1. What exactly is []?

its a code to create 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?

adds 99 to the list [3.14, 'cat', 11, 'cat', True, 99]

8. How does bacon.remove('cat') change the look of the list in bacon?

removes cat from the list

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

usually + for combining two list \* for replication of elements in list

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

both are used to insert elements in the list but .insert() has an additional parameter for specifying the index where the additional element is to be inserted

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

.pop(), .remove()

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

they both can be indexed and sliced

they both can be iterated over

using len() gives no. of elements present in both of them

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

tuples are 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?

to convert list values to tuple you use tuple in front ie …X = tuple(X)

to convert tuple values to list we use …list(X)

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

they can acts as identifier to the list

X = [some list]

y = X

here X refers to some\_list values in memory it doesn't actually has list values stored now y refers to x which in turn refers to list values hence print(y) and print(X) both will return [somelist]

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

deep copy will copy the references along with any nested objects and make them completely independent from the original list