1. What exactly is []?

In Python, [] represents 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 = [2, 4, 6, 8, 10]

```

Assign ‘hello’ to the third value (index 2) since indexing starts from 0:

```

spam[2] = 'hello'

```

The final result is:

```

[2, 4, 'hello', 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)]?

Given the list spam:

spam = ['a', 'b', 'c', 'd']

we can evaluate spam[int(int('3' \* 2) / 11)] as follows:

1. ('3' \* 2) results in '33'
2. int('33') converts '33' to 33
3. 33 / 11 equals 3.0
4. int(3.0) converts 3.0 to 3

Therefore, spam[3] is 'd'.

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

Using reverse indexing, -1 refers to the last element of a list, -2 to the second last, and so forth. Spam[-1] will return 'd'.

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

We can use slicing on the list as follows:

`spam[:2]` means starting from index 0 up to, but not including, index 2.

For example, `spam[:2]` results in `['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 `index` method returns the first occurrence of the string `'cat'` within the list named `bacon`.

Given the list:

```python

bacon = [3.14, 'cat', 11, 'cat', True]

```

the expression `bacon.index('cat')` will return the value `1`.

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

The `append` method is utilized to add a value to the end of a list. For example:

```python

bacon = [3.14, 'cat', 11, 'cat', True]

bacon.append(99)

```

Following the execution of the append method, the list `bacon` will be `[3.14, 'cat', 11, 'cat', True, 99]`.

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

The `remove` method deletes the first occurrence of an element from a list.

```python

bacon = [3.14, 'cat', 11, 'cat', True, 99]

bacon.remove('cat')

```

After removing, `bacon` will be `[3.14, 11, 'cat', True, 99]`.

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

Concatenation of lists means adding or merging two lists using the `+` operator, like:

```python

L1 = [1,2,3]

L2 = [4,5,6]

L1 + L2 = [1,2,3,4,5,6]

```

Replication duplicates the original list using the `\*` operator:

```python

Replicated\_list = L1 \* 3

Replicated\_list = [1,2,3,1,2,3,1,2,3]

```

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

Use append to add a value to the end of the list.

Use insert to add a value at a specified location in the list.

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

* List.remove(item) removes the first occurrence of the item.
* Del list[2] deletes the specified item by index.

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

Indexing and slicing apply to both strings and lists.

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

The primary difference between a list and a tuple is that lists are mutable, whereas tuples are immutable.

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

Add a comma inside parentheses, e.g., t = (42,).

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

Type casting can be utilized here to convert a tuple into a list, and a list into a tuple. For instance:

list(t)

tuple(spam)

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

A variable can hold an int, float, string, or a list.

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

copy.copy() : Returns a shallow copy of the list.

copy.deepcopy() : A deep copy creates a completely new object and recursively copies all the objects contained within it.