Comprehensive list

1.Write a list comprehension to generate squares of numbers from 1 to 10

```
list1=[x**2 for x in range(1,11)]
list1
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

2. Create a list of even numbers between 1 and 50 using list comprehension.

```
list1=[x for x in range (1,51) if x%2==0]
list1
```

```
[2,
 4,
 6,
 8,
 10,
 12,
 14,
 16,
 18,
 20,
 22,
 24,
 26,
 28,
 30,
 32,
 34,
 36,
 38,
 40,
 42,
 44,
 46,
 48,
 50]
```

3. Convert all strings in a list to uppercase using list comprehension.

```
lst = ["apple", "Banana", "cat"]
result = [x.upper() for x in lst]
result
```

```
['APPLE', 'BANANA', 'CAT']
```

4. Given a list of integers, create a new list that contains only the positive numbers

```
nums = [-2, 5, -1, 7, 0, 3]
result = [x for x in nums if x > 0]
result
```

```
[5, 7, 3]
```

5. Create a list of tuples (num, num^2) for numbers 1 to 5

```
result = [(x, x**2) for x in range(1, 6)]
result
```

```
[(1, 1), (2, 4), (3, 9), (4, 16), (5, 25)]
```

6. Extract all vowels from a given string using list comprehension

```
string1 = "Rohit"
vowels = [x for x in string1 if x.lower() in "aeiou"]

vowels
```

```
['o', 'i']
```

7. Flatten a 2D list using list comprehension

```
matrix = [[1,2], [3,4], [5,6]]
flat = [num for row in matrix for num in row]
flat
```

```
[1, 2, 3, 4, 5, 6]
```

8. Replace all negative numbers in a list with 0 using list comprehension

```
nums = [-3, 4, -1, 6]
result = [x if x >= 0 else 0 for x in nums]
result
```

```
[0, 4, 0, 6]
```

9. Given a list of words, create a list of lengths of each word.

```
words = ["apple", "banana", "cat"]
lengths = [len(x) for x in words]
lengths
```

```
[5, 6, 3]
```

10. Filter out words that start with the letter 'A' or 'a'

```
words = ["Apple", "banana", "Ant", "cat"]
result = [w for w in words if w.lower().startswith('a')]
result
```

```
['Apple', 'Ant']
```

11. From a list of numbers, generate a list of “even” or “odd” strings using list comprehension. (Like → [“even”, “odd”, “odd”, “even”...])

```
nums = [2, 3, 5, 8]
result = ["even" if x % 2 == 0 else "odd" for x in nums]
result
```

```
['even', 'odd', 'odd', 'even']
```

12. Create a list of numbers divisible by both 3 and 5 in range 1–100.

```
result = [x for x in range(1, 101) if x % 3 == 0 and x % 5 == 0]
result
```

```
[15, 30, 45, 60, 75, 90]
```

13. Write a nested list comprehension to generate a multiplication table for 1–5.

```
table = [(i, j, i*j) for i in range(1, 6) for j in range(1, 11)]
table
```

```
[(1, 1, 1),
 (1, 2, 2),
 (1, 3, 3),
 (1, 4, 4),
 (1, 5, 5),
 (1, 6, 6),
 (1, 7, 7),
 (1, 8, 8),
 (1, 9, 9),
 (1, 10, 10),
```

```
 (2, 1, 2),
 (2, 2, 4),
 (2, 3, 6),
 (2, 4, 8),
 (2, 5, 10),
 (2, 6, 12),
 (2, 7, 14),
 (2, 8, 16),
 (2, 9, 18),
 (2, 10, 20),
 (3, 1, 3),
 (3, 2, 6),
 (3, 3, 9),
 (3, 4, 12),
 (3, 5, 15),
 (3, 6, 18),
 (3, 7, 21),
 (3, 8, 24),
 (3, 9, 27),
 (3, 10, 30),
 (4, 1, 4),
 (4, 2, 8),
 (4, 3, 12),
 (4, 4, 16),
 (4, 5, 20),
 (4, 6, 24),
 (4, 7, 28),
 (4, 8, 32),
 (4, 9, 36),
 (4, 10, 40),
 (5, 1, 5),
 (5, 2, 10),
 (5, 3, 15),
 (5, 4, 20),
 (5, 5, 25),
 (5, 6, 30),
 (5, 7, 35),
 (5, 8, 40),
 (5, 9, 45),
 (5, 10, 50)]
```

14. Convert a dictionary's keys into a list using list comprehension.

```
d = {"a":1, "b":2, "c":3}
keys = [k for k in d]
keys
```

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

15. Extract numeric digits from a string using list comprehension.

```
s = "a1b2c3"
digits = [ch for ch in s if ch.isdigit()]
digits
```

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

16. Use list comprehension to remove all spaces from a string.

```
s = "python is fun"
result = [ch for ch in s if ch != " "]
result
```

```
['p', 'y', 't', 'h', 'o', 'n', 'i', 's', 'f', 'u', 'n']
```

```
s = "python is fun"
s = ''.join([c for c in s if c != ' '])
s
```

```
'pythonisfun'
```

17. Create a list of characters that appear more than once in a string

```
s = "programming"
result = [ch for ch in set(s) if s.count(ch) > 1]
result
```

```
['g', 'm', 'r']
```

18. From a list of sentences, generate a list of all words (split using list comprehension).

```
sentences = ["I love python", "Python is easy"]
words = [word for s in sentences for word in s.split()]
words
```

```
['I', 'love', 'python', 'Python', 'is', 'easy']
```

19. Create a list of unique elements from a list using list comprehension + condition.

