# LBSC 690: Information Technology
# Homework 01:
# Computers and Data

William Webber
CIS, University of Maryland

Spring semester, 2012

Answers due via email to `wew@umd.edu` by Tuesday, 7th February, at 3:30pm ET.

## Question 1: Binary addition

*Adding two multiple-digit binary numbers together is similar to adding two "regular" (decimal) numbers together. We work a column at a time, right to left, add each column, and carrying any remainder over into the next column to the left. Binary addition, however, takes a particularly simple form, because only two values are involved.*

***Part a**:*

*Add the following (6-bit) binary numbers together (please do the working carefully, but you only need to show me the answers):*

$$\textit{001010} \quad + \quad \textit{011001} = 100011 \tag{1}$$
$$\textit{010111} \quad + \quad \textit{010110} = 101101 \tag{2}$$
$$\textit{011111} \quad + \quad \textit{000001} = 100000 \tag{3}$$

**Part b**:

*Convert the above (operands and results) into decimal, and check that the addition is correct.*

$$10 + 25 \quad = \quad 35 \tag{4}$$
$$23 + 22 \quad = \quad 45 \tag{5}$$
$$31 + 1 \quad = \quad 32 \tag{6}$$

□

*Binary arithmetic in modern computers is done to a fixed width. That is, if you add two 6-bit values together, then the result also have to fit into 6 bits. Any carry over from the left-most bit is thrown away.*

***Part c**:*

*Obeying the rules of fixed-width binary arithmetic, and assuming that we are working with 6-bit values, compute the following sum:*

$$111111 \quad + \quad 000001 = 000000 \tag{7}$$

Explanation: we keep carrying the 1 to the left, then when we carry from the left-most (6th) bit, the 1 is thrown away, since we're restricted to 6 bits.
□

*Under the original UNIX operating system, the current time is represented as a 32-bit number, counting the number of seconds since January 1st, 1970. As each second ticks by, one is added to this internal time stamp.*

***Part d**:*

*What is the last date (in seconds, and in years — approximation is ok) that the 32-bit UNIX timestamp is able to represent correctly?*

As stated, the question assumes that the timestamp is unsigned. The maximum value that a 32-bit binary number can store is roughly 4 billion (exactly $2^{32} - 1$). Ignoring leap years (and leap seconds!) there are $60 \times 60 \times 24 \times 365 = 31{,}536{,}000$ seconds in a year, which means that an unsigned 32-bit timestamp can represent up to $((2^{32}) - 1)/(31{,}536{,}000) \approx 136$ years. Counting 136 years from 1970 takes us to 2106.

In fact, the UNIX timestamp is signed, so it can represent half as many years, or roughly $68$, taking us up to 2038.

**Part e**:

*What time will the UNIX timestamp show when we are one second past the last date it can represent?*

If the timestamp were unsigned, then the binary number wraps round again to $0$ (see the solution to Part c. above), in which case we're back to January 1st, 1970.

In fact, the UNIX timestamp in signed. A signed binary number wraps around from the largest representable positive number ($01111\ldots$) to the largest representable negative number ($10000\ldots$), so we'll go back $68$ years before 1970, to 1902. (These workings are all approximate.)

*[[ Note: this limitation has been fixed in most modern UNIX systems by moving to a $64$-bit timestamp—you might want to think about when this will run out. But there may be some historical UNIX system that will encounter this problem when the fatal time comes. ]]*

# Question 2: Sneaker-net

*When we think of transferring data from one computer to another nowadays, we almost automatically think of doing so using a network. An alternative, though, is to copy the data to some (semi-) removable medium, and then physically carry that medium to the destination. Such a method is facetiously known as "sneaker-net", since we are not using cables to communicate the data (as in ethernet and similar network techonologies), but our sneakers.*

*Assume that I need to transfer* $100$ *GB of data from a source computer to a destination computer, and that I have two possibilities for transferring the data:*

1. *Send it over a network, which has a speed of* $100$ *Mbps; or*
2. *Unplug the hard drive the data is on, carry it to the other computer (walking at* $5$ *feet per second), and plug it in.*

*Assume that unplugging and plugging-in the hard drive take no time; the only elapsed time is in walking.*

***Part a**:*

*Assuming I have to walk* $1{,}000$ *feet to the destination computer, what is the effective bandwidth of our sneaker-net connection?*

It will take me $1{,}000/5 = 200$ seconds to walk the distance. I'm carrying $100$GB with me, so I've transferred $100$GB in $200$ seconds, which is $100/200 = 0.5$GB per second. Network transfer times are generally stated in bits, not bytes, per second; there are $8$ bits in a byte; so my transfer speed is $0.5 * 8 = 4$Gbps. (Note carefully the difference between GBps and Gbps!)

**Part b**:

*How far away does the destination computer have to be before it is faster to send the data over the network than to carry it by sneaker-net?*

The network speed is $100$Mbps (note: bits!), which is $(100/8) = 12.5$MBps, or $(12.5)/1000 = 0.0125$GBps (using the modern convention that a GB is 1000 MB). I'm sending $100$GB, which will take $(100/0.0125) = 8000$ seconds over the network. In $8000$ seconds, I can walk $5 \times 8000 = 40{,}000$ feet. So the destination computer must be more than $40{,}000$ feet (roughly $7.5$ miles) away before it is faster to send the data by the network than to work it over.