Please turn in answers to the following questions on Monday, in class.

1. In the following table, the value **alice** is used to produce the value of **bob** using an assignment making use of indexing. Fill in the blanks.

<table>
<thead>
<tr>
<th>value of alice</th>
<th>value of bob</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>e.g. &quot;Hello, world&quot;</td>
<td>&quot;Hello&quot;</td>
<td>bob = alice[0:5]</td>
</tr>
<tr>
<td>a. &quot;Williams&quot;</td>
<td>&quot;W&quot;</td>
<td>bob =</td>
</tr>
<tr>
<td>b. &quot;Amherst&quot;</td>
<td>&quot;her&quot;</td>
<td>bob =</td>
</tr>
<tr>
<td>c. &quot;Ephriam&quot;</td>
<td></td>
<td>bob = alice[-3:]</td>
</tr>
<tr>
<td>d. &quot;piff poof paff&quot;</td>
<td></td>
<td>bob = alice[10:30]</td>
</tr>
<tr>
<td>e. &quot;ornation&quot;</td>
<td>&quot;onto&quot;</td>
<td>bob =</td>
</tr>
<tr>
<td>f. &quot;chainlet&quot;</td>
<td>&quot;hilt&quot;</td>
<td>bob =</td>
</tr>
<tr>
<td>g. &quot;tinker&quot;</td>
<td>&quot;reknit&quot;</td>
<td>bob =</td>
</tr>
<tr>
<td>h. &quot;approach&quot;</td>
<td>&quot;harp&quot;</td>
<td>bob =</td>
</tr>
<tr>
<td>i. &quot;blueness&quot;</td>
<td>&quot;snub&quot;</td>
<td>bob =</td>
</tr>
<tr>
<td>j.</td>
<td>&quot;python&quot;</td>
<td>bob = alice[:3]+alice[3:]</td>
</tr>
</tbody>
</table>
2. Rewrite the following expressions in a simpler or more elegant way (to the right of the code):

(a) # assume l is a list and x is an object
    l += [ x ]

(b) # assume u and v are strings
    s = ""
    for c in u:
        s = s + c
    for c in v:
        s += c

(c) def nstars(n):
    """Returns a string of n stars.""
    # (Hint: multiplication)
    assert n >= 0
    s = ""
    for i in range(n):
        s += "*"
    return s

(d) # assume a and b are boolean values
    result = True if (a or b) else False

(e) # assume vowel is an integer from 0 to 4
    if vowel == 0:
        c = 'a'
    elif vowel == 1:
        c = 'e'
    elif vowel == 2:
        c = 'i'
    elif vowel == 3:
        c = 'o'
    else:
        c = 'u'