Here is the detailed **practice plan** based on the structure you provided, which will guide you through mastering Python and problem-solving step by step, progressively moving from the basics to advanced topics:

**1. Milestone 1: Python Basics**

**Key Concepts Covered:**

• Variables & Data Types

• Input/Output

• Loops

• Conditionals

• Lists

• Strings

• Functions

• Dictionaries

**Practice Plan:**

**Variables and Data Types**

• Swap two numbers (both ways).

• Create variables for different types (int, float, string, list) and perform operations.

• Calculate compound interest.

• Convert seconds to hours, minutes, and seconds.

**Practice Questions:**

• **HackerRank:** [Python If-Else](https://www.hackerrank.com/challenges/python-if-else/problem)

• **LeetCode:** [Running Sum of 1d Array](https://leetcode.com/problems/running-sum-of-1d-array/)

• **CodeChef:** [FLOW001 - Add Two Numbers](https://www.codechef.com/problems/FLOW001)

**Input/Output**

• Write a program to take name, age, and gender as input, then print an introduction.

• Build a basic calculator for addition, subtraction, multiplication, and division.

• Take a sentence input and print the number of vowels and consonants.

**Practice Questions:**

• **HackerRank:** [Say “Hello, World!” With Python](https://www.hackerrank.com/challenges/py-hello-world/problem)

• **CodeChef:** [FLOW007 - Find the Remainder](https://www.codechef.com/problems/FLOW007)

**Loops**

• Print numbers from 1 to N using while and for loops.

• Find the sum of digits of a number using a loop.

• Create a pattern such as Floyd’s triangle.

• Check if a number is prime.

**Practice Questions:**

• **HackerRank:** [Loops](https://www.hackerrank.com/challenges/python-loops/problem)

• **GeeksforGeeks:** [Sum of Series](https://practice.geeksforgeeks.org/problems/sum-of-series2811/1)

**Conditionals**

• Check if a number is divisible by both 3 and 5.

• Determine if a number is a palindrome.

• Implement basic age category checks (child, teenager, adult).

**Practice Questions:**

• **LeetCode:** [Valid Palindrome](https://leetcode.com/problems/valid-palindrome/)

• **HackerRank:** [Nested Lists](https://www.hackerrank.com/challenges/nested-list/problem)

**Lists**

• Sort a list in ascending order.

• Find the maximum and minimum numbers in a list.

• Merge two sorted lists.

• Rotate a list by N positions.

**Practice Questions:**

• **LeetCode:** [Find Pivot Index](https://leetcode.com/problems/find-pivot-index/)

• **GeeksforGeeks:** [Mean and Median](https://www.geeksforgeeks.org/python-mean-median-mode/)

**Strings**

• Reverse a string.

• Count vowels and consonants in a string.

• Check if a string is a palindrome.

• Replace spaces with underscores.

**Practice Questions:**

• **LeetCode:** [Valid Palindrome](https://leetcode.com/problems/valid-palindrome/)

• **HackerRank:** [String Split and Join](https://www.hackerrank.com/challenges/python-string-split-and-join/problem)

**Functions**

• Write a function to check if a number is Armstrong.

• Create a function to calculate the factorial of a number.

• Write a function that returns the sum of squares of the first N natural numbers.

**Practice Questions:**

• **LeetCode:** [Fibonacci Number](https://leetcode.com/problems/fibonacci-number/)

• **GeeksforGeeks:** [Python | Functions](https://www.geeksforgeeks.org/python-functions/)

**Dictionaries**

• Count the frequency of words in a sentence.

• Merge two dictionaries.

• Remove a key from a dictionary.

