1. Google “why use assembly language?” Specify 3 different websites that answer this question in as many different ways as possible; and write a short paragraph from each *in your own words* (i.e. do NOT copy/paste) to explain the reason and importance (or lack thereof) for using assembly language.

2. Define the following as it relates to computer science IN YOUR OWN WORDS:
   a. Address bus
   b. Data bus
   c. Word (i.e. word size)
   d. Pipelining
   e. Address Alignment
   f. ISA (instruction set architecture)
   g. Load/Store architecture

3. Show your work to convert the following non-decimal values into the base 10 equivalent. You can check your work with a program or calculator, but you must write out the long format of how the resulting calculation was determined for each problem.
   a. \((54321)_6 = (\text{__________})_{10}\)
   b. \((345)_9 = (\text{__________})_{10}\)
   c. \((130210321)_4 = (\text{__________})_{10}\)

4. Show your work to convert the following base 10 values into the specified base equivalent. You can check your work with a program or calculator, but you must write out the long format of how the resulting calculation was determined for each problem.
   a. \((1,231,123)_{10} = (\text{__________})_5\)
   b. \((1,231)_{10} = (\text{__________})_8\)
   c. \((99,999)_{10} = (\text{__________})_2\)

5. There is a shortcut way of converting from binary to hex and octal. Assuming that the colored boxes represent a bit value of 1 and the non-colored boxes represent a bit value of 0, use this shortcut method to fill in the HEX and OCT columns to designate the equivalent value in these different bases.
7. Assume that each of the following five values is stored as a 32-bit integer value; and these 5 values are being stored sequentially in memory starting at address $0x200$. Show the hex version of memory for both the big endian and little endian (as designated) byte addressing schemes. Be sure to extend the “Byte order” column specifying the correct address values for each byte of memory needed to store these 5 values. HINT: 32-bits is 4 bytes, so 5 values at 4-bytes each would be a total of 20 bytes of memory.

<table>
<thead>
<tr>
<th>Value</th>
<th>Byte order</th>
<th>Big Endian</th>
<th>Little Endian</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x124CD1</td>
<td>0x200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xABAF5567</td>
<td>0x201</td>
<td>0x200</td>
<td></td>
</tr>
<tr>
<td>0x129C5</td>
<td>0x202</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x1</td>
<td>0x203</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x1123A9D</td>
<td>0x204</td>
<td>0x202</td>
<td>0x200</td>
</tr>
<tr>
<td></td>
<td>0x205</td>
<td>0x203</td>
<td>0x201</td>
</tr>
<tr>
<td></td>
<td>0x206</td>
<td>0x204</td>
<td>0x202</td>
</tr>
<tr>
<td></td>
<td>0x207</td>
<td>0x205</td>
<td>0x203</td>
</tr>
<tr>
<td>etc</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8. Write a C program to print the minimum and maximum values as designated by the #define directive in the limits.h file; and as pertains to the C language on the X86 architecture (i.e. on our system). Write both the signed and unsigned values to the screen for the following types:

   a. Minimum character value
   b. Maximum character value
   c. Minimum short integer value
   d. Maximum short integer value
   e. Minimum integer value
   f. Maximum integer value
   g. Minimum long integer value
   h. Maximum long integer value

HINT: here’s a decent website: [http://tigcc.ticalc.org/doc/limits.html](http://tigcc.ticalc.org/doc/limits.html); but there are others.

Given the following binary value, 001100110011, convert it to a decimal value for each of the specified integer formats; and show the character value for the text format. Show/explain your work where applicable.

   a. Text Unicode
   b. Integer unsigned (B2U)
   c. Integer signed (B2T)
   d. Integer B2O
   e. Integer B2S

Given the following binary value, 101100110011, convert it to a decimal value for each of the specified integer formats; and show the character value for the text format. Show/explain your work where applicable.

   a. Text Unicode
   b. Integer unsigned (B2U)
   c. Integer signed (B2T)
   d. B2O
   e. B2S

Given the decimal value 299, convert it to the equivalent **hex value** for each of the specified formats. Show/explain your work where applicable.

   a. Integer unsigned (B2U)
   b. Integer signed (B2T)
   c. B2O
   d. B2S

Given the decimal value -132, convert it to the equivalent **hex value** for each of the specified formats. Show/explain your work where applicable.

   a. Integer unsigned (B2U)
   b. Integer signed (B2T)
   c. B2O
   d. B2S