1. What exactly is []?

**→ [] is a list: A mutable collection of values, usually (but not necessarily) of the same type.**

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, 'cat', True, 99]`

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

**→ The operator for list concatenation is +, while the operator for replication is ***

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

**→ The difference between the two methods is that . append() adds an item to the end of a list, whereas . insert() inserts and item in a specified position in the list.**

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

**→The methods are remove(), pop() and clear(). It helps to remove the very first given element matching from the list. The pop() method removes an element from the list based on the index given. The clear() method will remove all the elements present in the list.**

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

**→ The similarity between Lists and Strings in Python is that both are sequences. The differences between them are that firstly, Lists are mutable but Strings are immutable. Secondly, elements of a list can be of different types whereas a String only contains characters that are all of String type.**

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

**→ The key difference between tuples and lists is that while tuples are immutable objects, lists are mutable. This means tuples cannot be changed while lists can be modified. Tuples are also more memory efficient than the lists.**

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?

**→Using the tuple() built-in function**

**An iterable can be passed as an input to the tuple () function, which will convert it to a tuple object.**

**→To convert a tuple into list in Python, call list() built in function and pass the tuple as argument to the function. list() returns a new list generated from the items of the given tuple.**

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 themselves. But for strings and integer values, variables simply contain the string or integer value.**

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

**→A shallow copy creates a new compound object and then adds a reference to the object found in the original. A deep copy creates a new compound object and then adds a reference to the object found in the original. We can copy arbitrary objects (including custom classes) with the copy module**.