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**: spam.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**: ‘3’\*2🡪’int(‘33’)🡪33/11🡪index(3) 🡪 ‘d’

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

**Ans**: ‘d’

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

**Ans**: ['a', 'c']

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, 99]

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

Ans: Concatenation: +

Replication: \*

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

Ans: Append will add the value at the end of the list.

By using insert we can add any value at any index.

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

Ans : by using remove we can remove particular item in the list and by using clear method we can

Remove entire items in the list.

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

Ans: The similarity between Lists and Strings in Python is that both are sequences.

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

Ans: Tuples are immutable objects and lists are mutable objects.

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: list to tuple

a=[1,2,3,4]

b=tuple(a)

tuple to list

a=(1,2,3,4)

b=list(a)

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

Ans: string or integer value.

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

Ans: copy() create reference to original object. If you change copied object - you change the original object. . deepcopy() creates new object and does real copying of original object to new one. Changing new deepcopied object doesn't affect original object.