```
lst = [1, 2, 2, 3, 4, 4]
unique = [x for x in lst if lst.count(x) == 1]
unique
```

```
[1, 3]
```

20. Generate all pairs (x, y) where x is from list A and y is from list B (cartesian product).

```
A = [1, 2]
B = ['a', 'b']
pairs = [(x, y) for x in A for y in B]
pairs
```

```
[(1, 'a'), (1, 'b'), (2, 'a'), (2, 'b')]
```

Lambda functions

1. Write a lambda to add two numbers.

```
add=lambda a,b : a+b
add(2,3)
```

```
5
```

2. Create a lambda to check if a number is even.

```
is_even = lambda x: x % 2 == 0
is_even(2)
```

```
True
```

3. Write a lambda to get the last character of a string

```
last_char = lambda s: s[-1]
last_char("python")
```

```
'n'
```

4. Use lambda with map() to square every number in a lis

```
square = lambda x: x ** 2
list(map(square, [1, 2, 3, 4]))
```

```
[1, 4, 9, 16]
```

5. Use lambda with filter() to get only odd numbers from a list

```
is_odd = lambda x: x % 2 != 0
list(filter(is_odd, [1, 2, 3, 4, 5]))
```

```
[1, 3, 5]
```

6. Use sorted() + lambda to sort a list of tuples by second value.

```
sorted([(1, 3), (4, 1), (2, 2)], key=lambda x: x[1])
```

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

7. Create a lambda to check if a string is a palindrome.

```
is_palindrome = lambda s: s == s[::-1]
is_palindrome("madam")
```

```
True
```

8. Use lambda to find maximum of three numbers.

```
max_of_three = lambda a, b, c: max(a, b, c)
max_of_three(3, 9, 5)
```

```
9
```

9. Write a lambda to reverse a string.

```
reverse_string = lambda s: s[::-1]
reverse_string("hello")
```

```
'olleh'
```

10. Use lambda with map() to convert a list of strings to integers

```
to_int = lambda x: int(x)
list(map(to_int, ["1", "2", "3"]))
```

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

11. Use lambda with filter() to remove empty strings from a list

```
non_empty = lambda x: x != ""
list(filter(non_empty, ["a", "", "b", ""]))
```

```
['a', 'b']
```

12. Use lambda to compute factorial using reduce() (yeah, that one-liner madness)

```
from functools import reduce


factorial = lambda n: reduce(lambda x, y: x * y, range(1, n + 1), 1)
factorial(5)
```

```
120
```

13. Write a lambda that returns the larger of two numbers.

```
larger = lambda a, b: a if a > b else b
larger(10, 7)
```

```
10
```

14. Use lambda to check if number is divisible by 5.

```
div_by_5 = lambda x: x % 5 == 0
div_by_5(25)
```

```
True
```

15. Use lambda + map() to add 10 to each element of a list.

```
add_10 = lambda x: x + 10
list(map(add_10, [1, 2, 3]))
```

```
[11, 12, 13]
```

16. Use lambda to sort a list of dictionaries by a key (like "age")

```
people = [{"name": "A", "age": 30}, {"name": "B", "age": 25}]
sorted(people, key=lambda x: x["age"])
```

```
[{'name': 'B', 'age': 25}, {'name': 'A', 'age': 30}]
```

17. Write a lambda that returns True if a character is a vowel

```
is_vowel = lambda c: c.lower() in "aeiou"
is_vowel("A")
```

```
True
```

18. Use lambda + filter to extract words of length > 5 from a list.

```
long_word = lambda w: len(w) > 5
list(filter(long_word, ["python", "java", "machine"]))
```

```
['python', 'machine']
```

19. Use lambda to calculate the area of a circle (πr^2)

```
circle_area = lambda r: 3.142 * r * r
circle_area(7)
```

```
153.958
```

20. Write a lambda to remove duplicates from a list using filter + set

```
seen = set()
remove_dupes = lambda x: x not in seen and not seen.add(x)
list(filter(remove_dupes, [1, 2, 2, 3, 3, 4]))
```

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

21. Use lambda with reduce() to find the product of all numbers in a list.

```
```

```
reduce(lambda x, y: x * y, [1, 2, 3, 4])
```

```
24
```

22. Write a lambda that returns absolute value of a number.

```
absolute = lambda x: abs(x)
absolute(-15)
```

```
15
```

23. Use lambda to sort a list of strings by their length

```
sorted(["apple", "kiwi", "banana"], key=lambda s: len(s))
```

```
['kiwi', 'apple', 'banana']
```

24. Use lambda to get only uppercase characters from a string

```
is_upper = lambda c: c.isupper()
list(filter(is_upper, "HeLLo"))
```

```
['H', 'L', 'L']
```

25. Write a lambda that returns the square if number is even, cube if odd

```
even_square_odd_cube = lambda x: x**2 if x % 2 == 0 else x**3
even_square_odd_cube(3)
```

```
27
```

26. Use lambda with map to convert Celsius to Fahrenheit.

```
c_to_f = lambda c: (c * 9/5) + 32
c_to_f(30)
```

```
86.0
```

27. Write a lambda to check if two strings are anagrams

```
is_anagram = lambda a, b: sorted(a) == sorted(b)
is_anagram("listen", "silent")
```

```
True
```

28. Use lambda to extract only numeric values from a mixed list

```
is_number = lambda x: isinstance(x, (int, float))
list(filter(is_number, [1, "a", 2.5, "3"]))
```

```
[1, 2.5]
```

29. Use lambda inside any() to check if any list element is negative.

```
any(map(lambda x: x < 0, [1, 2, -3, 4]))
```

```
True
```

30. Use lambda to generate a function that multiplies any number by n

```
multiplier = lambda n: lambda x: x * n
times_5 = multiplier(5)
times_5(4)
```

```
20
```