1. What exactly is []?

[ ] refers to 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.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)]?

[int(3)] -> [3] -> ‘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?

list concatenation is ‘+’ operator

list replication is ‘\*’ operator

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

append() will append at the end of the list while insert() accepts argument on which index you want to insert any element.

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

remove() and delete() methods

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

Both lists and strings can be concatenated , replicated , have indexes and slices, be used in for loops.

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

tuples are immutable written within () , we cannot change it once created

while list is mutable written within [],we can add and remove from the list

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

tup = (42,)

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

a is a collection of elements , a = (1,2,3,4,5)

tuplist = tuple(list(a))

listtup = list(tuple(a))

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

variables are instances of the list that internally refer to the list values inside it.

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

in deepcopy() a copy of the object is copied in another object. hence any change in the object will not reflect in the original copy.

while in shallow copy() the changes in the new copy will reflect the original list as well.