Put True (T) or False (F) in every cell, based on characteristics of each type.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tuple</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>namedtuple</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2

```python
x = [1, 2, 3]
y = [1, 2, 3]
z = x
z.append(4)
```

3

```python
nums1 = [1,2]
nums2 = nums1
x = nums2.pop(1)
```

4

```python
x = [1, 2]
y = [3]
z = x + y
y.append(4)
```

5

```python
people = {"alice":30, "bob":25}
x = people
y = people["bob"]
x["alice"] = 31
y = 26
```

6

```python
def f(items):
    return items.pop(0)
nums = [1,2,3]
nums.append(f(nums))
```
Remember to import copy for these in Python Tutor!

7. 
```
x = [2, 1]
y = copy.copy(y)
y.sort()
```

```
def biggest(items):
    items = copy.copy(items)
    items.sort()
    return items[-1]
```

```
ums = [3, 9, 6]
x = biggest(nums)
```

8. 
```
team1 = [
    {"name":"A", "age":7}
]
team2 = copy.copy(team1)
team2.append(
    {"name":"B", "age":9}
)
team2[0]["age"] = 8
x = team1[0]["age"]
```

9. 
```
Same as above, but with
```
```
copy.deepcopy(...) instead
of copy.copy(...).
```

10. 
```
orig = [1,[2,[3,4]]]
x = orig
y = copy.copy(orig)
z = copy.deepcopy(orig)
```