

Strings

1. Reverse a String

- **Description:** Write a function that takes a string and returns it reversed.
- **Explanation:** Use string slicing to reverse the string.
- **Sample Test Cases:**
 - Input: "hello"
Output: "olleh"
 - Input: "Python"
Output: "nohtyP"

2. Count Vowels

- **Description:** Count the number of vowels in a given string.
- **Explanation:** Loop through each character and check if it's a vowel.
- **Sample Test Cases:**
 - Input: "apple"
Output: 2
 - Input: "hello world"
Output: 3

3. Check Palindrome

- **Description:** Check if a given string is a palindrome.
- **Explanation:** A palindrome reads the same forward and backward. Use string slicing to check.
- **Sample Test Cases:**
 - Input: "racecar"
Output: True
 - Input: "python"
Output: False

4. Count Words in a String

- **Description:** Count the number of words in a given string.
- **Explanation:** Split the string by spaces and count the resulting list length.
- **Sample Test Cases:**
 - Input: "hello world"
Output: 2
 - Input: "Python is fun"
Output: 3

5. Replace a Character in a String

- **Description:** Replace a specified character in a string with another character.
- **Explanation:** Use `replace()` to substitute characters.
- **Sample Test Cases:**

- Input: "hello", "l", "x"
Output: "hexxo"
- Input: "banana", "a", "o"
Output: "bonono"

6. Count Occurrences of a Character

- **Description:** Count the occurrences of a specified character in a string.
- **Explanation:** Loop through the string or use `count()`.
- **Sample Test Cases:**
 - Input: "hello world", "o"
Output: 2
 - Input: "mississippi", "s"
Output: 4

7. Convert Uppercase to Lowercase

- **Description:** Convert all uppercase letters in a string to lowercase.
- **Explanation:** Use `lower()` to convert.
- **Sample Test Cases:**
 - Input: "HELLO"
Output: "hello"
 - Input: "Python"
Output: "python"

8. Find Longest Word

- **Description:** Find the longest word in a given sentence.
- **Explanation:** Split the string into words and find the longest.
- **Sample Test Cases:**
 - Input: "I love Python programming"
Output: "programming"

Lists

9. Sum of List Elements

- **Description:** Find the sum of all elements in a list.
- **Explanation:** Use `sum()` or a loop.
- **Sample Test Cases:**
 - Input: [1, 2, 3, 4]
Output: 10
 - Input: [-1, 2, -3, 4]
Output: 2

10. Remove Duplicates from List

- **Description:** Remove duplicates from a list and return unique elements.

- **Explanation:** Use a set to filter duplicates.
- **Sample Test Cases:**
 - Input: `[1, 2, 2, 3, 4, 4]`
Output: `[1, 2, 3, 4]`

11. Find Maximum in List

- **Description:** Return the largest number in a list.
- **Explanation:** Use `max()` or loop through the list.
- **Sample Test Cases:**
 - Input: `[1, 3, 5, 2, 4]`
Output: `5`

Dictionaries

12. Word Frequency Count

- **Description:** Count occurrences of each word in a string and return as a dictionary.
- **Explanation:** Split the string, then use a dictionary for counting.
- **Sample Test Cases:**
 - Input: `"apple apple orange"`
Output: `{'apple': 2, 'orange': 1}`

13. Find Key by Value

- **Description:** Find the key corresponding to a given value in a dictionary.
- **Explanation:** Loop through items and check if the value matches.
- **Sample Test Cases:**
 - Input: `{1: 'a', 2: 'b', 3: 'c'}, value = 'b'`
Output: `2`

Loops

14. Print Multiplication Table

- **Description:** Print the multiplication table of a given number up to 10.
- **Explanation:** Use a `for` loop and format output.
- **Sample Test Cases:**

Input: `3`

Output:

`3 x 1 = 3`

`3 x 2 = 6`

`...`

$$3 \times 10 = 30$$

15. Fibonacci Series up to N

- **Description:** Generate the Fibonacci series up to **N** numbers.
- **Explanation:** Use a loop to calculate each Fibonacci number based on the previous two.
- **Sample Test Cases:**
 - Input: 5
Output: [0, 1, 1, 2, 3]

Strings

16. Capitalize First Letter of Each Word

- **Description:** Given a string, capitalize the first letter of each word.
- **Explanation:** Use the `title()` function or split and capitalize each word.
- **Sample Test Cases:**
 - Input: "hello world"
Output: "Hello World"
 - Input: "python programming"
Output: "Python Programming"

17. Remove All White Spaces

- **Description:** Remove all spaces from a string.
- **Explanation:** Use `replace(" ", "")` or a loop to exclude spaces.
- **Sample Test Cases:**
 - Input: "hello world"
Output: "helloworld"
 - Input: " a b c "
Output: "abc"

18. Find Substring Occurrences

- **Description:** Count occurrences of a substring in a string.
- **Explanation:** Use `count()` to count the substring in the string.
- **Sample Test Cases:**
 - Input: "hello hello", "lo"
Output: 2
 - Input: "mississippi", "iss"
Output: 2

19. Remove Punctuation from String

- **Description:** Remove punctuation from a given string.
- **Explanation:** Use `string.punctuation` with `replace()` or list comprehension.

