1. What exactly is []?

* [] represents list type In python

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, 'cat’, True, 99]

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

* Concatenation : +
* Replication :\*

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

* append() add the given data at the end of list
* insert() add the given data at given index position

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

* remove() : to remove the first occurrence of the given value
* pop() : to remove the data from given index position

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

* Both are iterable and index positioned.
* Concatenation and replication operators are same

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

* List is mutable entity and have many in built-in function
* Tuple is immutable entity, doesn’t have many expect count, index

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

* Var = (42,)

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

* Tuple of list value: tuple(list\_object or list)
* List of tuple values: list(tuple\_object or tuple)

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

* Reference Id of actual list

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

* Copy.copy() creates a shallow copy, any changes in the copied object will not reflect in actual object
* Copy.deepcopy creates a deepcopy, any changes in one copy will reflect in the other copy.