==================================================================================

1. What exactly is []?

* [] refers to list which does not contain any elements in it.

==================================================================================

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]  
print("List - Before change: ", spam)  
spam[2] = "hello"  
print("List - After change: ", spam)

* Output:

List - Before change: [2, 4, 6, 8, 10]

List - After change: [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)]?

* This int(int('3' \* 2) / 11) results in value 3 and spam[3] is d
* List - Before change: ['a', 'b', 'c', 'd']

List - After change: d

==================================================================================

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

* Spam[-1] is d

==================================================================================

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

* spam[:2] 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')?

* 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] 🡪 appended 99 at the end

==================================================================================

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

* [3.14, 11, 'cat', True] 🡪 Removed the first occurrence of the element cat

==================================================================================

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

* + is the list concatenation operator
* \* is the list replication operator

==================================================================================

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

* append() : this insert the given data item always at the end of the list
* insert() : this insert the given data item at the desired position of the list as needed

==================================================================================

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

* pop() 🡪 remove the item from the end of the list
* remove() 🡪 remove the first occurrence of the value.

==================================================================================

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

* Lists and Strings are ordered sequences
* Both has length
* All the properties of both List and String are almost same except mutability and data type storage
* Strings are immutable but lists are mutable
* String can store only string data type elements but list can store different data type elements

==================================================================================

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

* Tuples and Lists have similar properties except mutability
* Tuples are immutable whereas Lists are mutable

==================================================================================

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

* Need to postfix comma(,) to identify it as tuple if it contains only one value

t = (42,)  
print(type(t))

* Output:  
  <class 'tuple'>

==================================================================================

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

* List value’s tuple form using **tuple()**

l = [34,56]  
print("List is ", l)  
print("List value's tuple form ",tuple(l))

* Output:

List is [34, 56]

List value's tuple form (34, 56)

* Tuple value's list form using **list()**

t = (42,43,34)  
print(“Tuple is ”,t)  
print(“Tuple value’s list form is ”, list(t))

* Output:  
  Tuple is (42, 43, 34)

Tuple value’s list form is [42, 43, 34]

==================================================================================

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

* List variables does not contain the lists itself, instead it contains the references to the list variables.
* These references have the ID numbers associated in python internally.

==================================================================================

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

* Copy.copy():
  + Creates new variable/object and share the reference of the original data
  + Does not create copy of nested objects
  + Update in one object reflects in another only if existing nested object is updated
* Copy.deepcopy():
  + Creates new variable/object and does not share the reference of the original data
  + Recursively add/copies the nested objects
  + Update in one object does not reflect in another object

==================================================================================