Instructions: This is the first homework for CS170 (Section 004). Unlike labs, you are expected to do homeworks on your own. Future homeworks will involve programming, but this first one is mostly written, and you should turn it in on paper.

Honor Code: Like all work for this class, the Emory Honor Code applies. You should do your own work on all problems, unless you are explicitly instructed otherwise. If you get stuck or have questions, ask your instructor or a TA for help.

Initial here to indicate that you followed the Honor Code and this work is your own. ______________

1. Bits and Bytes. (Recall, for CS 'kilo' means a number that is a power of 2 that is close to 1000.)
   a) (5pts) How many bytes are in 2.5 megabytes? _____2,621,440____________________
   b) (5 pts) How many bits are in 2.5 megabytes? _____20,971,520____________________

2. (20 pts) Using ALL of the following terms, describe how we can write programs for a computer.

<table>
<thead>
<tr>
<th>program</th>
<th>high-level language</th>
<th>Java</th>
</tr>
</thead>
<tbody>
<tr>
<td>machine language</td>
<td>assembly language</td>
<td>compiler</td>
</tr>
<tr>
<td>source code (or source files)</td>
<td>virtual machine</td>
<td></td>
</tr>
</tbody>
</table>

In this class, we write programs in Java, which is a high-level language. HLLs are easier for humans to read and write. However, computers do not understand HLLs. Instead, they understand binary machine languages. HLLs must be translated from a HLL into an intermediate assembly language and then to the machine language. This process occurs via the compiler. The executable byte code can then be run in the virtual machine on any platform.
3. ASCII Conversions
Letters in the English alphabet are stored inside the computer as numbers. That is, each letter is \textit{encoded} as a number. The encoding method used to represent the alphabet is known as the ASCII code (American Standard Code for Information Interchange). The website \url{http://www.ascii-code.com} contains ASCII code tables which shows the encoding from a decimal number to an English language character.

a) (10 pts) What text is encoded by the following sequence of (decimal) byte codes?

\begin{verbatim}
72 101 108 108 111 32 70 97 108 108 49 51 32 99 115 49 55 48
\end{verbatim}

Hello Fall13 cs170

(-1 per incorrect character up to a total of -10)

b) (10 pts) Give the sequence of decimal byte codes for the following phrase:

\begin{verbatim}
@987 xyz?
\end{verbatim}

\begin{verbatim}
64 57 56 55 32 120 121 122 63
\end{verbatim}

(-1 per incorrect number, up to total of -10)
4. (10 pts) Variable names:
   a) Give 3 valid identifiers in Java, two of which are NOT solely alphabetic.
      Answers vary. See Section 2.4 (8th ed)
      Examples: tax, _tax, tax2...
   
   b) Give 3 Java keywords which could NOT be used as variable names:
      Answers vary. See list of words in Appendix A
      Examples: break, char, else...
   
   c) Give 3 illegal identifiers that are NOT Java keywords:
      Answers vary. See Section 2.4 (8th ed)
      Examples: 2tax, tax^2, a+1...
   
   d) Give 1 convention (i.e., not a syntax rule enforced by Java) that you should follow when naming identifiers.
      Answers vary. See Section 2.8 (9th ed) or Section 2.16 (8th ed)
      Names should be descriptive, short enough to type easily, lowercase, ...

5. (10pts) Using Google or another search engine of your choice, fill in the table below by finding the minimum and maximum integer values which can be represented by the following Java datatypes. Also for each datatype, how many bits would be required to encode such an integer?

<table>
<thead>
<tr>
<th>Datatype</th>
<th>Minimum Value</th>
<th>Maximum Value</th>
<th>No. of Bits Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) byte</td>
<td>-128</td>
<td>127</td>
<td>8 bits</td>
</tr>
<tr>
<td>b) short</td>
<td>-32768</td>
<td>32767</td>
<td>16 bits</td>
</tr>
<tr>
<td>c) char</td>
<td>'\u0000' or 0</td>
<td>'\uffff' or 65535</td>
<td>16 bits</td>
</tr>
<tr>
<td>d) int</td>
<td>-2147483648</td>
<td>2147483647</td>
<td>32 bits</td>
</tr>
<tr>
<td>e) long</td>
<td>-9223372036854775808</td>
<td>9223372036854775807</td>
<td>64 bits</td>
</tr>
</tbody>
</table>
6. Algorithm Tracing
   Consider the following algorithm:

   \[
   \begin{align*}
   R &= 46; \\
   Q &= 0; \\
   D &= 6; \\
   \text{as long as } R \geq D \text{ do} \{ \\
   &\quad R = R - D; \\
   &\quad Q = Q + 1; \\
   \} \\
   \end{align*}
   \]

   a) (20 pts) Trace out each step as begun below for you:

   Initially: \hspace{1cm} R = 46 \hspace{1cm} Q = 0
   After 1 Step: \hspace{1cm} R = \hspace{1cm} Q =
   After 2 Steps: \hspace{1cm} R = \hspace{1cm} Q =
   \hspace{1cm} \text{... (use as many rows as needed)}

   init \hspace{1cm} R=46 \hspace{1cm} Q=0
   step1: \hspace{1cm} R=40 \hspace{1cm} Q=1
   step2: \hspace{1cm} R=34 \hspace{1cm} Q=2
   step3: \hspace{1cm} R=28 \hspace{1cm} Q=3
   step4: \hspace{1cm} R=22 \hspace{1cm} Q=4
   step5: \hspace{1cm} R=16 \hspace{1cm} Q=5
   step5: \hspace{1cm} R=10 \hspace{1cm} Q=6
   step5: \hspace{1cm} R=4 \hspace{1cm} Q=7

   (3 pts per step after the initialization)

   b) (10 pts) What does this algorithm accomplish? Explain the result of the algorithm (ie the relationship between R, Q, and D) WITHOUT simply explaining or restating the algorithm in part a.

   algorithm computes original value of R / D
   R ends w/ remainder
   Q ends w/ quotient (number of times D divides original value of R)
   \[46 / 6 = 7 \text{ remainder } 4\]