Basic List and List Comprehension Questions

1. Create a list of Numbers from 0 to 50 using List comprehension.

Numbers = [x for x in range(51)] →it prints numbers from 0 to 50
Numbers = [x for x in range(1,51)] →it prints numbers from 1 to 50
Numbers = [x for x in range(1,51,2)] →it prints odd numbers from 0 to 50
Numbers = [x for x in range(2,51,2)] →it prints even numbers from 0 to 50

2. Create a list of square of a number from 1 to 20.

Square = [x**2 for x in range(21)]

```
//squares using Null list and append method
squares = []

for x in range(10):
   # raise x to the power of 2
   squares.append(x**2)

print(squares)
```

```
//squares using existing list and append method
squares = [1,4,9,16,25]

for x in range(6,11):
   # raise x to the power of 2
   squares.append(x**2)

print(squares)
```

3. Create a list of 10 characters and make a slice of list from character 4 to 8.

Char = [“A”,”B”,”C”, “D”,”E”,”F”, “G”,”H”,”I”,”J”]
Slice = Char[3:8]

4. Give the differences between list insert and list append with suitable examples.

Char = [“A”,”B”,”C”, “D”,”E”,”F”, “G”,”H”,”I”,”J”]
Char.append(“k”)
print(Char) →it append value K at last position
Char.insert(0,”a”)
Print(Char) → It insert value at desired index

5. Create two list and combine / merge them using list extend method and concatenation operator.

```
L = ['red', 'green', 'yellow']
M = [1,2,3]
L.extend(M)
print(L)
```

6. Create a list of subject names L = [“EPR”, “EPA”,”FEP”, “BMA”, “FOM”,’’OS”]
   A. Remove “BMA” using index and store it in another variable.
   B. Remove “OS” by value.

```
L = [“EPR”, “EPA”,”FEP”, “BMA”, “FOM”,’’OS”]
x = L.pop(3)
print(L)
print(x)

L.remove(“OS”)
Print(L)
```

7. Create a list which prints only even numbers from 0 to 25.

Numbers = [x for x in range(2,25,2)] →it prints even numbers from 0 to 25

8. Create a list of numbers from 0 to 10 and multiply all numbers using 5 using list comprehension.

```
multiply = [x*5 for x in range(11)]
```

9. Create a list L = [“EPR”, “EPA”,”FEP”, “BMA”, “FOM”,’’OS”] and create a new list which has first letter of each subjects.

```
L = ["EPR","EPA","FEP","BMA","FOM","OS"]

letters = [ name[0] for name in L ]
print(letters)

letters = [ name[1] for name in L ]
print(letters)
```

10. Use list comprehension to separate characters in a string and store it in list.

```
letters = [ letter for letter in "embedded programming"]
print(letters)
```

11. Create following list
    L = [‘a’,’b’,’c’,’d’,’e’]
    M=[‘P’,’Q’,’R’,’S’,’T’]
    1. Convert all lower case to upper from list L.
    2. Convert all upper case to lower from list M.

```
L = ['a','b','c','d','e']
M = ['P','Q','R','S','T']
upper_case = [ letter.upper() for letter in L ]
lower_case = [ letter.lower() for letter in M ]

print(upper_case, lower_case)
```

12. Create a string with your name and phone number, Separate out phone number’s digit from the string.
    X = “My number is 9898989898.”

```
X = “My number is 9898989898.”
phone_number = [ a for a in x if a.isdigit()]

print(phone_number)
```

13. Create a text file which has email addresses of a student in each line. Open the file in read mode and print all lines in python.

```
file = open("E:\list.txt", 'r')
email = [ line for line in file ]

for x in email:
   print(x)
```

14. Create a list
    1. to double a number from 1 to 10 using function and list comprehension.
    2. to double only even numbers from 1 to 10 using function and list comprehension.

```
def double(x):
   return x*2

nums = [double(x) for x in range(1,11)]
print(nums)

nums = [double(x) for x in range(1,11) if x%2 == 0]
print(nums)
```

