1. What exactly is []?

with this you can start an array or make a 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]?

ab

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 meaning in bacon?

[3.14, 'cat,' 11, True, 99]

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

concatenate we can use ‘+’, for replication we can use ‘\*’

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

append() will add the value at the end of the list; insert() will add a value based on the index given as the first parameter

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

pop(), remove()

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

if you take a string and make a list out of it by using list(string) we can have the same output as [‘s’,’t’,’r’,’i’,’n’,’g’]

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

tuples are immutable and lists are mutable

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?

tuple(values **for** values **in** list )

list(tuple\_values)

[list(value) **for** value **in** tuples]

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

they contain a reference which points to the object of the assigned value

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

deepcopy() makes a copy of the original, so that in the future when you make a change in the value it will not automatically update the value of the original object;

copy() will make a copy of the reference of the object so that if there is an update in the new object the original will be updated automatically