CS2253 Assignment 1 – Isaac Shoebottom

1.

a) You would need 9 bits to have enough unique bit patterns to have a unique pattern for 453 scientists. (2^9).

b) This could store 59 more unique scientists. (512-453)

2.

a) The largest number you can store with 10 bits in 2s compliment format is 511. (01 1111 1111)

b) The largest negative number you can store with 10 bits in 2s compliment format is -512. (10 0000 0000)

c) The largest number you can store in the n^{th} bit 2s compliment is represented by this formula (2^(n-1)) – 1

d) The largest negative number you can store in n^{th} bit 2s compliment is represented by this formula -(2^(n-1))

3.

a)
11010
negation
00101
+
00001
=
00111
(1+2+3)
=
6
(first bit is 1, therefor negative)
-6

b)
01100110011
1+2+0+0+16+32+0+0+256+512
819

c)
11111111010
negation
00000000101
+
00000000001
=
00000000111
(1+2+3)
=
6
(first bit is 1 therefor negative)

-6
d)
01110000010000
(0+0+0+0+16+0+0+0+0+0+1024+2048+4096)
7184

4.

a)
11011 – 111

1111111
 11111011
 -
 11111111
 =
 11111100
-5 – (-1) = -4

b)
11111111
-
11111011
=
00000100
-1 – (-5) = 4

5.

a)
00111 + 00110 = 01101 No overflow
7 + 6 = 13.
b)
10111 – 11110 = 11001 No overflow
-9 – (-2) = -7
c)
11000 – 00011 = 10101 No overflow
-8 – (3) = -11
d)
10110 + 10011 = 01001 Overflow
The two leftmost carry bits are 10, which means there is overflow. We also have added two negative numbers and resulted in a positive number, which means overflow.
-10 + -13 = -23 not 9

6.

```
      11
F6 =  1111 0110
49 =  0100 1001
=     0011 1111
```

1111 0110 + 0100 1001 = 0011 1111

There cannot be overflow since you are adding a negative and a positive number. It is impossible because any combination is within the maximum and minimum values. It is also not overflow because the two most left carry bits are the same.

7.

a)
0101 0100 0111 1000
AND
1111 1101 1110 1010
=
0101 0100 0110 1000
=
0x5468
b)
1010 1011 1100 1101
OR
0001 0010 0011 0100
=
1011 1011 1111 1101
=
0xBBFD

c)
NOT((NOT(1101 1110 1111 1010)) AND (NOT(1111 1111 1111 1111)))
NOT(0010 0001 0000 0101 AND 0000 0000 0000 0000)
NOT(0000 0000 0000 0000)
1111 1111 1111 1111
=
0xFFFF
d)
0000 0000 1111 1111
XOR
0011 0010 0011 0010
=
0011 0010 1100 1101
=
0x32CD

8.

a)
1 01111110 01000000000000000000000
- 2^(126-127) (1 + 1/4) = -0.625

b)
0 10000001 10100000000000000000000
+ 2^(129-127) (1 + 1/2 + 1/8) = 6.5

9.

10.75 = 0 10000010 01011000000000000000000

10.

A mathematical expression to change a decimal digit's binary representation to it's ascii counterpart would be to say that, for $9 \geq n \geq 0$, $(n_2 + 110000) = n$ in ascii.

For example, take 4:
$4_2 = 100$. 110000 + 100 = 110100 which is 4 in ASCII format.