1. What exactly is []?

Ans:- [] i.e. Square brackets are used to create a list in python.  
 [] denotes 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:- We can use the insert method with the list as shown below.

spam=[2,4,6,8,10]

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:- Value will be “d”.

Explanation:- ‘3’ \* 2 will be evaluated as 33 and thus casting it to int and dividing by 11 will be 3. Hence, the value at index 3 is “d”.

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

Ans:- Since, Lists have -1 as the index for last value, the value of spam[-1] will be “d”.

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

Ans:- spam[:2] will give the values till index 1 and thus output will be [‘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:- It will provide the index at which value ‘cat’ is present i.e. “1”.

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

Ans:- append will append or add the given value 99 after the last element list and hence the look of the list would be [3.14, 'cat,' 11, 'cat,' True,99].

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

Ans:- bacon.remove(‘cat’) will remove the value cat from the list and the list would look like  
 [3.14, 'cat,' 11, True]

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

Ans:- “+” and “\*” are list concatenation and replication operators respectively.  
 + operator will add or concatenate a value to the list.  
 \* will create a replica of the existing values and add the same to the list.

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

Ans:- append() will add the new value after the last value of the current value whereas with insert we can add new values at a specific index of the list.

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

Ans:- remove() and pop() are the two methods for removing items from the list.

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

Ans:- 1. List values and string values are accessed using an index.  
 2. Both list and string are sequences.  
 3. Both can use the predefined len() method to know the exact number of elements.

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

Ans:- 1.Tuples are immutable whereas lists are mutable.  
 2.Tuples are defined using () whereas lists are defined using [].

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

Ans:- To type a tuple value that only contains the integer 42, we can write it as  
 exampletuple=(42,)

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

Ans:- 1. To get a list values’s tuple form, we can use the tuple() function.

2. To get a tuple value’s list form, we can use the list() function.

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

Ans:- Variables that contain list values are not necessarily lists themselves instead they contain the reference to the list values.

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

Ans:- In the case of shallow copy i.e. copy.copy(), a reference of an object is copied into another object whereas in Deep copy i.e. copy.deepcopy(), a copy of the object is copied into another object. and hence any changes made to even the innermost object is not reflected in the original list.

For e.g:-

list1 = [[1,2,3],[4,5,7]]

list2=copy.copy(list2)

list2[0][1] = 100

list1

The output for the above code would be [[1,100,3],[4,5,7]] as though the copy has been created for the items the objects inside of it are still refering to the original values and not exactly copied.

if we had use deepcopy instead of copy then print original list i.e.

list2 = copy.deepcopy(list1)

list2[0][1]=100

list1

The output would be [[1,2,3],[4,5,7]] that is the value hasn’t changed as for deepcopy the original collection has been copied and the changes are made to the objects in the collection and hence no change in the original list or items.