1011

Thus X_4, X_2 and X_1 are 1, and X_3 is 0. Thus, the expression is $\mathbf{X_4X_3'X_2X_1}$

8.