- **Sample Test Cases:**
 - Input: "Hello, World!"
Output: "Hello World"
 - Input: "Python's fun!"
Output: "Pythons fun"

20. Find Shortest Word in Sentence

- **Description:** Find the shortest word in a given sentence.
 - **Explanation:** Split the sentence and find the minimum length word.
 - **Sample Test Cases:**
 - Input: "I love Python programming"
Output: "I"
 - Input: "This is a test"
Output: "a"
-

Lists

21. Count Even and Odd Numbers in List

- **Description:** Write a function to count even and odd numbers in a list.
- **Explanation:** Loop through each number and check if it's even or odd.
- **Sample Test Cases:**
 - Input: [1, 2, 3, 4, 5]
Output: {'odd': 3, 'even': 2}
 - Input: [2, 4, 6, 8]
Output: {'odd': 0, 'even': 4}

22. Find Second Largest Number in List

- **Description:** Return the second largest number in a list.
- **Explanation:** Sort the list or use two variables to track the largest and second largest.
- **Sample Test Cases:**
 - Input: [3, 1, 4, 1, 5]
Output: 4
 - Input: [10, 20, 20, 5]
Output: 10

23. Merge Two Lists

- **Description:** Merge two lists and remove duplicates.
- **Explanation:** Use list concatenation and convert to a set for unique values.
- **Sample Test Cases:**
 - Input: [1, 2, 3], [3, 4, 5]
Output: [1, 2, 3, 4, 5]

- Input: [10, 20], [20, 30, 40]
Output: [10, 20, 30, 40]

24. Split List into Even and Odd Lists

- **Description:** Split a list of integers into two lists: one containing even numbers, the other odd numbers.
- **Explanation:** Loop through the list and separate even and odd numbers.
- **Sample Test Cases:**
 - Input: [1, 2, 3, 4, 5]
Output: {'even': [2, 4], 'odd': [1, 3, 5]}
 - Input: [10, 11, 12]
Output: {'even': [10, 12], 'odd': [11]}

25. Rotate List Elements

- **Description:** Rotate a list by *n* positions.
- **Explanation:** Use slicing to rearrange list elements.
- **Sample Test Cases:**
 - Input: [1, 2, 3, 4, 5], n=2
Output: [4, 5, 1, 2, 3]
 - Input: [10, 20, 30], n=1
Output: [30, 10, 20]

Dictionaries

26. Invert a Dictionary

- **Description:** Invert keys and values in a dictionary.
- **Explanation:** Use dictionary comprehension to swap keys and values.
- **Sample Test Cases:**
 - Input: {'a': 1, 'b': 2}
Output: {1: 'a', 2: 'b'}
 - Input: {'x': 10, 'y': 20}
Output: {10: 'x', 20: 'y'}

27. Merge Two Dictionaries

- **Description:** Merge two dictionaries. If a key exists in both, sum the values.
- **Explanation:** Use `update()` and a loop to add values for duplicate keys.
- **Sample Test Cases:**
 - Input: {'a': 1, 'b': 2}, {'b': 3, 'c': 4}
Output: {'a': 1, 'b': 5, 'c': 4}
 - Input: {'x': 10}, {'x': 20, 'y': 30}
Output: {'x': 30, 'y': 30}

28. Find Minimum Key in Dictionary

- **Description:** Find the key with the smallest value in a dictionary.
- **Explanation:** Use `min()` with dictionary values to identify the minimum key.
- **Sample Test Cases:**
 - Input: `{'a': 3, 'b': 1, 'c': 2}`
Output: `'b'`
 - Input: `{'x': 10, 'y': 5, 'z': 15}`
Output: `'y'`

29. Sum of Values in Dictionary

- **Description:** Calculate the sum of all values in a dictionary.
- **Explanation:** Use `sum()` on the values of the dictionary.
- **Sample Test Cases:**
 - Input: `{'a': 1, 'b': 2, 'c': 3}`
Output: `6`
 - Input: `{'x': 10, 'y': 20}`
Output: `30`

