1) How are lists like strings?

a) Lists have length:

   i) What is the result of the following code?

   ```python
   a = ['soccer', 6, 8.32]
   print(len(a))
   ```

   ii) What is the result of the following code?

   ```python
   b = ['double', 'triple', [8, 10, 12], [], 4.5]
   print(len(b))
   ```

b) Lists have order and can be indexed.

   i) What is the result of the following code?

   ```python
   print(b[1])
   ```

   ii) What is the result of the following code?

   ```python
   print(b[2][1])
   ```
c) List membership can be tested.

i) What is the result of the following code?

```python
a = ["soccer", 6, 8.32]
print(6 in a)
```

ii) What is the result of the following code?

```python
b = ["double", "triple", [8, 10, 12], [], 4.5]
print(10 in b)
```

d) Lists can be concatenated.

What is the result of the following code?

```python
(1) c = [22, 31, 43]
(2) d = [18, 19, 20]
(3) print(c + d)
```

e) Lists can be multiplied.

What is the result of the following code?

```python
(1) print(d * 2 + c)
```
f) Lists can be sliced.

What is the result of the following code?

(1) \( e = [\text{"poppy"}, \text{"rose"}, \text{"petunia"}, \text{"daisy"}] \)
(2) \( f = [8, 10, [\text{\'a\'}, \text{\'b\'}, \text{\'c\'}], \text{"twenty"}, []] \)
(3) \( g = e[1:2] * 2 + f[2:] \)
(4) \( \text{print}(g) \)

1) How are lists not like strings?

a) Lists are mutable.

i) What does the following code do?

\[
\begin{align*}
\text{names} & = [\text{"charlie"}, \text{"twenty"}, \text{"keni"}, \text{"lara"}] \\
\text{print(names[2])} \\
\text{names[1]} & = \text{"meg"} \\
\text{print(names)}
\end{align*}
\]

ii) What does the following code do?

\[
\begin{align*}
\text{list\_a} & = [4, 5, 6, 7, 8] \\
\text{list\_b} & = [3, 4, 5] \\
\text{list\_a}[2:2] & = \text{list\_b} \\
\text{print(list\_a)}
\end{align*}
\]
iii) What does the following code do?

```python
list_a = [4, 5, 6, 7, 8]
del list_a[2:4]
print(list_a)
```

b) What does the following code do?

```python
list_a = [6, 7, 8]
list_b = [6, 7, 8]
print(list_a is list_b)
print(list_a == list_b)
```
c) What happens in the following code?

```python
list_a = [6, 7, 8]
list_b = [6, 7, 8]
print(list_a is list_b)
print(list_a == list_b)

list_a = list_b
print(list_a is list_b)
print(list_a == list_b)

list_b[1] = 200
print(list_a)
```

d) What can we do if we want `list_b` to be a copy of `list_a` but be a separate list so that when we modify `list_b`, `list_a` does not change?
2) Using Lists

a. Write a function called `itemsLessThan` that takes a list and a value and returns a new list with only the items that were less than the given value. This function should not modify the original list.