INFOSYS INTERNSHIP 6.0
PYTHON TASK

Prompt:

Hey, I need a set of important Python tasks to complete. Please give me a list of commonly used Python codes that every learner should know and that are mostly asked in interviews. Also, include a short comment line for each part of the code or query explaining what it does.

Solution:

1. **Reverse a string**

```
s = "hello"
print(s[::-1])  # Reverses the string using slicing
```

2. **Check if a string is a palindrome**

```
s = "madam"
print(s == s[::-1])  # Returns True if the string reads the same backward
```

3. **Count vowels in a string**

```
s = "beautiful"
vowels = "aeiou"
count = sum(1 for ch in s if ch.lower() in vowels)
print(count)  # Counts how many vowels are in the string
```

4. **Find factorial of a number (using recursion)**

```
def factorial(n):
    return 1 if n == 0 else n * factorial(n - 1)
print(factorial(5))  # Returns 120
```

5. **Check if a number is prime**

```
n = 13
is_prime = all(n % i != 0 for i in range(2, int(n**0.5) + 1))
print(is_prime)  # True if prime
```

6. **Fibonacci sequence (first n terms)**

```
n = 10
a, b = 0, 1
for _ in range(n):
```

```
    print(a, end=" ")
    a, b = b, a + b  # Generates next Fibonacci number
```

7. Find maximum and minimum in a list

```
nums = [4, 7, 2, 9, 1]
print(max(nums), min(nums))  # Prints highest and lowest values
```

8. Remove duplicates from a list

```
nums = [1, 2, 2, 3, 4, 4]
print(list(set(nums)))  # Converts to set to remove duplicates
```

9. Count frequency of elements in a list

```
from collections import Counter
nums = [1, 2, 2, 3, 3, 3]
print(Counter(nums))  # Shows count of each unique number
```

10. Check if two strings are anagrams

```
s1, s2 = "listen", "silent"
print(sorted(s1) == sorted(s2))  # True if both contain same letters
```

11. Find second largest number in a list

```
nums = [4, 10, 9, 2]
print(sorted(set(nums))[-2])  # Removes duplicates and gets 2nd last
```

12. Sum of digits of a number

```
n = 1234
print(sum(int(d) for d in str(n)))  # Adds each digit
```

13. List comprehension example (squares of even numbers)

```
nums = [1, 2, 3, 4, 5]
print([x**2 for x in nums if x % 2 == 0])  # Efficient filtering + transformation
```

14. File read & write

```
with open("test.txt", "w") as f:
    f.write("Hello, world!")  # Writes text into file
with open("test.txt", "r") as f:
    print(f.read())  # Reads and prints content
```

15. Exception handling

```
try:
    x = 10 / 0
```

```
except ZeroDivisionError:
    print("Division by zero is not allowed.")  # Prevents program crash
```

16. Lambda and map

```
nums = [1, 2, 3, 4]
print(list(map(lambda x: x**2, nums)))  # Applies function to each item
```

17. Class & Object basics

```
class Person:
    def __init__(self, name):
        self.name = name  # Constructor initialization
    def greet(self):
        return f"Hello, {self.name}"

p = Person("Vamsi")
print(p.greet())  # Object method call
```

18. Dictionary sorting by value

```
d = {"a": 3, "b": 1, "c": 2}
print(sorted(d.items(), key=lambda x: x[1]))  # Sorts dict by value
```

19. List flattening

```
nested = [[1, 2], [3, 4], [5]]
flat = [num for sublist in nested for num in sublist]
print(flat)  # [1, 2, 3, 4, 5]
```

20. Using enumerate to loop with index

```
for i, val in enumerate(["apple", "banana", "cherry"]):
    print(i, val)  # Prints index and value
```