30. Count Character Frequency in String Using Dictionary

- **Description:** Given a string, count the frequency of each character and store it in a dictionary.
- **Explanation:** Loop through the string and update the dictionary for each character.
- **Sample Test Cases:**
 - Input: `"hello"`
Output: `{'h': 1, 'e': 1, 'l': 2, 'o': 1}`
 - Input: `"mississippi"`
Output: `{'m': 1, 'i': 4, 's': 4, 'p': 2}`

Loops

31. Find Prime Numbers up to N

- **Description:** Write a function to print all prime numbers up to a given number `N`.
- **Explanation:** Use nested loops to check if a number is only divisible by 1 and itself.
- **Sample Test Cases:**
 - Input: `10`
Output: `[2, 3, 5, 7]`
 - Input: `20`
Output: `[2, 3, 5, 7, 11, 13, 17, 19]`

32. Calculate Factorial of a Number

- **Description:** Calculate the factorial of a given number.

- **Explanation:** Use a loop or recursion to multiply all integers up to n .
- **Sample Test Cases:**
 - Input: 5
Output: 120
 - Input: 3
Output: 6

33. Sum of Digits in a Number

Description: Find the sum of digits of a given number.

Explanation: Convert to string, iterate over digits, and sum them.

Sample Test Cases:

- Input: 123
Output: 6
- Input: 456
Output: 15

34. Generate Pattern of Stars

Description: Print a right-angled triangle pattern of $*$ with n rows.

Explanation: Use nested loops to print rows and columns of $*$.

Sample Test Cases:

Input: 3

Output:

```
*
**
***
```

35. Check Anagram

- **Description:** Check if two strings are anagrams (contain the same characters with the same frequency).
- **Explanation:** Sort both strings and compare them, or use a dictionary to count character frequency.
- **Sample Test Cases:**
 - Input: "listen", "silent"
Output: True

- Input: "python", "typhon"

Output: True

36. Remove Vowels from String

- **Description:** Remove all vowels from a given string.
- **Explanation:** Use a loop to skip vowels or use `replace()` with each vowel.
- **Sample Test Cases:**
 - Input: "hello"
Output: "hll"
 - Input: "apple"
Output: "ppl"

37. Find Most Frequent Character

- **Description:** Find the character that appears most frequently in a string.
- **Explanation:** Use a dictionary to count characters and find the max.
- **Sample Test Cases:**
 - Input: "mississippi"
Output: 's'
 - Input: "hello"
Output: 'l'

38. Remove Duplicate Characters

- **Description:** Remove duplicate characters in a string, keeping only the first occurrence.
- **Explanation:** Loop through the string and build a new string with unique characters.
- **Sample Test Cases:**
 - Input: "banana"
Output: "ban"
 - Input: "hello"
Output: "helo"

39. Check if String is Pangram

- **Description:** Check if a given string contains all letters of the alphabet.
- **Explanation:** Convert string to lowercase, then check if all 26 letters are present.
- **Sample Test Cases:**
 - Input: "The quick brown fox jumps over the lazy dog"
Output: True
 - Input: "Hello world"
Output: False

Lists

40. Find Unique Elements in List

- **Description:** Return a list of unique elements, excluding duplicates.
- **Explanation:** Use a set or loop to filter unique items.
- **Sample Test Cases:**
 - Input: [1, 2, 2, 3, 4, 4]
Output: [1, 3]
 - Input: [5, 5, 5, 5]
Output: []

41. Concatenate Two Lists Alternately

- **Description:** Merge two lists alternately, picking elements from each list.
- **Explanation:** Use a loop to alternate between elements of the two lists.
- **Sample Test Cases:**
 - Input: [1, 2, 3], [a, b, c]
Output: [1, 'a', 2, 'b', 3, 'c']
 - Input: [5, 6], [7, 8, 9]
Output: [5, 7, 6, 8, 9]

42. Flatten Nested List

- **Description:** Convert a nested list into a flat list.
- **Explanation:** Use recursion or nested loops to flatten.
- **Sample Test Cases:**
 - Input: [[1, 2], [3, 4], [5]]
Output: [1, 2, 3, 4, 5]
 - Input: [1, [2, [3]]]
Output: [1, 2, 3]

43. Find Missing Number in List

- **Description:** Given a list of consecutive numbers with one missing, find the missing number.
- **Explanation:** Use the sum formula to calculate the expected total and find the missing number.
- **Sample Test Cases:**
 - Input: [1, 2, 4, 5]
Output: 3
 - Input: [10, 11, 12, 14]
Output: 13

