1. What exactly is []?

It is an Array representation.

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, 4, ‘hello’,6, 8, 10]

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, 'cat', True, 99]

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

+ can be used to cancat list and = can be used to replicate list

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

Append will add item at the last position of list where insert need a index/position where item will assign

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 list and string manage item in a sequence. List is mutable but a string immutable, it means a string does not change after initialize.

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

List is mutable where a touple is immutable. It means a touple item cannot change after initialize.

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

Touple = (42)

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

list=[12,23,34,45,56]

tp=(67,78,89,90,100)

tupleFromList=tuple(list)

listfromtuple=List(tp)

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

It contains a reference of a list.

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

Copy() operation is know as shallow copy that means when we changed anything in copied object it reflect on original object. In copy operation, operation reference will be sharing during copy operation.

During deepcopy() operation, a new object is created that not linked to original object. If we changed something on new object, original object does not change.