1. What exactly is []?

**Ans:** 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.)

**Ans:** a.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)]?

**Ans:** ‘d’

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

**Ans:** ‘d’

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

**Ans:** [‘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 operator is +, while list replication operator is \*.

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

**Ans:** Append will append object in last of list, however insert can insert object anywhere in list using provided index.

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

**Ans:** remove and pop.

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

**Ans:** for both of the objects values can be retrieved using index subscriptions, also both can be iterate in the same way using for loop.

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

**Ans:** tuples are immutable, while lists are mutable.

Tuples values can be assigned during at the time of declaration only, however list can be amended later at run time also.

Only count and index are 2 supported functions for tuples, while lists supports multiple functions for manipulation.

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

**Ans:** a=(42)

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

**Ans:** iterate tuple or list using for loop, and for every object check the type if it is tuple or list type and we can get the list/tuple values from there.

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

**Ans:** address of list object

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

**Ans:**copy.copy() will keep the reference only of old object and not create a new object any hange in one will reflect in other, however copy.deepcopy() will create a new object any change in other will not reflect in other.