1.What exactly is []?

Solution:

We represent an empty list with []

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] #List Follows Zero Level Indexing

spam[2]='hello'

print(spam)
```

Output:

[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)]?

Solution:

```
spam = ['a','b','c','d']

print(int(int('3'*2)//11))
```

Output:

3

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

Solution:

```
spam = ['a','b','c','d']

print(spam[-1])
```

output:

d

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

Solution:

```
spam = ['a','b','c','d']

print(spam[:2])
```

output:

```
['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:

```
bacon = [3.14, 'cat', 11, 'cat',True]

print(bacon.index('cat'))
```

Output:

```
1
```

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

Solution:

```
bacon = [3.14, 'cat', 11, 'cat',True]

bacon.append(99)

print(bacon)
```

Output:

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

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

Solution:

```
bacon = [3.14, 'cat', 11, 'cat',True]

bacon.remove('cat')

print(bacon)
```

Output:

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

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

Solution:

+ operator is for list concatenation and * operator is for replication

Example:

```
a = ['a','b','c','d']

b = [1,2,3,4]


print(a+b)

print(b*2)
```

Solution:

```
['a', 'b', 'c', 'd', 1, 2, 3, 4]

[1, 2, 3, 4, 1, 2, 3, 4]
```

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

Solution:

append() adds an item to the end of a list, whereas . insert() inserts and item in a specified position in the list.

Example:

```
a = ['a','b','c','d']

a.append('Kedar')

a.insert(2,'Rutu')

print(a)
```

Solution:

```
['a', 'b', 'Rutu', 'c', 'd', 'Kedar']
```

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

Solution:

Remove() helps to remove the very first given element matching from the list. The pop() method removes an element from the list based on the index given.

Example:

a = ['a','b','c','d']

a.remove('b')

a.pop(2)

print(a)

Solution:

['a', 'c']

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

Solution:

Both lists and strings can be passed to len() function, have indexes and slices, be used in for loops, be concatenated or replicated, and be used with the in and not in operators.

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

Solution:

Lists are Mutable, Indexable and Slicable. they can have values added, removed, or changed. Tuples are Immutable but Indexable and Slicable. the tuple values cannot be changed at all. Also, tuples are represented using parentheses (), while lists use the square brackets [].

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

Solution:

(42,) (The trailing comma is mandatory. otherwise its considered as a int by python Interpreter)

```
tup1=(42)
tup2=(42,)
print(type(tup1))
print(type(tup2))
```

Solution:
<class 'int'>
<class 'tuple'>

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

Solution:

The tuple() and list() functions, respectively are used to convert a list to tuple and vice versa.

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

Solution:

They contain references to list values

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

Solution:

The copy() returns a shallow copy of the list, and deepcopy() returns a deep copy of the list.