1. What exactly is []?

**[] defines 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.)

**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)]?

**‘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, True]**

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

**list concatenation +**

**list replication \***

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

**Append inserts at last index of the list**

**Insert function allows us to insert number at the specified index**

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

**Pop()**

**Remove()**

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

**Both allows indexing operations**

**concatenation and replication possible in both**

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

**Tuples are set of data that are immutable represented by open brackets**

**List are set of data that are mutable represented by closed brackets**

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

**(42,0)**

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

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

**Variables will contain references to list values rather than list values themselve**

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

**A shallow copy constructs a new compound object and then (to the extent possible) inserts references into it to the objects found in the original.**

**A deep copy constructs a new compound object and then, recursively, inserts copies into it of the objects found in the original**