1. What exactly is []?

A: Empty list.

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.)

A: spam.insert(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)]?

A: ‘d’

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

A:’d’

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

A:[‘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')?

A: 1

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

A: [3.14, ‘cat’, 11, ‘cat’, ‘True’]

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

A: [3.14, 11, ‘cat’, ‘True’]

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

A: + and \* are the list concatenation and replication operators respectively.

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

A: append() method adds element at the end of the list. In Insert() method we need to specify the index where we want the element.

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

A: remove() and del methods.

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

A: Both are ordered data types, can be passed in len() function, can be concatenated, have indexes and can be replicated.

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

A: Tuples are immutable, whereas lists are mutable.

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

A: tup =(42)

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

A: Using tuple() and list() functions respectively.

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

A: They contain reference to list values.

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

A: copy.copy() does not do recursive copying. copy.deepcopy() does recursive copying and creates a completely independent object.