Announcements

Reminder
- Program 8 due tomorrow night!

List of Topics to Review

- Functions
- Local vs. Global Variables
- If/Else Statements
- For Loops
- While Loops
- Nested loops
- Reading/Writing to/from Files
- Exception Handling
- Strings
- Lists
- 2-D Lists
- Dictionaries
- Simple Graphics package
- Binary numbers
- Sorting algorithms

Underlying Data Representation

- Remember back to the beginning of the semester
- We said that all data in a computer is stored in sequences of 0s and 1s
- **Byte**: just enough memory to store letter or small number
  - Divided into eight bits
  - **Bit**: electrical component that can hold positive or negative charge, like on/off switch
  - The on/off pattern of bits in a byte represents data stored in the byte
Binary Numbers

A Binary Number is made up of only 0s and 1s.

Example of a Binary Number

110100

There is no 2,3,4,5,6,7,8 or 9 in Binary!

How do we count using binary?

<table>
<thead>
<tr>
<th>Binary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>We start at 0</td>
</tr>
<tr>
<td>1</td>
<td>Then 1</td>
</tr>
<tr>
<td>???</td>
<td>But then there is no symbol for 2...what do we do?</td>
</tr>
</tbody>
</table>

How do we count in Decimal?

<table>
<thead>
<tr>
<th>Decimal</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Start at 0</td>
</tr>
<tr>
<td>...</td>
<td>Count 1,2,3,4,5,6,7,8</td>
</tr>
<tr>
<td>9</td>
<td>This is the last digit in Decimal</td>
</tr>
<tr>
<td>10</td>
<td>So we start back at 0 again, but add 1 on the left</td>
</tr>
</tbody>
</table>

Applying to Binary

<table>
<thead>
<tr>
<th>Binary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>We start at 0</td>
</tr>
<tr>
<td>*</td>
<td>1</td>
</tr>
<tr>
<td>**</td>
<td>10</td>
</tr>
<tr>
<td>***</td>
<td>11</td>
</tr>
<tr>
<td>****</td>
<td>100</td>
</tr>
</tbody>
</table>
Decimal vs. Binary

<table>
<thead>
<tr>
<th>Decimal</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>0</td>
<td>1</td>
<td>10</td>
<td>11</td>
<td>100</td>
<td>101</td>
<td>110</td>
<td>111</td>
<td>1000</td>
<td>1001</td>
<td>1010</td>
<td>1011</td>
<td>1100</td>
<td>1101</td>
<td>1110</td>
<td>1111</td>
</tr>
</tbody>
</table>

"Binary is as easy as 1, 10, 11."

Binary Numbers

- Each position in a binary number represents $2^n$
- Thus $10101 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
- $101101 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

This is the same as the decimal system:

$193 = 1 \times 10^2 + 9 \times 10^1 + 3 \times 10^0$

Converting Decimal to Binary

- $47 / 2 = 23$ rem 1
- $23 / 2 = 11$ rem 1
- $11 / 2 = 5$ rem 1
- $5 / 2 = 2$ rem 1
- $2 / 2 = 1$ rem 0
- $1 / 2 = 0$ rem 1

Hence 47 in decimal format equals 101111 in binary format.

Adding Binary Numbers

\[
\begin{array}{c}
10011 \\
+ 1111 \\
\hline
100010
\end{array}
\]
Subtracting Binary Numbers

\[
\begin{array}{c}
101100 \\
- \ 1111 \\
\hline
11101
\end{array}
\]

Converting Binary to Decimal

\[
\begin{array}{c}
10111 \\
\times 16 \quad \times 8 \quad \times 4 \quad \times 2 \quad \times 1 \\
\hline
16 + 0 + 4 + 2 + 1 = 23
\end{array}
\]

Practice

- Write 2 functions:
  - toBinary(decimal) – takes in a decimal number and returns its binary equivalent
  - toDecimal(binary) – takes in a binary number and returns its decimal equivalent
- Hints:
  - In toDecimal, you should convert binary to a string
  - In toBinary, you should create binary as a string, then typecast it to an integer before returning.
- Examples:

```python
print(toBinary(1198))  # Prints 10010101110
print(toBinary(5))     # Prints 101
print(toDecimal('10001110'))  # Prints 142
print(toDecimal('11'))   # Prints 3
```