44. Find Intersection of Two Lists

- **Description:** Return a list containing common elements from two lists.
- **Explanation:** Use sets or a loop to find common elements.
- **Sample Test Cases:**
 - Input: [1, 2, 3], [2, 3, 4]
Output: [2, 3]
 - Input: [10, 20, 30], [15, 20, 25]
Output: [20]

Dictionaries

45. Sort Dictionary by Keys

- **Description:** Sort a dictionary by its keys and return it.
- **Explanation:** Use `sorted()` with dictionary keys.
- **Sample Test Cases:**
 - Input: `{'b': 2, 'a': 1, 'c': 3}`
Output: `{'a': 1, 'b': 2, 'c': 3}`

46. Filter Dictionary by Values

- **Description:** Given a dictionary and a threshold, return a new dictionary with values above the threshold.
- **Explanation:** Use dictionary comprehension to filter values.
- **Sample Test Cases:**
 - Input: `{'a': 1, 'b': 5, 'c': 3}`, threshold = 2
Output: `{'b': 5, 'c': 3}`

47. Add Key-Value Pair if Key Doesn't Exist

- **Description:** Add a key-value pair to a dictionary only if the key doesn't exist.
- **Explanation:** Use `if key not in dict` to check existence.
- **Sample Test Cases:**
 - Input: `{'a': 1}`, key = 'b', value = 2
Output: `{'a': 1, 'b': 2}`
 - Input: `{'x': 10}`, key = 'x', value = 20
Output: `{'x': 10}`

48. Update Dictionary with Default Values

- **Description:** Given a dictionary, update missing keys with a specified default value.
- **Explanation:** Loop through desired keys and set default if missing.
- **Sample Test Cases:**
 - Input: `{'a': 1}`, default keys: `['a', 'b']`, default value: 0
Output: `{'a': 1, 'b': 0}`
 - Input: `{'x': 10}`, default keys: `['x', 'y']`, default value: -1
Output: `{'x': 10, 'y': -1}`

49. Count Frequency of Words in List Using Dictionary

- **Description:** Given a list of words, count the frequency of each word using a dictionary.
- **Explanation:** Loop through the list and increment the count for each word.
- **Sample Test Cases:**
 - Input: `["apple", "banana", "apple"]`
Output: `{'apple': 2, 'banana': 1}`

- Input: ["cat", "dog", "cat", "cat"]
Output: {'cat': 3, 'dog': 1}

50. Delete Key from Dictionary

- **Description:** Remove a specified key from a dictionary if it exists.
 - **Explanation:** Use `del` or `pop()` to remove the key.
 - **Sample Test Cases:**
 - Input: {'a': 1, 'b': 2}, key = 'b'
Output: {'a': 1}
 - Input: {'x': 10, 'y': 20}, key = 'z'
Output: {'x': 10, 'y': 20}
-

Loops

51. Sum of Squares up to N

- **Description:** Find the sum of squares of all integers from 1 to N.
- **Explanation:** Loop through numbers and add squares.
- **Sample Test Cases:**
 - Input: 3
Output: 14 ($1^2 + 2^2 + 3^2$)
 - Input: 5
Output: 55

52. Generate Fibonacci Series Using While Loop

- **Description:** Generate the Fibonacci series up to a given number using a `while` loop.
- **Explanation:** Start with two variables and repeatedly add them.
- **Sample Test Cases:**
 - Input: 10
Output: [0, 1, 1, 2, 3, 5, 8]

53. Count Occurrences of a Digit in a Number

- **Description:** Count occurrences of a specified digit in a given number.
- **Explanation:** Convert number to string or use mod and div operations.
- **Sample Test Cases:**
 - Input: 12345, digit=3
Output: 1
 - Input: 33333, digit=3
Output: 5

54. Reverse a Number Using Loop

- **Description:** Reverse a given integer without converting to a string.
- **Explanation:** Use mod and div operations to reverse the digits.
- **Sample Test Cases:**

- Input: 123
Output: 321
- Input: 4567
Output: 7654

55. Sum of Factorials of Each Digit

- **Description:** Find the sum of the factorials of each digit in a number.
- **Explanation:** Use factorial computation on each digit.
- **Sample Test Cases:**
 - Input: 145
Output: 145 ($1! + 4! + 5! = 145$)
 - Input: 123
Output: 9

56. Convert String to Title Case

- **Description:** Convert each word in a string to title case (capitalize the first letter of each word).
- **Explanation:** Use `title()` or manually capitalize each word.
- **Sample Test Cases:**
 - Input: "hello world"
Output: "Hello World"
 - Input: "python programming"
Output: "Python Programming"

57. Check if All Characters are Unique

- **Description:** Check if all characters in a string are unique.
- **Explanation:** Use a set to track characters and check for duplicates.
- **Sample Test Cases:**
 - Input: "abcdef"
Output: True
 - Input: "hello"
Output: False

