**1. What exactly is []?**

**Ans.** In python [] is a list which is a mutable collection of values usually of the same type but could also be heterogeneous.

**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.** All we need to do is to but the index to be able to access and perform any operation of the list, like here if we want to replace the 3rd value the index will be 2 so in list let’s say ls we will write:

ls[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.** The value will be “d”

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

**Ans.** The value will be “d”

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

**Ans.** The values will be [‘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.** The value in the output will be 1, here even though we have multiple occurrences of ‘cat’ only the index of the first occurrence is taken into account for the output.

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

**Ans.** There is no change other than a new element being added at the ending of the list. The final list is like [3.14, ‘cat’, 11, ‘cat’, True, 99]

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

**Ans.** It will remove the first occurrence of the ‘cat’ element of the list. The final list will be [3.14, 11, ‘cat’, True]

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

**Ans.** ‘+’ is used for the list concatenation operation as a operator, and for replication we use ‘\*’ operator. Like list3 = list1 + list2, and for replication list4 = list1 \* 3

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

**Ans.** The append will add the element at the end of the list, where as in insert we specify the index where we want to add the element.

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

**Ans.** First method would be using the remove() method where we specify the element to be deleted. The other method would be to use the pop() method.

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

**Ans.** In both we can access the element using the index, both are sequences and have lengths.

**13. What’s the difference between tuples and lists?**

**Ans.** Tuples are immutable where as lists are mutable, tuple is defined using () where as for list we have []. Iteration and memory consumption is more in list than in tuple.

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

**Ans.** The way to make a tuple with value only 42 we write tubleVar = (42)

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

**Ans.** Similar to what we do for other data type which is type conversion, similar can be done for the tuple and list.

TupleVar = (1,2,3)

listTuple = list(TupleVar)

and

listVar = [1,2,3]

tupleList = tuple(listVar)

**16. Variables that “contain” list values are not necessarily lists themselves. Instead, what do they**

**contain?**

**Ans.** The variable actually contain the reference to the list values not the lists themselves.

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

**Ans.** They are different in the way the operate that is in deepcopy a whole new object is made with the data we want to copy which is a complete different entity independent of from what it has been copied from where as in copy the reference is copied rather that a complete copy, here its like both lists will point to the same location reference rather than different one’s as they are pointing to the same set of values unlike deepcopy where new location is used to store a separate independent copy of the values.