1. What exactly is []?

**Ans 1 : []**  is an empty 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 2 :** 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 3 : d**

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

**Ans 4 : spam[-1]** indicates last value in the list that is **d**

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

**Ans 5 : spam[:2]** gives first two values of the list **a** and  **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 6 :** index of cat in the list : **1**

7. How does bacon.append(99) change the look of the list value in bacon?

**Ans 7 : [**3.14, 'cat,' 11, 'cat,' True,**99]**

8. How does bacon.remove('cat') change the look of the list in bacon?

**Ans 8 : [3.14, 11, 'cat,' True] :** removes the ‘cat’ from the very first index of it occurence.

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

**Ans 9 : List concatenation :**  Add two or more list , operator is +.

Eg : list 1 = [12,45,87], list2 = [49,23]

list1 + list2 = [2,45,87,49,23]

**List replication :** Duplicates the data of the list number of time defined after \*.

List1 = [45,56]

List1 \* 3 = [45,56,45,56,45,56]

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

**Ans 10 :** append() : add the element at the last location of the list.

Insert() : we can insert element anywhere in the list by specifying its location.

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

**Ans 11 : list.remove(data) :** Removes the first occurence of the data in the list

**list.pop() :** if no index is defined pop removes the last element

**list.pop(2):** if index is passed, removes the element at given index from the list.

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

**Ans 12 : list values**  and **string values** both are sequential collection of data.

List can hold data of any type where string holds only character of string data type.

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

**Ans 13 :**  Tuples are immutable and lists are mutable that is, once tuple is created its data cannot be altered where list data can be altered at any time.

Tuple = (12,56,98,’yash’)

list = [12,56,98,’yash’]

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

**Ans 14 :** For a tuple with single data we need to add , at the end of the data

tuple = (12,) . If , is not given, it will be considered as a integer data.

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

**Ans 15 : list to tuple =** list(tuple \_name)

**tuple to list =** tuple(list\_name)

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

**Ans 16 : Python variable** contain references for mutable data type and values itself for immutable data type.

List is a mutable data type, so a variable stores a reference for the list and not the values.

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

**Ans 17 : copy.copy( ) :** Copies the data of the original object. If any changes are made in the **copied** object, those are reflected in the original object.

Whereas in **copy.deepcopy() :** Deep Copies the data of the object. If any changes are made in the **copied object,** that are not reflected in the original object.