1. What exactly is []?

Ans: List constructor as square brackets

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.)

Ans: **spam.insert(2, 'hello')** is the correct way to insert the value **'hello'** as the third element in the list stored in the variable **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)]?

ANS: 3

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

ANS: ‘d’

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

Ans: [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')?

Ans: 1

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

Ans: [3.14, 'cat,' 11, 'cat,' True,99]

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

Ans: [3.14,' 11, 'cat ,' True,99]

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

Ans:

List Concatenation (**+**): Combines two or more lists into a new list.

List Replication (**\***): Creates a new list with multiple copies of the original list's elements

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

Ans:

* **append()** adds an element to the end of the list.
* **insert()** adds an element at a specific index in the list.

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

Ans: using Pop or Remove

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

* Similarities: Both lists and strings are sequences, support indexing, and slicing.
* Differences: Lists are mutable, can store different data types, and are represented with square brackets **[ ]**, while strings are immutable, specifically for characters, and are represented with single or double quotes.

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

Ans: list are mutable and tuples are immutable

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

Ans: To create a tuple with the integer **42**, use parentheses and include a comma after the value

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

Ans:

* o get a list value's tuple form: Use **my\_tuple = tuple(my\_list)**.
* To get a tuple value's list form: Use **my\_list = list(my\_tuple)**.

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

Ans: Variables that "contain" list values in Python actually store a reference or memory address to the list. They don't hold the list directly but point to where the list's elements are stored in memory. Modifying the list through one variable affects all variables that refer to the same list

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

Ans:

* **copy()** creates a shallow copy with nested elements still referencing the original elements.
* **deepcopy()** creates a deep copy with all nested elements being independent copies of the original elements