1. What exactly is []?

 [] is a 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.)

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)]?

'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, True, 99]

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

The operator for list concatenation is +, while the operator for replication is \*

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

append() adds an item to the end of a list, whereas . insert() inserts and item in a specified position 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.

both are sequences

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

tuples are immutable and list is mutable

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

(42,) (The comma at the end is mandatory)

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

**list value's tuple form**

my\_list = [12, 34, 65, 57]

tuples = tuple(i for i in my\_list)

**tuple value's list form**

my\_tuple = [12, 34, 65, 57]

list1 = list(i for i in my\_tuple)

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

references to list values

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

copy.copy() creates a new compound object and then adds a reference to the object found in the original.

copy.deepcopy() creates a new compound object and then adds a reference to the object found in the original