15. Create two list of numbers (same size) and add all possible combination of numbers from list using list comprehension.
    L=[1,2,3,4,5]
    M=[10,20,30,40,50]

```
L=[1,2,3,4,5]
M=[10,20,30,40,50]
addition = [x+y for x in L for y in M ]
print(addition)
```

16. Count a number of occurrence from the list.
    1. Count how many times 9 occurs in the list L = [9,1,2,9,2,9,9,3,4]
    2. Count how many times ‘red’ occurs in the list M = [‘red’, ‘red’,’blue’,’yellow’]

```
L=[9,1,2,9,2,9,9,3,4]
x=L.count(9)
print(x)

M=['red','red','blue','yellow','red']
y=M.count('red')
print(y)
```

17. Count all the items from the list using counter() function.
    L=[9,2,2,3,3,4,4,5,5,5,9,9,9,2]

```
L=[9,2,2,3,3,4,4,5,5,5,9,9,9,2]
from collections import Counter
print(Counter(L))
```

18. Reverse the list items for the following list.
    L=[1,2,3,4,5,6,7,8,9]
    M=[‘a’,’b’,’c’,’d’,’e’]

```
L=[1,2,3,4,5,6,7,8,9]
M=['a','b','c','d','e']
L.reverse()
M.reverse()
print(L)
print(M)
print(L,M)
```

19. Access each list items for the following list in reversed order.
    L=[1,2,3,4,5,6,7,8,9]
    M=[‘a’,’b’,’c’,’d’,’e’]

```
L=[1,2,3,4,5,6,7,8,9]
M=['a','b','c','d','e']
```

```
for x in reversed(L):
   print(x)
for y in reversed(M):
   print(y)
```

20. Reverse the following list and then Sort the list
L=[1,2,3,3,5,3,2,8,1]

```
L=[1,2,3,3,5,3,2,8,1]
L.reverse()
L.sort()
print(L)
```
(Python does not differentiate between numbers and strings while sorting.)

21. Sort the list in reverse order.
L=[1,2,3,3,5,3,2,8,1]

```
L=[1,2,3,3,5,3,2,8,1]
L.sort(reverse=True)
print(L)
```

22. Sum all the elements of a list.
L=[1,2,3,3,5,3,2,8,1]

```
L=[1,2,3,3,5,3,2,8,1]
x=sum(L)
print(x)
```

23. What is the use of following command?
print(L[::-1]) if L=[1,2,3,3,5,3,2,8,1]

```
v L=[1,2,3,3,5,3,2,8,1]
print(L[::-1])
```
It prints the List in Reverse Order.

24. How to insert items at the end of the list using len() function?
L=['a','b','c'] and items to be inserted at the is [1,2,3,4,5]

```
L=['a','b','c']
L[len(L):] = [1,2,3,4,5]
Print(L)
```

25. Insert value= 4 at index 2 in the following list without insert() or append() method. To be inserted=[1,2,3,4,5] in the list L=['a','b','c']

```
L=['a','b','c']
L[2:2] = [1,2,3,4,5]
print(L)
```

26. Given list is L = ['a', 'b', 'c', 'd', 'e']. Delete items from index 1 to 3 without delete(), pop() or del() functions

```
L = ['a', 'b', 'c', 'd', 'e']
L[1:4] = []
Print(L)
```

27. What is the output of the following commands?

```
L1 = ['a', 'b', 'c', 'd', 'e']
L2 = L1[:]
```

It will copy the list L1 to L2. Their locations are different so L1 is not equal to L2.

28. Create a list with absolute values of given list L=[-4,-4,-3,-2,0,1,2,3,-4]

```
L=[-4,-4,-3,-2,0,1,2,3,-4]
Absolute = [abs(x) for x in L]
print(Absolute)
```

29. How to remove the whitespace added with following list items?
colors = [' red', ' green ', 'blue ']

Use strip() function to remove the whitespace as follows.

```
colors = ['  red', '  green ', 'blue  ']
L = [x.strip() for x in colors]
print(L)
```

30. why you should use list comprehension more often?

