1. The empty list represented by [] is a list that contains no items. This is similar to " which represents an empty string.
2. spam= [2, 4, 6, 8, 10] , Replace third value with value"Hello" i.e. spam[2]='Hello'
3. spam=['a','b','c','d'], spam[int(int('3' * 2) / 11)]= d
4. spam[-1]=d
5. spam[:2]=a,b
6. bacon=[3.14,'cat',11,'cat',True], bacon.index('cat')=1
7. bacon.append(99), print(bacon) -> 3.14,'cat',11,'cat',True,99
8. bacon.remove('cat') , print(bacon) -> 3.14,11,'cat',True,99
9. The list concatenation and list replication operators are

```
list_1 = ['ML','DL','AI','CV','NLP']
list_2 = ['RNN','CNN','SVN']
print(list_1 + list_2) # List Concatenation
print(list_2*2) # List Replication

Ans:: 1. 'ML','DL','AI','CV','NLP','RNN','CNN','SVN'
      2. 'RNN','CNN','SVN','RNN','CNN','SVN'
```

10. the list methods append() and insert()::

```
list= [3.14,'cat', 11, 'cat',True]
list.append(22)
print(list)
list.insert(0,33)
print(list)
```

Ans :: `[3.14, 'cat', 11, 'cat', True, 22]#append means valued is added at last.`
`[33, 3.14, 'cat', 11, 'cat', True, 22]#inset means at 0 index we inserted 33.`

11. Two methods for removing items from list: pop and remove
12. The difference between tuples and lists are list are mutable, but tuples are immutable. Tuples are represented using parentheses, (), while lists use the square brackets, []
13. `tuple=(2,34,'d',45)`
    `Tuple[3]=45`
14. List to Tuple :: `list= [3.14,'cat', 11, 'cat',True]`
    `Print(tuple(list))`
    `o/p : (3.14,'cat',11,'cat', True)`
    Tuple to List :: `tuple = (3.14,'cat', 11, 'cat',True)`
    `Print(list(tuple))`
    `o/p : list= [3.14,'cat', 11, 'cat',True]`
15. Variables that "contain" list values are not necessarily lists themselves. They contain references to list values.
16. The copy.copy() function will do a shallow copy of a list, while the copy.deepcopy() function will do a deep copy of a list. That is, only copy.deepcopy() will duplicate any lists inside the list