Adam Gincel

Professor: Antonio R. Nicolosi

September 12th 2014

I pledge my honor I have abided by the Stevens Honor System.

1.

a) 0011 1001 1001 1001
0011 1001 1001 1010
0011 1001 1001 1011
0011 1001 1001 1100
0011 1001 1001 1101
0011 1001 1001 1110
0011 1001 1001 1111
0011 1001 1010 0000
0011 1001 1010 0001

b)

$(1100\ 0001\ 0111)_2 = (3095)_{10}$
$(1001\ 1111\ 0000)_2 = (2544)_{10}$
$(1101\ 0001\ 1110)_2 = (3358)_{10}$
$(0001\ 1111\ 1111\ 1111)_2 = (8191)_{10}$

$(554)_8 = (364)_{10}$
$(3440)_8 = (1824)_{10}$
$(7777)_8 = (4095)_{10}$

$(554)_{16} = (1364)_{10}$
$(720)_{16} = (1824)_{10}$
$(1ff)_{16} = (511)_{10}$

c)

```
    1001 1100 0011 1000
+   0101 0111 0101 1011

=   1111 0011 1001 0011


    1000 1001 1010 1111
+   0001 1011 0011 1111

=   1010 0100 1110 1110
```

	0011 0010 0101 0111
+	0110 1011 1001 1001
=	1001 1101 1111 0000

d)

$(35)_{10} = (0010\ 0011)_2, (43)_8, (23)_{16}$

$(512)_{10} = (0010\ 0000\ 0000)_2, (1000)_8, (200)_{16}$

$(1000)_{10} = (0011\ 1110\ 1000)_2, (1750)_8, (3E8)_{16}$

$(2014)_{10} = (0111\ 1101\ 1110)_2, (3736)_8, (7DE)_{16}$

$(3010)_{10} = (1011\ 1000\ 0010)_2, (5702)_8, (BC2)_{16}$

$(5555)_{10} = (0001\ 0101\ 1011\ 0011)_2, (12663)_8, (15B3)_{16}$

$(8192)_{10} = (0010\ 0000\ 0000\ 0000)_2, (20000)_8, (2000)_{16}$

$(10001)_{10} = (0010\ 0010\ 0011\ 1001)_2, (21071)_8, (2239)_{16}$

e)

$\log_2(n) = 30.51$…round up to next integer, it takes 31 bits to store 1,534,223,121 in binary as an unsigned integer.

$\log_8(n) = 10.17 = 11$ bits.

$\log_{16}(n) = 7.628 = 8$ bits.

f)

$(0011\ 1111\ 1111)_2$ needs 4 decimal digits.

$(6666)_8$ needs 4 decimal digits.

$(fad)_{16}$ also needs 4 decimal digits.

2.

a)

$(-102)_{10} = (1001\ 1010)_2$ under two’s complement.
$(-748)_{10} = (1101\ 0001\ 0100)_2$ under two’s complement.

$(-8191)_{10} = (1110\ 0000\ 0000\ 0001)_2$ under two's complement.
$(-16384)_{10} = (1100\ 0000\ 0000\ 0000)_2$ under two's complement.

b)

```
      1001 0011 1001 1001
+     0111 0000 1001 0110
=   1 0000 0100 0010 1111 – overflow
```

```
      0100 1001 0101 1111
+     0011 1010 1101 1111
=     1000 0100 0011 1110 – overflow under two's complement (two positives added up to a negative)
```

```
      1000 0101 1010 1101
      1101 1111 1110 1110
=   1 0110 0101 1001 1011 – overflow
```

c) As long as there is enough room for a leading 0 in positive numbers, you cannot overflow while negating a number. The process of flipping the 0s and 1s never adds more digits.

d)

```
      0111 0001 1101 0011
+     0011 1110 0110 1101
=     1011 0000 0100 0000 – overflow (two positives = negative)
```

```
      0001 0011 0011 1111
+     1100 1000 1111 1010
=     1101 1100 0011 1001
```

```
      1111 0001 0011 0111
+     1111 0000 0000 0001
=   1 1110 0001 0011 1000 - overflow
```