ASCII

American Standard Code for Information Interchange
ASCII

• ASCII is a character-encoding scheme
  – Originally developed for teleprinters in the 1960s
  – Now used to represent text in almost all computers and communications equipment
• We’ll see it again when we study communications standards such as RS-232
• It has code words that are 7 bits long*
  – Code words represent letters, numbers, punctuation symbols, etc
  – In addition there are some code words that are “control” characters
  – Some control characters were useful for teleprinters but don’t make much sense today

* There are (non standard) extensions to 8 bits; see for example http://www.ascii-code.com/
### ASCII Chart

- Codes 0 to $1f$ and $7f$ are non-printing, but represent special functions.
- Most common are: CR - carriage return, LF - line feed, HT - horizontal tab, BS - backspace, BEL - bell, NUL - null (ignored), ESC – escape.
- Code $20$ prints an empty space.

<table>
<thead>
<tr>
<th>x</th>
<th>0x</th>
<th>1x</th>
<th>2x</th>
<th>3x</th>
<th>4x</th>
<th>5x</th>
<th>6x</th>
<th>7x</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NUL</td>
<td>DLE</td>
<td>0</td>
<td>@</td>
<td>P</td>
<td>`</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SOH</td>
<td>DC1</td>
<td>1</td>
<td>A</td>
<td>Q</td>
<td>a</td>
<td>q</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>STX</td>
<td>DC2</td>
<td>2</td>
<td>B</td>
<td>R</td>
<td>b</td>
<td>r</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ETX</td>
<td>DC3</td>
<td>3</td>
<td>C</td>
<td>S</td>
<td>c</td>
<td>s</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>EOT</td>
<td>DC4</td>
<td>4</td>
<td>D</td>
<td>T</td>
<td>d</td>
<td>t</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ENQ</td>
<td>NAK</td>
<td>5</td>
<td>E</td>
<td>U</td>
<td>e</td>
<td>u</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ACK</td>
<td>SYN</td>
<td>6</td>
<td>F</td>
<td>V</td>
<td>f</td>
<td>v</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>BEL</td>
<td>ETB</td>
<td>7</td>
<td>G</td>
<td>W</td>
<td>g</td>
<td>w</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>BS</td>
<td>CAN</td>
<td>8</td>
<td>H</td>
<td>X</td>
<td>h</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>HT</td>
<td>EM</td>
<td>9</td>
<td>I</td>
<td>Y</td>
<td>i</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>LF</td>
<td>SUB</td>
<td>*</td>
<td>:</td>
<td>J</td>
<td>z</td>
<td>j</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>VT</td>
<td>ESC</td>
<td>+</td>
<td>;</td>
<td>K</td>
<td>[</td>
<td>k</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>FF</td>
<td>FS</td>
<td>,</td>
<td>&lt;</td>
<td>L</td>
<td>\</td>
<td>l</td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>CR</td>
<td>GS</td>
<td>-</td>
<td>=</td>
<td>M</td>
<td>]</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>SO</td>
<td>RS</td>
<td>.</td>
<td>&gt;</td>
<td>N</td>
<td>^</td>
<td>n</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>SI</td>
<td>US</td>
<td>/</td>
<td>?</td>
<td>O</td>
<td>_</td>
<td>o</td>
<td>DEL</td>
</tr>
</tbody>
</table>

**Note -** the CodeWarrior debugger displays the contents of memory as hex (left) and ASCII (right).
Write C function to do ...

• Write a function to return the value of an ASCII character if it represents a digit (return -1 otherwise)

• Approach:
  – The codes for digits 0..9 are in order in the table, from 0x30 to 0x39.
  – So if the character’s value is between 0x30 to 0x39, return its value minus 0x30.

```c
int getvalue(char c)
{
    if (c >= ‘0’ && c <= ‘9’)
        return (c - ‘0’);
    else
        return -1;
}
```

Recall that a character in single quotes has the value of the ASCII code for that character
Write C function to do ...

• Convert a lower case letter to upper case

• Code

/* Convert a lower case letter to upper case */
char lower2upper(char cx)
{
    if (cx >= 'a' && cx <= 'z')
        return (cx - ('a' - 'A'));
    else
        return cx;
}
Parity

• The most significant 8\textsuperscript{th} bit can be used as a parity bit for error checking
  – “Odd parity”: make the total # of 1’s odd
  – “Even parity”: make the total # of 1’s even

• Examples:
  – ASCII character “A”
    • Hex code $41$
    • 7-bit binary code 100 0001
  – Code words (8-bit):
    • Even parity 0100 0001
    • Odd parity 1100 0001

• If parity isn’t being used, the 8\textsuperscript{th} bit is ignored or just set to zero (call it “zero parity”)

Summary / Questions

• What is the difference between these C instructions?
  \[ x = 3; \]
  \[ x = '3'; \]

• Not all ASCII characters are printable ... how many printable characters are there?