1. What exactly is []? Ans: It is 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.)

Ans: 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: “c”

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: list bacon becomes [3.14, 'cat', 11, 'cat', True, 99]

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

Ans: list bacon becomes [3.14, 11, 'cat', True], remove deletes only the first instance.

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

Ans: “+” is the list concatenation operator.

“:” is the list replication operator. Example a=[1,2,3,4,5] b=a[:], then b=[1,2,3,4,5]

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

And: append() method inserts an item at the end of the list, where as insert(position, item) inserts an element in the specific position in the list.

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

Ans: pop() and remove() method

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

Ans: both list and string values are indexed and can be accessed by slicing and indexing.

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

Ans: Tuples are immutable i.e tuple values cannot be altered after creation of a tuple, where as lists are mutable, i.e. list values can be altered.

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

tuple\_t=(42,)

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

Example: list value's tuple form

roll\_no = [1, 2, 3, 4, 5]

name = ['arijit', 'subhajit', 'samrat', 'bob', 'shubho']

data = list(zip(roll\_no, name))

data

Output: [(1, 'arijit'), (2, 'subhajit'), (3, 'samrat'), (4, 'bob'), (5, 'shubho')]

Example: tuple value's list form

list\_names=['arijit', 'subhajit', 'samrat', 'bob', 'shubho']

tuple\_names= tuple(list\_names)

print(tuple\_names)

Output: ('arijit', 'subhajit', 'samrat', 'bob', 'shubho')

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

Ams: They contain the address pointing to the list in memory.

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

Example:

import copy

a = [1, 2, 3]

b = [4, 5, 6]

c = [a, b]

d = copy.copy(c)

print(id(c) == id(d)) # False - d is now a new object

print(id(c[0]) == id(d[0])) # True - d[0] is the same object as c[0]

Output:

False

True

d = copy.deepcopy(c)

print(id(c) == id(d)) # False - d is now a new object

print(id(c[0]) == id(d[0])) # False - d[0] is now a new object

Output:

False

False

Conclusion:  
copy.deepcopy(c) creates a new object and copy.copy() creates no new object.