**Practice Questions:**

• **HackerRank:** [DefaultDict Tutorial](https://www.hackerrank.com/challenges/defaultdict-tutorial/problem)

• **GeeksforGeeks:** [Ways to remove a key from dictionary](https://www.geeksforgeeks.org/python-ways-to-remove-a-key-from-dictionary/)

**2. Milestone 2: Intermediate Python**

**Key Concepts Covered:**

• List Comprehensions

• Nested Data Structures

• String Manipulations

• File Handling

• Error Handling

• Object-Oriented Programming (OOP)

**Practice Plan:**

**List Comprehensions**

• Write a list comprehension to find all even numbers in a list.

• Create a 2D matrix using list comprehension.

**Practice Questions:**

• **LeetCode:** [Transpose Matrix](https://leetcode.com/problems/transpose-matrix/)

• **GeeksforGeeks:** [Python List Comprehension](https://www.geeksforgeeks.org/python-list-comprehension/)

**Nested Data Structures**

• Create a dictionary with lists as values.

• Write a program to flatten a 2D list.

• Implement a basic phone book system with dictionaries and lists.

**Practice Questions:**

• **HackerRank:** [Nested List](https://www.hackerrank.com/challenges/nested-list/problem)

• **LeetCode:** [Merge Intervals](https://leetcode.com/problems/merge-intervals/)

**String Manipulations**

• Implement a function to reverse words in a string.

• Count the frequency of characters in a string using a dictionary.

**Practice Questions:**

• **LeetCode:** [Count and Say](https://leetcode.com/problems/count-and-say/)

• **GeeksforGeeks:** [Python | Ways to check if a string is a palindrome](https://www.geeksforgeeks.org/python-program-to-check-whether-a-string-is-palindrome-or-not/)

**File Handling**

• Write a program to read from and write to a file.

• Count the number of words in a file.

**Practice Questions:**

• **GeeksforGeeks:** [Reading Files](https://www.geeksforgeeks.org/reading-and-writing-files-in-python/)

• **HackerRank:** [File Handling](https://www.hackerrank.com/challenges/python-file-handling/problem)

**Error Handling**

• Write code to handle division by zero error.

• Implement a custom exception.

**Practice Questions:**

• **HackerRank:** [Exceptions](https://www.hackerrank.com/challenges/python-exceptions/problem)

• **GeeksforGeeks:** [Python Exception Handling](https://www.geeksforgeeks.org/python-exception-handling/)

**Object-Oriented Programming (OOP)**

• Create a Person class with attributes like name and age.

• Implement inheritance where a Student inherits from Person.

• Implement polymorphism using method overriding.

**Practice Questions:**

• **LeetCode:** [Implement Stack using Queues](https://leetcode.com/problems/implement-stack-using-queues/)

• **GeeksforGeeks:** [Object Oriented Programming in Python](https://www.geeksforgeeks.org/object-oriented-programming-in-python/)

**3. Milestone 3: Advanced Python and DSA Basics**

**Key Concepts Covered:**

• Recursion

• Lambda Functions

• Functional Programming

• Core Data Structures (Arrays, Stacks, Queues, Linked Lists, Trees)

• Algorithms (Sorting, Searching, Dynamic Programming)

• Time Complexity

**Practice Plan:**

**Recursion**

• Write a recursive function to calculate the factorial of a number.

• Implement the Tower of Hanoi problem.

• Find Fibonacci numbers using recursion.

**Practice Questions:**

• **LeetCode:** [Fibonacci Number](https://leetcode.com/problems/fibonacci-number/)

• **GeeksforGeeks:** [Recursion Problems](https://www.geeksforgeeks.org/recursion/)

**Lambda Functions**

• Use a lambda function to sort a list of tuples by the second value.

• Implement a function that returns a lambda for squaring a number.

**Practice Questions:**

• **GeeksforGeeks:** [Lambda Functions in Python](https://www.geeksforgeeks.org/python-lambda-anonymous-functions-filter-map-reduce/)

**Functional Programming**

• Use filter() and map() to process a list of numbers.

• Implement the reduce() function to compute the product of a list of numbers.

