1. What exactly is []?

**An 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.)

**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)]?

**‘d’**

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

**‘d’**

5. What is the value of 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')?

**1**

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

**[3.14, ‘cat’, 11, ‘cat’, True, 99]**

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

**[3.14, 11, ‘cat’, True, 99]**

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

**List concatenation is +**

**List replication is \***

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

**Append method inserts at index last of the list, whereas insert method inserts the value at any index.**

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

**remove(), pop()**

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

**List contains sequence of elements, whereas string contains sequence of characters.**

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

**Lists are mutable, whereas tuples are immutable.**

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

**(42,)**

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

**List to tuple 🡪 tuple(list\_name)**

**Tuple to list 🡪 list(tuple\_name)**

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

**They contain reference to list values.**

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

**copy.copy() create reference to original object. If you change copied object then change happens in the original object too.**

**deepcopy() creates new object and does real copying of original object to new one. Changing new deepcopied object doesn't affect original object**.