1. What exactly is []? 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.)

spam[2] = "hello"

OR

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)]? d

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, 99]

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

for list concatenation -> +

for list replication -> \*

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

append() – append a value in the list means insert a value at the end.

insert() – insert a value in list at specified position.

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

Pop -> Remove element from specified index.

Remove -> Remove the specified value from list.

Clear -> Remove all the elements from list.

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

They both are Sequence type.

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

List – Mutable

Tuple – Immutable

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?

For list from tuple use the method - list(tuple)

For tuple from list use the method - tuple(list)

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

Variables will contain references to list values rather than list values themselves.

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

copy.copy() => It is shallow copy. It creates a new object which stores the reference of the original elements. It doesn't create a copy of nested objects, instead it just copies the reference of nested objects.

copy.deepcopy() => It is deep copy. creates a new object and recursively adds the copies of nested objects present in the original elements.