1. What exactly is []?

**Ans**: The empty list value, which is a list value that contains no items. This is similar to how '' is the empty string value.

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:** It will delete the first occurrence of string ‘cat’, after deleting the list shown as below

[3.14, 11, 'cat', True]

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

**Ans:** The operator for list concatenation is +, while the operator for replication is \*.

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

**Ans:** The difference is that with append, you just add a new entry at the end of the list. With insert(position, new\_entry) you can create a new entry exactly in the position you want.

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

**Ans:** The methods are remove() and pop(). Remove() method 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.

Ex: list = [3.14, 'cat', 123, True, 99, 'a', 'b', 'c']

List.pop(1) - [3.14, 123, True, 99, 'a', 'b', 'c']

List.remove(123) - [3.14,’cat’, True, 99, 'a', 'b', 'c']

We also have del statement and clear() to delete the elements from the list.

Del will remove elements based on index.

Clear() will remove all elements from list.

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

**Ans:** Both lists and strings can be passed to **len**(), have indexes and slices, be used in **for** loops, be concatenated or replicated, and be used with the **in** and **not** in operators.

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

**Ans:** Lists are mutable; they can have values added, removed, or changed. Tuples are immutable; they cannot be changed at all. Also, tuples are written using parentheses, ( and ), while lists use the square brackets, [ and ].

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

**Ans:** (42,) (The trailing comma is mandatory.)

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

**Ans:** Tuple = (1,2,3)

Converting into list from tuple

list = list(tuple) = [1,2,3]

we have a list - List = [4,5,6]

Converting into tuple from list

Tp = tuple(list) = (4,5,6)

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

**Ans:** They contain references to list values.

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

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

If we make in copied list then it will also reflect in original list.

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

If wont reflect in original list if we make any changes in copied list.