58. Count Upper and Lowercase Letters

- **Description:** Count the number of uppercase and lowercase letters in a string.
- **Explanation:** Use `isupper()` and `islower()` to classify each character.
- **Sample Test Cases:**
 - Input: "Hello World"
Output: {'uppercase': 2, 'lowercase': 8}
 - Input: "PYTHON"
Output: {'uppercase': 6, 'lowercase': 0}

59. Replace Spaces with Hyphens

- **Description:** Replace all spaces in a string with hyphens.
- **Explanation:** Use `replace(" ", "-")`.
- **Sample Test Cases:**

- Input: "hello world"
Output: "hello-world"
- Input: "a b c"
Output: "a-b-c"

60. Extract Digits from String

- **Description:** Extract all numeric digits from a string and return them as a string.
 - **Explanation:** Loop through the string and collect digits using `isdigit()`.
 - **Sample Test Cases:**
 - Input: "abc123def"
Output: "123"
 - Input: "5 apples and 10 oranges"
Output: "510"
-

Lists

61. Find Common Elements in Three Lists

- **Description:** Find the elements common in three different lists.
- **Explanation:** Use set intersections or nested loops.
- **Sample Test Cases:**
 - Input: [1, 2, 3], [2, 3, 4], [3, 4, 5]
Output: [3]
 - Input: [1, 2], [2, 3], [3, 4]
Output: []

62. Find List Symmetric Difference

- **Description:** Return a list containing elements unique to each of the two lists.
- **Explanation:** Use set symmetric difference.
- **Sample Test Cases:**
 - Input: [1, 2, 3], [2, 3, 4]
Output: [1, 4]
 - Input: [5, 6], [6, 7]
Output: [5, 7]

63. Sort List by Last Character

- **Description:** Sort a list of strings based on the last character of each string.
- **Explanation:** Use `sorted()` with a custom key for sorting.
- **Sample Test Cases:**
 - Input: ["apple", "banana", "cherry"]
Output: ["banana", "apple", "cherry"]
 - Input: ["car", "bat", "apple"]
Output: ["apple", "car", "bat"]

64. Transpose a 2D Matrix

- **Description:** Given a 2D list (matrix), transpose it (flip rows and columns).
- **Explanation:** Use nested loops to transpose elements.
- **Sample Test Cases:**
 - Input: `[[1, 2], [3, 4], [5, 6]]`
Output: `[[1, 3, 5], [2, 4, 6]]`
 - Input: `[[1, 2, 3], [4, 5, 6]]`
Output: `[[1, 4], [2, 5], [3, 6]]`

65. Filter List Based on Condition

- **Description:** Return a list of numbers from the input list that are greater than a given threshold.
 - **Explanation:** Use list comprehension to filter elements.
 - **Sample Test Cases:**
 - Input: `[1, 5, 8, 10]`, threshold = 6
Output: `[8, 10]`
 - Input: `[4, 3, 2, 1]`, threshold = 2
Output: `[4, 3]`
-

Dictionaries

66. Group Elements by Key

- **Description:** Group a list of dictionaries by a specified key.
- **Explanation:** Use a loop and dictionary to group items.
- **Sample Test Cases:**
 - Input: `[{"name": "Alice", "group": "A"}, {"name": "Bob", "group": "B"}, {"name": "Charlie", "group": "A"}]`
Output: `{"A": ["Alice", "Charlie"], "B": ["Bob"]}`

67. Filter Dictionary by Key

- **Description:** Return a new dictionary containing only specified keys from the original dictionary.
- **Explanation:** Use dictionary comprehension.
- **Sample Test Cases:**
 - Input: `{"a": 1, "b": 2, "c": 3}`, keys = `["a", "c"]`
Output: `{"a": 1, "c": 3}`
 - Input: `{"x": 10, "y": 20}`, keys = `["z"]`
Output: `{}`

68. Count Vowels in String Using Dictionary

- **Description:** Count the occurrences of each vowel in a given string.
- **Explanation:** Use a dictionary to track vowel counts.

