1. What exactly is []?

**-> [] braces represents the list structure, all the list items are inside the brackets seperated with comma.**

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.)

->With inbuilt function **spam.insert(3, '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)]?

**->3**

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]**

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

**-> +, \***

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

**->append adds the element after the last element while insert adds element at specific location**

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

**->remove(), pop()**

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

**->Both are sequential**

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

**->Tuples are immutable while list is mutable. Both list and tuples are represented differently.**

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?

**->By typecasting : list(tuple\_) tuple(list\_)**

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

**->Contains references or base address of the list**.

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

**In copy.copy if any changes made in the list it will affect original list as it is nothing but a new object referenced to the old object.**

**While copy.deepcopy() creates a new object and assigns the values of old to the new object, any changes made in new list will not affect older list.**