1. What exactly is []?

In Python, the **SQUARE BRACKETS []** reflect list comprehension used to make lists. The square brackets are used to tell Python to save the values within the brackets as a **LIST**. We may also use [] to make an empty list of no 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.)

**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)]?

I’m not clear by the word includes, hence giving two answers considering:

1. If spam = ['a', 'b', 'c', ‘d’] ==> **‘d’**

2. If spam = [2, 4, 6, 8, 10, ['a', 'b', 'c', ‘d’]] ==> 8

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

I’m not clear by the word includes, hence giving two answers considering:

1. If spam = ['a', 'b', 'c', ‘d’] ==> ‘d'

2. If spam = [2, 4, 6, 8, 10, ['a', 'b', 'c', ‘d’]] ==> ['a', 'b', 'c', ‘d’]

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

I’m not clear by the word includes, hence giving two answers considering:

1. If spam = ['a', 'b', 'c', ‘d’] ==> [‘a’, ’b’]

2. If spam = [2, 4, 6, 8, 10, ['a', 'b', 'c', ‘d’]] ==> [2, 4]

Let's pretend bacon has the list [3.14, 'cat,' 11, 'cat,' True] for the next three questions.

**I believe [3.14, 'cat,' 11, 'cat,' True] as mentioned above is a typo as it would be an invalid syntax in python. Hence considering**

**bacon = [3.14, ‘cat', 11, ‘cat', True]**

6. What is the value of bacon.index(‘cat')?

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?

bacon.remove(‘cat') will remove **first occurrence of cat** in the list ‘bacon’.

Hence, the result would be now : **[3.14, 11, 'cat', True, 99]**

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

* + The 'List Concatenation Operator' is the arithmetic sum operator, **+** that adds value to the existing list.
  + The 'List replication Operator' is the arithmetic multiplication operator, \*that creates a new list by repeating the values from the existing list.

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

* + list.append(x) adds x to the **tail (last position) of the list.**
  + list.insert(index,x) inserts the element at the **defined index position.**

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

* + **remove()** : list.remove(x) deletes the first instance of x from the list.
  + **del** operator : del list(index) deletes the element at the designated index. It can be used to remove multiple items in a single go by mentioning the index range to remove.

In addition to the above two methods, we can use **pop()** method to remove one element at a time. By default, it removes the last element of the list and returns it unless the index of the element to be removed.

bacon.pop() ==> removes last element in bacon and returns it,

bacon.pop(2) ==> removes the element in index 2 of the list bacon and returns it.

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

Lists contain a set of elements of various data types, numbered from 0 to (length of the list – 1). Strings are made up of a series of characters numbered from 0 to (length of the list – 1).

Thus, for strings and lists, the element calls using list[index] and a particular character from the string using str[index] are equivalent. **Both sub-setting and slicing operations that apply to a list apply to strings as well**.

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

Tuples are similar to lists in which they contain several items of different data types. Tuples, on the other hand, are **IMMUTABLE**, whereas **lists are MUTABLE**. The data contained inside tuples cannot be changed.

Tuples are denoted by parentheses (), while lists are denoted by square brackets [].

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

**tuple=(42, )**

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

The built-in methods list() and tuple() can be used to interconvert tuples and lists.

* + tuple(list) converts a list to a tuple
  + list(tuple) converts the tuple to a list.

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

**The memory address of the list object is usually stored in the list variable.**

When a new list is created and allocated to a list variable, a new list object is created in memory, with the memory address of the list object is stored in the list variable name.

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

* + Deep copy is a process in which the copying process is repeated indefinitely. It involves first creating a new collection object and then recursively populating it with copies of the original's child items. A copy of an object is replicated in another object in the case of deep copy. It means that any modifications made to a copy of an item are not reflected in the original. This is accomplished in Python by using the “deepcopy()” function.
  + A shallow copy is created by creating a new collection object and then populating it with references to the original's child objects. Since the copying process does not recurse, no copies of the child objects are made. In the case of a shallow copy, an object's relation is copied into another object. This implies that any modifications made to a copy of an item are reflected in the original. This is accomplished in Python by using the “copy()” function.