- **Sample Test Cases:**
 - Input: "hello"
Output: {"e": 1, "o": 1}
 - Input: "banana"
Output: {"a": 3}

69. Convert Dictionary Keys to Uppercase

- **Description:** Convert all keys in a dictionary to uppercase.
- **Explanation:** Use dictionary comprehension.
- **Sample Test Cases:**
 - Input: {"a": 1, "b": 2}
Output: {"A": 1, "B": 2}
 - Input: {"x": 10, "y": 20}
Output: {"X": 10, "Y": 20}

70. Calculate Product of Values in Dictionary

- **Description:** Calculate the product of all values in a dictionary.
- **Explanation:** Use `reduce()` or a loop to multiply values.
- **Sample Test Cases:**
 - Input: {"a": 1, "b": 2, "c": 3}
Output: 6
 - Input: {"x": 4, "y": 5}
Output: 20

Loops

71. Sum of Cubes up to N

- **Description:** Find the sum of cubes of all integers from 1 to N.
- **Explanation:** Use a loop to add cubes of numbers.
- **Sample Test Cases:**
 - Input: 3
Output: 36 ($1^3 + 2^3 + 3^3$)
 - Input: 4
Output: 100

72. Count Digits in a Number

- **Description:** Count the number of digits in a given integer.
- **Explanation:** Use mod and div operations or convert to string.
- **Sample Test Cases:**
 - Input: 12345
Output: 5

■ Input: 100

Output: 3

73. Generate Number Pyramid

- **Description:** Print a number pyramid of height **N**.
- **Explanation:** Use nested loops to construct the pyramid pattern.
- **Sample Test Cases:**

Input: 3

Output:

1

2 2

3 3 3

■

74. Find Perfect Numbers up to N

- **Description:** Find all perfect numbers up to a given number **N**.
- **Explanation:** A perfect number is equal to the sum of its divisors.
- **Sample Test Cases:**

■ Input: 28

Output: [6, 28]

75. Reverse String Without Using Built-in Function

- **Description:** Reverse a string without using built-in functions.
- **Explanation:** Use a loop to reverse characters manually.
- **Sample Test Cases:**

■ Input: "hello"

Output: "olleh"

■ Input: "world"

Output: "dlrow"

76. Dictionaries

66. Group Elements by Key

- **Description:** Group a list of dictionaries by a specified key.
- **Explanation:** Use a loop and dictionary to group items.
- **Sample Test Cases:**

■ Input: [{ "name": "Alice", "group": "A" }, { "name": "Bob", "group": "B" }, { "name": "Charlie", "group": "A" }]

Output: {"A": ["Alice", "Charlie"], "B": ["Bob"]}

67. Filter Dictionary by Key

- **Description:** Return a new dictionary containing only specified keys from the original dictionary.
- **Explanation:** Use dictionary comprehension.
- **Sample Test Cases:**
 - Input: `{"a": 1, "b": 2, "c": 3}`, keys = `["a", "c"]`
Output: `{"a": 1, "c": 3}`
 - Input: `{"x": 10, "y": 20}`, keys = `["z"]`
Output: `{}`

68. Count Vowels in String Using Dictionary

- **Description:** Count the occurrences of each vowel in a given string.
- **Explanation:** Use a dictionary to track vowel counts.
- **Sample Test Cases:**
 - Input: `"hello"`
Output: `{"e": 1, "o": 1}`
 - Input: `"banana"`
Output: `{"a": 3}`

69. Convert Dictionary Keys to Uppercase

- **Description:** Convert all keys in a dictionary to uppercase.
- **Explanation:** Use dictionary comprehension.
- **Sample Test Cases:**
 - Input: `{"a": 1, "b": 2}`
Output: `{"A": 1, "B": 2}`
 - Input: `{"x": 10, "y": 20}`
Output: `{"X": 10, "Y": 20}`

70. Calculate Product of Values in Dictionary

- **Description:** Calculate the product of all values in a dictionary.
- **Explanation:** Use `reduce()` or a loop to multiply values.
- **Sample Test Cases:**
 - Input: `{"a": 1, "b": 2, "c": 3}`
Output: `6`
 - Input: `{"x": 4, "y": 5}`
Output: `20`

Loops

71. Sum of Cubes up to N

- **Description:** Find the sum of cubes of all integers from 1 to `N`.
- **Explanation:** Use a loop to add cubes of numbers.
- **Sample Test Cases:**

- Input: 3
Output: 36 ($1^3 + 2^3 + 3^3$)
- Input: 4
Output: 100

72. Count Digits in a Number

- **Description:** Count the number of digits in a given integer.
- **Explanation:** Use mod and div operations or convert to string.
- **Sample Test Cases:**
 - Input: 12345
Output: 5
 - Input: 100
Output: 3

73. Generate Number Pyramid

- **Description:** Print a number pyramid of height N.
- **Explanation:** Use nested loops to construct the pyramid pattern.
- **Sample Test Cases:**

Input: 3
Output:

1

2 2

3 3 3

■

74. Find Perfect Numbers up to N

- **Description:** Find all perfect numbers up to a given number N.
- **Explanation:** A perfect number is equal to the sum of its divisors.
- **Sample Test Cases:**
 - Input: 28
Output: [6, 28]

75. Reverse String Without Using Built-in Function

- **Description:** Reverse a string without using built-in functions.
- **Explanation:** Use a loop to reverse characters manually.
- **Sample Test Cases:**
 - Input: "hello"
Output: "olleh"
 - Input: "world"
Output: "dlrow"

This continues the list up to problem 75. Please let me know if you'd like more problems!

