1. What exactly is []?

Ans: it represents list. It s collection of values.

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: 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)]?

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: the list concatenation -> **+** and list replication operators -> *****

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

Ans: append() add element at the last position of list whereas insert() add element at the specified position of list(e.g., lst.insert(elem,pos))

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

Ans: remove(element) → remove first occurrence of element in a list.

Pop(index) → remove element at specified index.

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

Ans: Lists are similar to strings, which are ordered collections of characters, except that the elements of a list can have any type.

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

Ans: tuples are immutable as opposed to lists which are mutable i.e., after initialization of tuple modification are not possible but in list we can modify it after initialization also.

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

Ans: t = (42)

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

Ans: t = tuple([1,2,3,4]) → tuple

L = list((1,2,3,4)) → list

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

Ans: a variable that "contains" a list value typically refers to a variable that holds a reference to a list. This means that the variable does not contain the list itself, but rather it contains a value that points to the location in memory where the list is stored.

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

Ans: The main difference between these two methods is that copy() creates a shallow copy of an object, while deepcopy() creates a deep copy of an object.

A shallow copy of an object creates a new object with the same value as the original object, but it only copies the object's references to its sub-objects, rather than creating new sub-objects with their own copies. This means that if the original object contains mutable sub-objects, the shallow copy will still refer to the original sub-objects, rather than creating new copies of them.

In contrast, a deep copy of an object creates a new object with its own copy of the original object's sub-objects, rather than just copying references to the original sub-objects. This means that a deep copy will create new copies of all mutable sub-objects, rather than referring to the original sub-objects.