Questions

1. Convert the following decimal numbers to binary and back again.
   (a) 34
   (b) 82

2. The algorithm for addition of two positive integers is as easy with the binary representation
   as it is with the decimal representation.
   Show that 26 + 27 = 53, by converting 26 and 27 to binary, computing the sum, and then
   converting back to decimal.

3. A byte is $n = 8$ bits. When the bits are interpreted as positive 8 bit integers, the values range
   from 0 to 255. Write out the bytes (binary numbers) that correspond to integers 127 to 130.

4. Integers are typically represented in a computer using a fixed number of bits, namely 8, 16,
   32, or 64. In Java, these correspond to primitive types `byte`, `short`, `int`, and `long`. To represent
   positive integers one has to declare the type to be unsigned.
   What is the positive largest number that can be represented using the above lengths, and
   roughly how much is each (written in decimal)?

5. We represent numbers in base 8 as follows:

   \[
   (a_{N-1}\cdots a[2] a[1] a[0] )_8 = a[N-1] \times 8^{N-1} + \cdots + a[2] \times 8^2 + a[1] \times 8 + a[0]
   \]

   where the $a[i]$ are all in \{0, 1, 2, \cdots, 7\}.
   Convert (736)$_8$ and (1205)$_8$ from base 8 to decimal.

6. Perform the sum $(1205)_8 + (736)_8$.

7. Perform the following sum. The two numbers are represented in base 5:
   \[
   (2430)_5 + (2243)_5 .
   \]
   Do not convert the numbers from base 5 to decimal!

8. Convert 238 from decimal to base 5, that is, write 238 as a sum of powers of 5.
Answers

1. (a) $100010 = 2^5 + 2^1 = 32 + 2$
   (b) $1010010 = 2^6 + 2^4 + 2^1 = 64 + 16 + 2$

2. 
   \[
   \begin{array}{c}
   \text{00011010} \leftrightarrow 26 \\
   + \text{00011011} \leftrightarrow 27
   \end{array}
   \]
   To perform the addition, use the same algorithm that you use with decimal, except that you are only allowed 0’s and 1’s. Whenever the sum in a column is 2 or more, you carry to the next column, since it contributes to the next power of 2:
   \[
   2^i + 2^i = 2^{i+1}
   \]
   So,
   \[
   \begin{array}{c}
   \text{00110100} \leftarrow \text{carry bits} \\
   \text{00011010} \leftrightarrow 26 \\
   + \text{00011011} \leftrightarrow 27 \\
   \text{00110101} \leftrightarrow 53
   \end{array}
   \]
   In fact, this is basically how computers do arithmetic as you will learn in COMP 273.

3. 127 is 0111111, 128 is 10000000, 129 is 10000001, 130 is 10000010

4. The largest positive integer you can represent with $N$ bits is $2^N - 1$. Since $2^{10} \approx 10^3$, we have
   - $2^{16} - 1 \approx 2^6 \times 10^3 = 64,000$
   - $2^{32} - 1 \approx 2^4 \times 2^{30} = 2^2 \times (10^3)^3 = 4 \times 10^9$
   - $2^{64} - 1 \approx 2^4 \times 2^{60} = 16 \times (10^3)^6 = 16 \times 10^{18}$ which is a very big number.

5. 
   \[
   (736)_8 = 7 \times 8^2 + 3 \times 8 + 6 = (478)_{10}
   \]
   \[
   (1205)_8 = 1 \times 8^3 + 2 \times 8^2 + 0 \times 8 + 5 = (645)_{10}.
   \]

6. 
   \[
   \begin{array}{c}
   101 \\
   + (736)_8
   \end{array}
   \]
   \[
   \begin{array}{c}
   \hline
   \text{(2143)}_8 \\
   \text{which is 1123 in base 10.}
   \end{array}
   \]
7. The answer is 

\[
\begin{array}{ccc}
110 & \leftarrow \text{ carry} \\
2430 \\
+2243 \\
\hline
10223
\end{array}
\]

If you would like to practice converting between different bases, see the website: [http://www.cleavebooks.co.uk/scol/calnumba.htm](http://www.cleavebooks.co.uk/scol/calnumba.htm)

8. Apply the same algorithm we saw for binary, but now we divide by 5 at each step and write down the remainder.

<table>
<thead>
<tr>
<th>i</th>
<th>decimal</th>
<th>base 5 coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>47</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

and so \((238)_{10} = (1423)_{5} = 1 \cdot 5^3 + 4 \cdot 5^2 + 2 \cdot 5^1 + 3 \cdot 5^0 = 125 + 100 + 10 + 3.\)