40

You said:

continue

ChatGPT said:

ChatGPT

Here's the continuation from problem 76.

Strings

76. Count Consonants in a String

- **Description:** Count the number of consonants in a given string.
- **Explanation:** Use a loop to check each character; exclude vowels and non-alphabet characters.
- **Sample Test Cases:**
 - Input: "hello world"
Output: 7
 - Input: "python"
Output: 5

77. Remove Leading and Trailing Spaces

- **Description:** Remove spaces from the beginning and end of a string.
- **Explanation:** Use `strip()` to trim leading and trailing whitespace.
- **Sample Test Cases:**
 - Input: " hello world "
Output: "hello world"
 - Input: " python "
Output: "python"

78. Count Specific Word in Sentence

- **Description:** Count the occurrences of a specified word in a sentence.
- **Explanation:** Split the string into words and count occurrences of the word.
- **Sample Test Cases:**
 - Input: "hello world hello", word = "hello"
Output: 2

- Input: "apple banana apple", word = "apple"
Output: 2

79. Extract Initials from Name

- **Description:** Given a full name, return the initials of each word.
- **Explanation:** Split the name and take the first character of each word.
- **Sample Test Cases:**
 - Input: "John Doe"
Output: "J.D"
 - Input: "Alice Bob Carol"
Output: "A.B.C"

80. Remove Digits from String

- **Description:** Remove all numeric digits from a string.
 - **Explanation:** Use `isdigit()` to filter out digits.
 - **Sample Test Cases:**
 - Input: "hello123"
Output: "hello"
 - Input: "4pp13s"
Output: "ppls"
-

Lists

81. Find Duplicate Elements in List

- **Description:** Return a list of duplicate elements found in the input list.
- **Explanation:** Use a dictionary or set to track duplicate elements.
- **Sample Test Cases:**
 - Input: [1, 2, 2, 3, 3, 4]
Output: [2, 3]
 - Input: [5, 5, 6]
Output: [5]

82. Find Missing Elements in Consecutive List

- **Description:** Given a list of consecutive numbers with some missing, return the missing numbers.
- **Explanation:** Use a range and set difference to find missing elements.
- **Sample Test Cases:**
 - Input: [1, 2, 4, 5]
Output: [3]
 - Input: [10, 12, 13]
Output: [11]

83. Count Positive and Negative Numbers in List

- **Description:** Count the positive and negative numbers in a list.

- **Explanation:** Loop through and check if each number is positive or negative.
- **Sample Test Cases:**
 - Input: `[1, -2, 3, -4, 5]`
Output: `{'positive': 3, 'negative': 2}`
 - Input: `[-1, -2, -3]`
Output: `{'positive': 0, 'negative': 3}`

84. Sort List of Tuples by Second Element

- **Description:** Sort a list of tuples based on the second element of each tuple.
- **Explanation:** Use `sorted()` with a lambda function.
- **Sample Test Cases:**
 - Input: `[(1, 3), (2, 1), (3, 2)]`
Output: `[(2, 1), (3, 2), (1, 3)]`
 - Input: `[(4, 5), (2, 6), (1, 4)]`
Output: `[(1, 4), (4, 5), (2, 6)]`

85. Split List into Chunks

- **Description:** Split a list into chunks of a specified size.
- **Explanation:** Use slicing in a loop to create sublists.
- **Sample Test Cases:**
 - Input: `[1, 2, 3, 4, 5]`, chunk size = 2
Output: `[[1, 2], [3, 4], [5]]`
 - Input: `[10, 20, 30, 40]`, chunk size = 3
Output: `[[10, 20, 30], [40]]`

Dictionaries

86. Convert Two Lists into Dictionary

- **Description:** Given two lists, one of keys and one of values, combine them into a dictionary.
- **Explanation:** Use `zip()` to create key-value pairs.
- **Sample Test Cases:**
 - Input: `['a', 'b', 'c'], [1, 2, 3]`
Output: `{'a': 1, 'b': 2, 'c': 3}`
 - Input: `['x', 'y'], [10, 20]`
Output: `{'x': 10, 'y': 20}`

87. Count Characters in a List of Words

- **Description:** Given a list of words, count the occurrences of each character.
- **Explanation:** Loop through each word, then each character, updating counts in a dictionary.
- **Sample Test Cases:**

