1.What exactly is []?

**Solution**: [] it is representation of 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.)

**Solution**: 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)]?

**Solution**:the value of spam[int(int(‘3’\*2)/11)] 🡺’d’

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

**Solution**: The value of spam is 🡺 ‘d’

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

**Solution**: The value of spam if 🡺 it prints the range of 0 to 2 were :[ ‘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')?

**Solution**:the value of bacon.index(‘cat’) is🡺 1

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

**Solution**: bacon.append(99) give the value of🡺 [3.14, 'cat', 11, 'cat', True,99]

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

**Solution**:bacon.remove(‘cat’) 🡺 [3.14, 11, 'cat', True,99]

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

**Solution**: list concatenation : The list concatenation is adding both list into one list. Were the operator used for concatenation is “+”.

Ex: l1=[1,1,2,3,4]

l2= [2,3,4,5,5]

l1 + l2 # concatenation

List replication: The list replication is repeat each element in the list amount of time. The operator is used for the list replication is “\*”.

Ex: l2 \* 3 # replication of the list l2 into there times.

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

**Solution**: append(): the append is used for append in the list at last position of list .

Ex: l = [1,2,3,4,5]

l.append(“anand”)

output: l = [1,2,3,4,5,’anand’]

insert(): the insert is used for add the value in list at specific location.

Ex: l.insert(2,”anand”) # it add the string in the 2nd position.

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

**Solution**: the two methods for removing items from a list are:

1. remove: the remove keyword is used to remove the item in the list at specific location

ex: l = [1,2,3,4,5,6]

l.remove(3) # it removes the 3 value.

1. pop: the pop keyword is used for pop the last item in the list. And also it can remove the specific item in the list

ex: l.pop() # pop the last item.

l.pop(3) # pop the 3rd index element.

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

**Solution**: list values: we add different types of values in list ,were we can do any type of primitive operation. And list is mutable.

String values: we do any operation in string which having same data type of string. were string is immutable.

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

**Solution**: They are both used to store collection of data. They are both heterogeneous data types means that you can store any kind of data type. They are both sequential data types so we can iterate over the items contained. Items of both types can be accessed by an integer index operator, provided in square brackets, [index]. The key difference between the tuples and lists is that while the **tuples are immutable objects** the **lists are mutable**.

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

**Solution**: No we can’t type a tuple value that contain the integer 42. Because ‘int' object is not iterable.

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

**Solution**: By type casting we can get a list value’s into tuple form and tuple value’s into list form.

Ex: l = [“anand”]

l = tuple(l) # type casting list into tuple. After type casting again reassigned to l.

l = list(l) # type casting tuple into list. After type casting again reassigned to l.

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

**Solution**: l = [ “contain”] # the list contain the string “contain”.

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

**Solution**: copy.copy() 🡺 constructs a new compound object and them inserts references into it to the objects found in original .

Copy.deepcopy() 🡺 construct a new compound object and then, recursively, inserts copies into it of the objects found in original.