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;Pixel&quot;</td>
<td>&quot;P&quot;</td>
<td>bob =</td>
</tr>
<tr>
<td>b. &quot;February&quot;</td>
<td>&quot;bru&quot;</td>
<td>bob =</td>
</tr>
<tr>
<td>c. &quot;Ephraim&quot;</td>
<td></td>
<td>bob = alice[-3:]</td>
</tr>
<tr>
<td>d. &quot;grace hopper&quot;</td>
<td></td>
<td>bob = alice[6:30]</td>
</tr>
<tr>
<td>e. &quot;ornation&quot;</td>
<td>&quot;onto&quot;</td>
<td>bob =</td>
</tr>
<tr>
<td>g. &quot;desserts&quot;</td>
<td>&quot;stressed&quot;</td>
<td>bob =</td>
</tr>
<tr>
<td>h. &quot;blueness&quot;</td>
<td>&quot;snub&quot;</td>
<td>bob =</td>
</tr>
<tr>
<td>i. &quot;tapia&quot;</td>
<td></td>
<td>bob = alice[:3]+alice[3:]</td>
</tr>
</tbody>
</table>

For the next few questions, we will think about the implications of working with mutable and immutable objects in Python. Beside each # prints:, indicate what is printed.

2a. hopper = [ 20, 21, 22 ] # some upcoming years
tapia = [ 20, 21, 22 ] # some classes of students
print(hopper is tapia) # prints:
hopper.append(23) ###
print(hopper) # prints:
print(tapia) # prints:

Explain what is happening to hopper and tapia (if anything) on the statement marked ###.
2b.  hopper = tapia = [23, 20, 21, 22]  # some upcoming graduating years
print(hopper is tapia)  # prints:
  sorted(hopper)  ###
print(hopper)  # prints:
print(tapia)  # prints:
tapia.sort()  ###
print(hopper)  # prints:
print(tapia)  # prints:

Explain what is happening on the statements marked ###.

2c.  hopper = tapia = 'upcoming years'
print(hopper is tapia)  # prints:
tapia.replace('upcoming', 'graduating') ###
print(tapia)  # prints:
print(hopper)  # prints:
hopper = hopper.replace('upcoming','pre-alumni') ###
print(hopper)  # prints:
print(tapia)  # prints:

Explain what is happening on the statements marked ###.

2d.  nestedList = [[1, 2], [3, 4]]  # list of lists
nestedList.append(nestedList[1]) ###
nestedList[2][1] = 6 ###
print(nestedList)  # prints:

Explain what is happening on the statements marked ###.

*