- List comprehensions are more concise to write and hence they turn out to be very useful in many contexts.
- Since a list comprehension is an expression, you can use it wherever you need an expression (e.g. as an argument to a function, in a return statement).
- List comprehensions run substantially faster than manual for loop statements (roughly twice as fast).
- It offers a major performance advantage especially for larger data sets.

31. Print only positive numbers from the list given below.
L=[-4,-4,-3,-2,0,1,2,3,-4]

32. How to convert List L=[1,2,3] into Tuple?

```
L=[1, 2, 3]
T = tuple(L)
print(T)
```

33. How to convert string **‘abcd’** into tuple?

```
T = tuple('abcd')
print(T)
```

34. With example explain the concepts of Tuple Packing and Tuple Unpacking.

```
T = ('red', 'green', 'blue', 'cyan')
print(T)
(a, b, c, d) = T
print(a)
print(b)
print(c)
print(d)
```

35. Swap two numbers a=10, b=20 using concept of Python Tuple.

```
a=10
b=20
print(a)
print(b)
b,a = a,b
print(a)
print(b)
```

36. Split an email address abc@xyz.com into a user name and a domain using Python Tuple.

```
email = 'abc@xyz.com'
user,domain=email.split('@')
print(user)
print(domain)
```

37. Create a tuple and access the items of tuple using negative indexing.

```
T = ('red', 'green', 'blue', 'yellow', 'black')

print(T[-5])
print(T[-4])
print(T[-3])
print(T[-2])
print(T[-1])
```

38. Explain Tuple slicing with example.

```
T = ('a', 'b', 'c', 'd', 'e', 'f')
print(T[2:5])
# Prints ('c', 'd', 'e')
print(T[0:2])
#Prints('a','b')
```

39. Can you change the value of the following tuples. Justify
    1. T = ('a','b','c') →change value b
    2. T = (1, [2, 3], 4)→change value 2

    1. Value of b cannot be edited because Tuple is immutable.
    2. Value of 2 in the list can be changed because List inside the tuple is mutable.

```
T = (1, [2, 3], 4)
T[1][0] = 'EPR'
print(T)
# It prints T = (1, ['EPR', 3], 4)
```

40. Sort the following tuple T = ('cc', 'aa', 'dd', 'bb')

Method-1

```
Print(Tuple(sorted(T)))
```

Method-2

```
T = ('cc', 'aa', 'dd', 'bb')
tmp = list(T)
tmp.sort()
T = tuple(tmp)
print(T)
```

41. Count the number of occurrence from the following tuple.
    1. Red in ('red', 'blue','green','red')
    2. 9 in (9,2,3,4,9,8,6,9,9,4)

```
T = ('red', 'green', 'blue','red')
print(T.count('red'))

T = (9,2,3,4,9,8,6,9,9,4)
print(T.count(9))
```

42. Make the summation of all the elements of following List
L = [1, 2, 3, 4, 5]

```
L = [1, 2, 3, 4, 5]
x=sum(L)
print(x)

# Start with '10' and add all items in a list
L = [1, 2, 3, 4, 5]
x = sum(L, 10)
print(x)
# Prints 25

L = [1, 2, 3, 4, 5]
x = sum(L[3:5], 10)
print(x)
```

43. Create a tuple of sequence 0 to 9 using range function.

```
T=tuple(range(0,10))
print(T)

T=tuple(range(10))
print(T)

T=tuple(range(30,10,-2))
print(T)
```

44. How to create set from string **‘abcde’**?

```
S = set(‘abcde’)
```

45. How to convert List into set?

```
L=[1,2,3,4,5]
S = set(L)
Print(S)
```

46. How to create a set of a number from 0 to 9?
S = set(range(0,10)) Print(s)