1. What exactly is []?

-> Square brackets [ ] denotes the data structure ‘***List***’. Lists are a collection of iterable, mutable and ordered data. They can contain duplicate data. Here [] means empty list value that contains no item

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’,’6’,’8’,’10’]

***spam.insert(2, “hello”)***

print(spam)

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)]?

-> The output is ‘***d*** ’

4. What is the value of spam[-1]?

-> The output is ‘***d***’

5. What is the value of spam[:2]?

-> The output is ***[‘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')?

-> The output is 1.

7. How does bacon.append(99) change the look of the list value in bacon?

-> The append() method add an item to the end of the list.

So the list becomes ***[3.14, ‘cat’, 11, ’cat’, True, 99]***

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

-> The remove() method, removes the 1st instance of the value provided from the list.

So the list becomes ***[3.14, 11, ‘cat’, True]***

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

-> List concatenation operator is ‘ ***+*** ‘.

List replication operator is ‘ ***\**** ‘.

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

-> The ***append()*** method add an item to the end of the list.

The ***insert()*** method adds an item at any specified index.

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

-> Methods for removing items from list are, ***remove***() and ***pop***()

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

-> Both list and string are ***sequential collections of characters***.

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

-> Lists are mutable, but tuples are immutable.

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

-> t = (42)

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

-> To get a list value’s tuple form, we use tuple()

spam = [1, 2, 3, 4]

print(tuple(spam))

The output is (1, 2, 3, 4)

To get a tuple value’s list form, we use list()

spam = (1, 2, 3, 4)

print(list(spam))

The output is [1, 2, 3, 4]

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

-> They contain ***references*** to list values.

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

-> copy.copy() refers to shallow copy. In the case of shallow copy, a reference of an object is copied into another object. It means that any changes made to a copy of an object does reflect in the original object.

copy.deepcopy() refers to deep copy. In the case of deep copy, a copy of the object is copied into another object. It means that any changes made to a copy of the object do not reflect in the original object.