- Input: ["apple", "banana"]
Output: {'a': 5, 'p': 2, 'l': 1, 'e': 1, 'b': 1, 'n': 2}
- Input: ["cat", "dog"]
Output: {'c': 1, 'a': 1, 't': 1, 'd': 1, 'o': 1, 'g': 1}

88. Update Dictionary with Maximum Values

- **Description:** Given two dictionaries with the same keys, update each key with the maximum value.
- **Explanation:** Use a loop and `max()` for each key.
- **Sample Test Cases:**
 - Input: {'a': 1, 'b': 5}, {'a': 3, 'b': 2}
Output: {'a': 3, 'b': 5}
 - Input: {'x': 10}, {'x': 20, 'y': 5}
Output: {'x': 20, 'y': 5}

89. Remove Keys with None Values

- **Description:** Remove all keys from a dictionary that have `None` as a value.
- **Explanation:** Use dictionary comprehension to filter out `None` values.
- **Sample Test Cases:**
 - Input: {'a': 1, 'b': None, 'c': 3}
Output: {'a': 1, 'c': 3}
 - Input: {'x': None, 'y': 20}
Output: {'y': 20}

90. Check if Dictionary is Empty

- **Description:** Check if a given dictionary is empty.
- **Explanation:** Use `len()` or simply check if the dictionary evaluates to `False`.
- **Sample Test Cases:**
 - Input: {}
Output: `True`
 - Input: {'a': 1}
Output: `False`

Loops

91. Generate Multiples of a Number

- **Description:** Given a number and `n`, return a list of the first `n` multiples of that number.
- **Explanation:** Use a loop to generate multiples.
- **Sample Test Cases:**
 - Input: 2, n=5
Output: [2, 4, 6, 8, 10]

- Input: 3, n=3
Output: [3, 6, 9]

92. Check if Number is Prime

- **Description:** Check if a given number is prime.
- **Explanation:** Use a loop to check divisibility up to the square root.
- **Sample Test Cases:**
 - Input: 7
Output: True
 - Input: 10
Output: False

93. Sum of Digits of All Elements in List

- **Description:** Given a list of integers, return the sum of the digits of each integer.
- **Explanation:** Convert each number to a string and sum the digits.
- **Sample Test Cases:**
 - Input: [123, 45, 6]
Output: 21 (1+2+3 + 4+5 + 6)
 - Input: [10, 20, 30]
Output: 6

94. Check for Armstrong Number

- **Description:** Check if a given number is an Armstrong number.
- **Explanation:** Sum the cubes of each digit and check if it equals the original number.
- **Sample Test Cases:**
 - Input: 153
Output: True
 - Input: 123
Output: False

95. Find GCD of Two Numbers

- **Description:** Find the Greatest Common Divisor (GCD) of two numbers.
- **Explanation:** Use the Euclidean algorithm.
- **Sample Test Cases:**
 - Input: 24, 36
Output: 12
 - Input: 20, 15
Output: 5

Miscellaneous (Combining Skills)

96. Sort List of Dictionaries by Key

- **Description:** Sort a list of dictionaries based on a specified key in each dictionary.
- **Explanation:** Use `sorted()` with a key.
- **Sample Test Cases:**
 - Input: `[{"name": "Alice", "age": 25}, {"name": "Bob", "age": 20}]`, key = "age"
Output: `[{"name": "Bob", "age": 20}, {"name": "Alice", "age": 25}]`

97. Generate Fibonacci Series as Dictionary

- **Description:** Generate a dictionary where keys are positions and values are Fibonacci numbers up to N.
- **Explanation:** Use a loop to generate Fibonacci numbers and add them to the dictionary.
- **Sample Test Cases:**
 - Input: 5
Output: `{1: 0, 2: 1, 3: 1, 4: 2, 5: 3}`

98. List of Squares of Even Numbers

- **Description:** Given a list of integers, return a list of squares of even numbers.
- **Explanation:** Use list comprehension with condition.
- **Sample Test Cases:**
 - Input: `[1, 2, 3, 4]`
Output: `[4, 16]`

99. Replace Vowels with Position in Alphabet

- **Description:** Replace each vowel in a string with its position in the alphabet.
- **Explanation:** Use dictionary for vowel positions, replace each vowel.
- **Sample Test Cases:**
 - Input: "apple"
Output: "1pp15"

100. Generate Prime Factors of a Number - **Description:** Generate a list of prime factors of a given number. - **Explanation:** Use trial division method to find prime factors.

- **Sample Test Cases:**

- Input: 28
Output: `[2, 7]`
- Input: 45
Output: `[3, 5]`

