Operations that modify lists
Two useful functions

- `ord(ch)`
  if `ch` is a single character string, this function returns the ASCII code for `ch`

- `chr(i)`
  returns a string of one character whose ASCII code is the integer `i`

What is ASCII?
It stands for the *American Standard Code for Information Interchange*. It assigns a number in the range 0..255 to every character that can be entered at the keyboard.
More on ASCII

- The numbers 0..31 are reserved for unprintable characters, e.g., the tab character ("\t"), the end of line character ("\n"), etc.
- 32 is the ASCII value of the space character (" ")
- 33..47 is used for some punctuation characters
- 48..57 is used for digits “0” through “9”
- 65..90 is used for upper case letters
- 97..122 is used for lower case letters
### ASCII Table

<table>
<thead>
<tr>
<th>Dec</th>
<th>Hx Oct</th>
<th>Char</th>
<th>Dec</th>
<th>Hx Oct</th>
<th>Html</th>
<th>Char</th>
<th>Dec</th>
<th>Hx Oct</th>
<th>Html</th>
<th>Char</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000</td>
<td>NUL</td>
<td>32</td>
<td>20 40</td>
<td>#32</td>
<td>Space</td>
<td>64</td>
<td>40 100</td>
<td>#64</td>
<td>@</td>
</tr>
<tr>
<td>1</td>
<td>001</td>
<td>SOH</td>
<td>33</td>
<td>21 041</td>
<td>#33</td>
<td>!</td>
<td>65</td>
<td>41 101</td>
<td>#65</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>002</td>
<td>STX</td>
<td>34</td>
<td>22 042</td>
<td>#34</td>
<td>&quot;</td>
<td>66</td>
<td>42 102</td>
<td>#66</td>
<td>B</td>
</tr>
<tr>
<td>3</td>
<td>003</td>
<td>ETX</td>
<td>35</td>
<td>23 043</td>
<td>#35</td>
<td>#</td>
<td>67</td>
<td>43 103</td>
<td>#67</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>004</td>
<td>EOT</td>
<td>36</td>
<td>24 044</td>
<td>#36</td>
<td>$</td>
<td>68</td>
<td>44 104</td>
<td>#68</td>
<td>D</td>
</tr>
<tr>
<td>5</td>
<td>005</td>
<td>ENQ</td>
<td>37</td>
<td>25 045</td>
<td>#37</td>
<td>%</td>
<td>69</td>
<td>45 105</td>
<td>#69</td>
<td>E</td>
</tr>
<tr>
<td>6</td>
<td>006</td>
<td>ACK</td>
<td>38</td>
<td>26 046</td>
<td>#38</td>
<td>&amp;</td>
<td>70</td>
<td>46 106</td>
<td>#70</td>
<td>F</td>
</tr>
<tr>
<td>7</td>
<td>007</td>
<td>BEL</td>
<td>39</td>
<td>27 047</td>
<td>#39</td>
<td>'</td>
<td>71</td>
<td>47 107</td>
<td>#71</td>
<td>G</td>
</tr>
<tr>
<td>8</td>
<td>010</td>
<td>BS</td>
<td>40</td>
<td>28 050</td>
<td>#40</td>
<td>(</td>
<td>72</td>
<td>48 110</td>
<td>#72</td>
<td>H</td>
</tr>
<tr>
<td>9</td>
<td>011</td>
<td>TAB</td>
<td>41</td>
<td>29 051</td>
<td>#41</td>
<td>)</td>
<td>73</td>
<td>49 111</td>
<td>#73</td>
<td>I</td>
</tr>
<tr>
<td>10</td>
<td>A12</td>
<td>LF</td>
<td>42</td>
<td>2A 052</td>
<td>#42</td>
<td>*</td>
<td>74</td>
<td>4A 112</td>
<td>#74</td>
<td>J</td>
</tr>
<tr>
<td>11</td>
<td>B13</td>
<td>VT</td>
<td>43</td>
<td>2B 053</td>
<td>#43</td>
<td>+</td>
<td>75</td>
<td>4B 113</td>
<td>#75</td>
<td>K</td>
</tr>
<tr>
<td>12</td>
<td>C14</td>
<td>FF</td>
<td>44</td>
<td>2C 054</td>
<td>#44</td>
<td>,</td>
<td>76</td>
<td>4C 114</td>
<td>#76</td>
<td>L</td>
</tr>
<tr>
<td>13</td>
<td>D15</td>
<td>CR</td>
<td>45</td>
