1. What exactly is []?
Ans-Empty brackets are used when there is no value in the 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=[2,4,6,8,10]
>>> spam[2]='hello'
>>> print(spam)

[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)]?
Ans-

```
spam='a','b','c','d'
spam[int(int('3' * 2) / 11)]
'd'
```

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

```
spam='a','b','c','d'
spam[-1]

'd'
```

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

```
spam='a','b','c','d'
spam[:2]

('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-

```
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=[3.14, 'cat', 11, 'cat', True]
bacon.append(99)
print(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]
bacon.remove('cat')
print(bacon)

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

9. What are the list concatenation and list replication oper10**ators?**
Ans- list concatenation is(+)and list replication is(*).

10. What is difference between the list methods append() and insert()?
Ans- In Append the item is added to the last and Adds the items to a specified position in the insert.

11. What are the two methods for removing items from a list?
Ans- pop and remove are two methods for removing items from a list.

12. Describe how list values and string values are identical.
Ans- list and string value both are a group of charecters.
13. What's the difference between tuples and lists?
Ans- List is a sequence of items separated by commas and the items are enclosed in square brackets [ ] and it is mutable.Tuple a sequence of items separated by commas and items are enclosed in ( )and it is immutable.
14. How do you type a tuple value that only contains the integer 42?
Ans-

```
>>> thing = (42,)
>>> print(thing)
>>> type (thing)

(42,)
tuple
```

(Tuples a sequence of items separated by commas and items are enclosed in parentheses ().)

```
Tuple = (3.14,1+2j,True,11,'pat')
#tuple to list
List = list(Tuple)
#print list
print(type(List))
print(List)
```

```
<class 'list'>
[3.14, (1+2j), True, 11, 'pat']
```

```
# list to tuple
list_names=[4,4.11,'Ram']
tuple_names= tuple(list_names)
print(tuple_names)
print(type(tuple_names))

(4, 4.11, 'Ram')
<class 'tuple'>
```

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

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

```
#copy()
a = [1,2,3]
b = a.copy()
print(a)
print(b)

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

```
# deepcopy()
import copy
A_list = [[1,2], [2,3], [3,4]]
B_list = copy.deepcopy(B_list)

print("A list:", A_list)
print("B list:", B_list)

A list: [[1, 2], [2, 3], [3, 4]]
B list: [[1, 1, 1], [2, 2, 2], [3, 3, 3]]
```