1. What exactly is []?

Ans: An empty list is represented by [].

It is a list value that contains no items.

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 = [2,4,6,8,10]

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)]?

Ans: 'd'

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

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

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

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

Ans: list concatenation operators is --> +

list replication operator is --> *

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

Ans: The difference between **append and insert** in python list is
append method can be used for adding new element in the list only but by using **insert** we can add as well as can modify already occupied position.
append method takes one argument (which you have to insert in the list) while insert method takes two elements (first will be the position of element and second will the element itself), Below are examples for both methods use:

Use of Append:

```
list = [1,2,3,4,5]

list.append(6)

print(list) # [1,2,3,4,5,6]
```

Use of Insert:

```
list = [1,2,3,4,5]

list.insert(5, 10) # [1,2,3,4,5,10]

list.insert(1, 10) # [1,10,3,4,5]
```

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

Ans:

The **del** statement and the **remove**() list method are two ways to remove values from a list.

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

Ans:

Both lists and strings can be passed to len().

Indexing and slicing is possible with both lists and strings.

Both used in for loops and concatenated or replicated.

Both are used with the **in** and **not** in operators.

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

Ans:

List

List is a container to contain different types of objects and is used to iterate objects. List is mutable. List is useful for insertion and deletion operations.

Example

list = ['a', 'b', 'c', 'd', 'e']

Tuples

Tuple is also similar to list but contains immutable objects. Tuple processing is faster than List**.** Tuple is useful for read-only operations like accessing elements.

Example

tuples = ('a', 'b', 'c', 'd', 'e')

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

Ans: (42,) (The trailing comma is mandatory.

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

Ans: By using list tuple() and list() functions

Example:

l=[1,2,3,4]

tuple(l)

o/p : (1, 2, 3, 4)

t=(1,2,3,4)

list(t)

o/p : [1, 2, 3, 4]

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

Ans:

They contain references to list values.

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

Ans:

copy.copy() function will do a shallow copy of a list, any changes made to a copy of object **do reflect** in the original object.

copy.deepcopy() function will do a deep copy of a list. any changes made to a copy of object **do not reflect** in the original object in deepcopy()