What exactly is []?

It represents and 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.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)]?

‘d’ spam[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?

Concatenation – Using + operatore, extend( function)

Replication – by assigning to different variable or by copy() function

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

Append adds the element to the last index of the list while the insert can add the element to specific index in the list

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

Remove() and pop()

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

The elements of both list and string can be accessed by their index values and can be iterated through

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

Lists are mutable while tuples are not

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?

**Unable to understand question**

a = [1,2,3,4,5]

b = (6,7,8,9,10)

a\_tuple = tuple(a)

b\_list = list(b)

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

**Unable to understand question**

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

Lists created with .copy() is still connected with the original list so modifiying the original will change the newly copied list

But deepcopy creates a separate clone that does’nt change when the original is modified