<td>2D 055</td>
<td>#45</td>
<td>-</td>
<td>77</td>
<td>4D 115</td>
<td>#77</td>
<td>M</td>
</tr>
<tr>
<td>14</td>
<td>E16</td>
<td>SO</td>
<td>46</td>
<td>2E 056</td>
<td>#46</td>
<td>.</td>
<td>78</td>
<td>4E 116</td>
<td>#78</td>
<td>N</td>
</tr>
<tr>
<td>15</td>
<td>F17</td>
<td>SI</td>
<td>47</td>
<td>2F 057</td>
<td>#47</td>
<td>/</td>
<td>79</td>
<td>4F 117</td>
<td>#79</td>
<td>O</td>
</tr>
<tr>
<td>16</td>
<td>020</td>
<td>DLE</td>
<td>48</td>
<td>30 060</td>
<td>#48</td>
<td>0</td>
<td>80</td>
<td>50 120</td>
<td>#80</td>
<td>P</td>
</tr>
<tr>
<td>17</td>
<td>021</td>
<td>DC1</td>
<td>49</td>
<td>31 061</td>
<td>#49</td>
<td>1</td>
<td>81</td>
<td>51 121</td>
<td>#81</td>
<td>Q</td>
</tr>
<tr>
<td>18</td>
<td>022</td>
<td>DC2</td>
<td>50</td>
<td>32 062</td>
<td>#50</td>
<td>2</td>
<td>82</td>
<td>52 122</td>
<td>#82</td>
<td>R</td>
</tr>
<tr>
<td>19</td>
<td>023</td>
<td>DC3</td>
<td>51</td>
<td>33 063</td>
<td>#51</td>
<td>3</td>
<td>83</td>
<td>53 123</td>
<td>#83</td>
<td>S</td>
</tr>
<tr>
<td>20</td>
<td>024</td>
<td>DC4</td>
<td>52</td>
<td>34 064</td>
<td>#52</td>
<td>4</td>
<td>84</td>
<td>54 124</td>
<td>#84</td>
<td>T</td>
</tr>
<tr>
<td>21</td>
<td>025</td>
<td>NAK</td>
<td>53</td>
<td>35 065</td>
<td>#53</td>
<td>5</td>
<td>85</td>
<td>55 125</td>
<td>#85</td>
<td>U</td>
</tr>
<tr>
<td>22</td>
<td>026</td>
<td>SYN</td>
<td>54</td>
<td>36 066</td>
<td>#54</td>
<td>6</td>
<td>86</td>
<td>56 126</td>
<td>#86</td>
<td>V</td>
</tr>
<tr>
<td>23</td>
<td>027</td>
<td>ETB</td>
<td>55</td>
<td>37 067</td>
<td>#55</td>
<td>7</td>
<td>87</td>
<td>57 127</td>
<td>#87</td>
<td>W</td>
</tr>
<tr>
<td>24</td>
<td>030</td>
<td>CAN</td>
<td>56</td>
<td>38 070</td>
<td>#56</td>
<td>8</td>
<td>88</td>
<td>58 130</td>
<td>#88</td>
<td>X</td>
</tr>
<tr>
<td>25</td>
<td>031</td>
<td>EM</td>
<td>57</td>
<td>39 071</td>
<td>#57</td>
<td>9</td>
<td>89</td>
<td>59 131</td>
<td>#89</td>
<td>Y</td>
</tr>
<tr>
<td>26</td>
<td>032</td>
<td>SUB</td>
<td>58</td>
<td>3A 072</td>
<td>#58</td>
<td>:</td>
<td>90</td>
<td>5A 132</td>
<td>#90</td>
<td>Z</td>
</tr>
<tr>
<td>27</td>
<td>033</td>
<td>ESC</td>
<td>59</td>
<td>3B 073</td>
<td>#59</td>
<td>;</td>
<td>91</td>
<td>5B 133</td>
<td>#91</td>
<td>[</td>
</tr>
<tr>
<td>28</td>
<td>034</td>
<td>FS</td>
<td>60</td>
<td>3C 074</td>
<td>#60</td>
<td>&lt;</td>
<td>92</td>
<td>5C 134</td>
<td>#92</td>
<td>\</td>
</tr>
<tr>
<td>29</td>
<td>035</td>
<td>GS</td>
<td>61</td>
<td>3D 075</td>
<td>#61</td>
<td>=</td>
<td>93</td>
<td>5D 135</td>
<td>#93</td>
<td>]</td>
</tr>
<tr>
<td>30</td>
<td>036</td>
<td>RS</td>
<td>62</td>
<td>3E 076</td>
<td>#62</td>
<td>&gt;</td>
<td>94</td>
<td>5E 136</td>
<td>#94</td>
<td>^</td>
</tr>
<tr>
<td>31</td>
<td>037</td>
<td>US</td>
<td>63</td>
<td>3F 077</td>
<td>#63</td>
<td>?</td>
<td>95</td>
<td>5F 137</td>
<td>#95</td>
<td>_</td>
</tr>
</tbody>
</table>
Some examples of `chr` and `ord` in action

```python
>>> ord("a")
97
>>> chr(97)
'a'
>>> ord(" ")
32
>>> ord("0")
48
>>> chr(48)
'o'
>>> chr(49)
'1'
>>> ord("A")
65
>>> ord("B")
66
```
How are these functions useful?

