1. What exactly is []?

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.insert(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)]?

‘d’

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

‘d’

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

['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')?

1

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

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

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

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

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()?

If we use append ,value will add at the end of list

If we use insert, we can insert the value at specific index

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

Remove

clear

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

Both are sequences

Lists are mutable but string are immutable

Elements in the list can be different type

But in string all are string type

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

Tuples are immutable and lists are mutable

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

tuple(“42”)

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

#tuple to list

tuple=(3,4,3)

a=list(tuple)

print(type(a))

print(a)

#list to tuple

sample\_list=["3","4","3"]

tuple1=tuple(sample\_list)

print(type(a))

print(a)

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

String or integer value

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

copy.copy() creates reference to original object.if you change the copied object- you can change the original object

copy.deepcopy() creates new object and does real copying of original object to new