1. What exactly is []?

Ans: a = [] will create a blank list with the variable name a.

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 = [2,4,6,8,10]
```

```
spam
```

```
[2, 4, 6, 8, 10]
```

```
spam.insert(2,"hello")
```

```
spam
```

```
[2, 4, 'hello', 6, 8, 10]
```

Let's pretend the spam includes the list ['a', 'b', 'c', 'd'] for the next three queries.

Ans:

```
spam.append(['a','b','c','d'])
```

```
spam
```

```
[2, 4, 'hello', 6, 8, 10, ['a', 'b', 'c', 'd']]
```

3. What is the value of spam[int(int('3' * 2) / 11)]?

Ans:

```
spam
```

```
[2, 4, 'hello', 6, 8, 10, ['a', 'b', 'c', 'd']]
```

```
spam[int(int('3' * 2) / 11)]
```

```
6
```

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

Ans:

```
spam[-1]
```

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

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

```
spam[:2]
```

```
[2, 4]
```

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: This will produce the first index of ‘cat’

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

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

```
1
```

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

Ans:

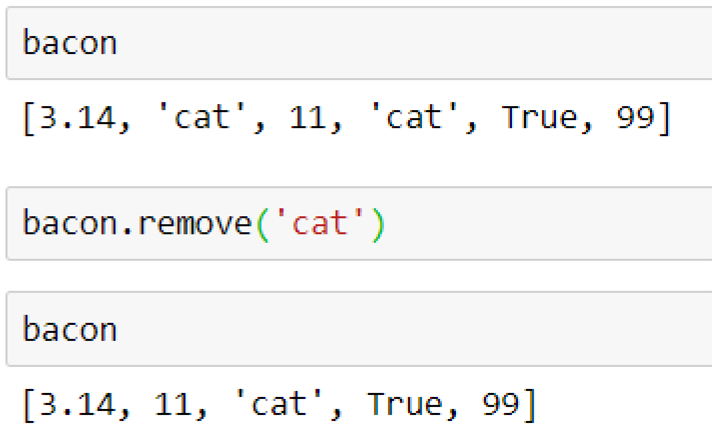
```
bacon.append(99)
```

```
bacon
```

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

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

Ans:



```
bacon
```

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

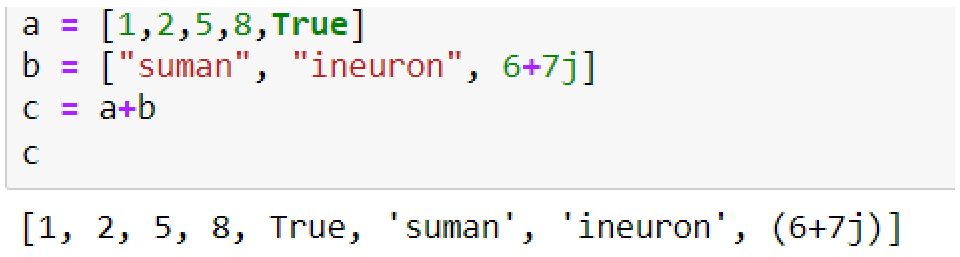
```
bacon.remove('cat')
```

```
bacon
```

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

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

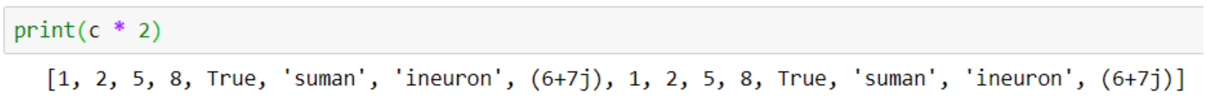
Ans: List concatenation means joining two links.



```
a = [1,2,5,8,True]
b = ["suman", "ineuron", 6+7j]
c = a+b
c
```

```
[1, 2, 5, 8, True, 'suman', 'ineuron', (6+7j)]
```

Replication operator



```
print(c * 2)
```

```
[1, 2, 5, 8, True, 'suman', 'ineuron', (6+7j), 1, 2, 5, 8, True, 'suman', 'ineuron', (6+7j)]
```

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

Ans: Append object to the end of the list. Whereas insert object before the defined index.

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

Ans:

Remove() delete the element

Pop() will delete the object based on given index and return the removed value.

```
c.remove(1)
```

```
c
```

```
[2, 5, 8, True, 'suman', 'ineuron', (6+7j)]
```

```
c.pop(3)
```

```
True
```

```
c
```

```
[2, 5, 8, 'suman', 'ineuron', (6+7j)]
```

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

Ans: List value and string value both are in sequence.

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

Ans: List is mutable but tuple is immutable. One can append, Insert, modify the list value but in tuple these operations cannot be performed.

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

Ans:

```
t = (42,)
```

```
t
```

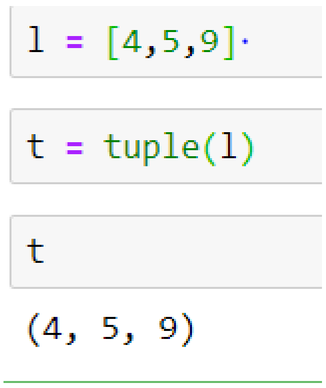
```
(42,)
```

```
type(t)
```

```
tuple
```

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

Ans:

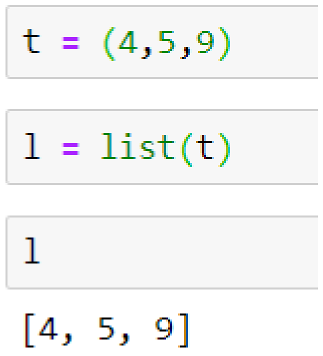


```
l = [4,5,9]
```

```
t = tuple(l)
```

```
t
```

```
(4, 5, 9)
```



```
t = (4,5,9)
```

```
l = list(t)
```

```
l
```

```
[4, 5, 9]
```

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

Ans: Variables will contain references to list values rather than list values themselves.

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

Ans:

For a normal list shallow copy output is same as deepcopy output.

```python
import copy
l1 = [1,2,3,4]
l2 = copy.copy(l1)
l2[2] = 500
print(l1,l2)
```

```
[1, 2, 3, 4] [1, 2, 500, 4]
```

```python
import copy
l1 = [1,2,3,4]
l2 = copy.deepcopy(l1)
l2[2] = 500
print(l1,l2)
```

```
[1, 2, 3, 4] [1, 2, 500, 4]
```

But for nested list shallow copy != deep copy

```python
# Shallow copy
import copy
l1 = [[1,2,3,4], [5,6,7,8]]
l2 = copy.copy(l1)
l2[1][0] = 500
print(l1,l2)
```

```
[[1, 2, 3, 4], [500, 6, 7, 8]] [[1, 2, 3, 4], [500, 6, 7, 8]]
```

```python
# deepcopy
import copy
l1 = [[1,2,3,4], [5,6,7,8]]
l2 = copy.deepcopy(l1)
l2[1][0] = 500
print(l1,l2)
```

```
[[1, 2, 3, 4], [5, 6, 7, 8]] [[1, 2, 3, 4], [500, 6, 7, 8]]
```