1. What exactly is []?

Ans: Empty array

2. In a list of values stored in a variable called spam, how would you assign the value 'hello' as the third value? (Assume [2, 4, 6, 8, 10] are in spam.)

Ans: spam[2] = 'hello'

Let's pretend the spam includes the list ['a', 'b', 'c', 'd'] for the next three queries.

3. What is the value of spam[int(int('3' \* 2) / 11)]?

Ans: spam[3] = 'd'

4. What is the value of spam[-1]?

Ans: spam[-1] = 'd'

5. What is the value of spam[:2]?

Ans: spam[:2] = ['a','b']

Let's pretend bacon has the list [3.14, 'cat,' 11, 'cat,' True] for the next three questions.

6. What is the value of bacon.index('cat')?

Ans: 1

7. How does bacon.append(99) change the look of the list value in bacon?

Ans: [3.14, 'cat,' 11, 'cat', True,99]

8. How does bacon.remove('cat') change the look of the list in bacon?

Ans: [3.14, 11, 'cat,' True]

9. What are the list concatenation and list replication operators?

Ans: list concatenation: + ; list replication: \*

10. What is difference between the list methods append() and insert()?

Ans: append() adds value to the end of the list and insert() can add anywhere based on the index provided

11. What are the two methods for removing items from a list?

Ans: del and remove

12. Describe how list values and string values are identical.

Ans: Both list values and string values can be indexed, sliced, concatenated, iterated over, and have a length, making them similar in terms of their basic manipulations and behaviors.

13. What's the difference between tuples and lists?

Ans: Lists are mutable and represented in [] while tuples are immutable and represented in ()

14. How do you type a tuple value that only contains the integer 42?

Ans: (42,)

15. How do you get a list value's tuple form? How do you get a tuple value's list form?

Ans: tuple() and list() to get their respective values.

16. Variables that "contain" list values are not necessarily lists themselves. Instead, what do they contain?

Ans: contains reference to list values

17. How do you distinguish between copy.copy() and copy.deepcopy()?

Ans: copy.copy() creates shallow copy of object while copy.deepcopy() includes all nested objects recursively.