- Because of the fact that all the upper case letters occur consecutively in the ASCII table, the expression $\text{ord}(\text{ch}) - \text{ord}(\text{"A"})$ has value 0 for $\text{ch} = \text{"A"}$, value 1 for $\text{ch} = \text{"B"}$, has value 2 for $\text{ch} = \text{"C"}$, etc.

- Similarly, $\text{ord}(\text{ch}) - \text{ord}(\text{"a"})$ has value 0 for $\text{ch} = \text{"a"}$, has value 1 for $\text{ch} = \text{"b"}$, has value 2 for $\text{ch} = \text{"c"}$, etc.
A program to count letter frequencies

```python
f = open("war.txt")
L = [0]*26
s = f.read()
for ch in s:
    if ch.isupper():
        L[ord(ch)-ord("A")] = L[ord(ch)-ord("A")] + 1
    elif ch.islower():
        L[ord(ch)-ord("a")] = L[ord(ch)-ord("a")] + 1
print L
```

Notice how `ord(ch)-ord("A")` and `ord(ch)-ord("a")` are used to index into the list `L`. 
Another example

- The `ord` and `chr` functions can be used to perform Caeser’s Cipher (Problem 3, HW 7).

- Try this: `chr(ord("a") + 4)`

- What does this expression evaluate to?
Lists and strings also have important differences

- In Python some data types are *mutable*, i.e., they can be modified in place.
- Of the data types we have seen so far, e.g., `int`, `long`, `float`, `bool`, `str`, and `list`, only list is mutable.

**Example:**

```python
>>> L = [3, 4, 5]
>>> type(L)
<type 'list'>
>>> L[0] = 8
>>> L
[8, 4, 5]

>>> s = "hello"
>>> type(s)
<type 'str'>
>>> s[0]
'h'
>>> s[0] = "t"
>>> s[0] = "t"
Traceback (most recent call last):
  File "<string>", line 1, in <fragment>
TypeError: 'str' object does not support item assignment
```

By doing an assignment to `L[0]`, we have replaced the first element in the list `L`.

We can examine elements in the string `s` in a similar manner, but we cannot assign anything to `s[0]`.
Example:

```python
>>> id(L)
12494888
>>> L[0] = 11
>>> id(L)
12494888
```

Recall that we said the id function when applied to a variable name, returns the location pointed to by that variable. Notice how the location of L does not change as a result of replacing the first element by something else.

```python
>>> n = 10
>>> id(n)
10022540
>>> n = 12
>>> id(n)
10022516
```

An assignment to an int variable does not modify the variable “in place.” The variable ends up pointing to another location.
List operations that modify a list “in place”

Replacing single elements or slices of lists
- \( L[0] = 10, \)
- \( L[3:5] = [10, 12], \)
- \( L[3:10:2] = [12, 14, 16, 18] \)

Deleting a list or its parts
- \( \text{del } L \)
- \( \text{del } L[3] \)
- \( \text{del } L[3:5] \)
- \( \text{del } L[3:10:2] \)
Try and understand all of these operations.

- `L.append("hi")`
- `L.extend(["good"])`
- `L.insert(4, "bye")`
- `L.pop(), L.pop(4)`
- `L.remove("hello")`

None of these work on strings.

And here are the last two:

- `L.reverse(), L.sort()`