**Practice Questions:**

• **LeetCode:** [Map Sum Pairs](https://leetcode.com/problems/map-sum-pairs/)

• **GeeksforGeeks:** [Functional Programming Concepts](https://www.geeksforgeeks.org/functional-programming-python/)

**Core Data Structures & Algorithms**

• Implement a stack and queue using arrays.

• Implement a binary search tree (BST).

• Solve problems using breadth-first search (BFS) and depth-first search (DFS).

**Practice Questions:**

• **LeetCode:** [Valid Parentheses](https://leetcode.com/problems/valid-parentheses/)

• **GeeksforGeeks:** [Tree Traversals](https://www.geeksforgeeks.org/tree-traversals-inorder-preorder-and-postorder/)

**Dynamic Programming**

• Solve the Fibonacci problem using dynamic programming.

• Solve the Coin Change problem.

**Practice Questions:**

• **LeetCode:** [Climbing Stairs](https://leetcode.com/problems/climbing-stairs/)

• **GeeksforGeeks:** [Dynamic Programming Problems](https://www.geeksforgeeks.org/fundamentals-of-algorithms/#DynamicProgramming)

**Time Complexity**

• Analyze the time complexity of various sorting algorithms.

• Solve problems related to Big-O notation and optimize brute force solutions.

**Practice Questions:**

• **LeetCode:** [Merge Intervals](https://leetcode.com/problems/merge-intervals/)

• **GeeksforGeeks:** [Time Complexity Analysis](https://www.geeksforgeeks.org/analysis-of-algorithms/)

**Additional Resources:**

• **Books:**

• *“Python Crash Course”* by Eric Matthes

• *“Automate the Boring Stuff with Python”* by Al Sweigart

• *“Introduction to Algorithms”* by Cormen et al.

• **Websites and Online Courses:**

• **LeetCode Explore Section:** [LeetCode Explore](https://leetcode.com/explore/)

• **GeeksforGeeks:** [GeeksforGeeks Python Section](https://www.geeksforgeeks.org/python-programming-language/)

• **Coursera (Python for Everybody):** [Python for Everybody](https://www.coursera.org/specializations/python)

By following this comprehensive practice plan, you will systematically build your Python and algorithmic problem-solving skills. Let me know if you’d like further details or modifications!

I understand that you need **real-world problem statements** with **at least 5 practice questions for each concept** within each milestone. Below, I’ve revised the plan to include **real-world problem examples** with the corresponding **coding questions** for every concept, focusing on practical application.

**1. Milestone 1: Python Basics**

**Key Concepts Covered:**

• Variables & Data Types

• Input/Output

• Loops

• Conditionals

• Lists

• Strings

• Functions

• Dictionaries

**Variables and Data Types**

• **Real-World Example:** Storing employee data (name, age, salary, etc.) and performing calculations (e.g., annual salary).

1. Store the details of a person (name, age, salary) and calculate annual salary.

2. Convert temperatures from Celsius to Fahrenheit.

3. Write a program to calculate the area of a circle given the radius.

4. Create a program to calculate the profit or loss from a transaction.

5. Calculate the simple interest based on principal, rate, and time.

**Practice Questions:**

• **HackerRank:** [Data Types](https://www.hackerrank.com/challenges/python-data-types/problem)

• **LeetCode:** [Running Sum of 1d Array](https://leetcode.com/problems/running-sum-of-1d-array/)

• **CodeChef:** [FLOW001 - Add Two Numbers](https://www.codechef.com/problems/FLOW001)

• **GeeksforGeeks:** [Sum of Numbers](https://practice.geeksforgeeks.org/problems/sum-of-numbers/)

• **GeeksforGeeks:** [Python Basics](https://www.geeksforgeeks.org/python-programming-language/)

**Input/Output**

• **Real-World Example:** Taking user input for a list of items and calculating the total price of a shopping list.

1. Write a program that accepts the user’s name and age and prints a greeting message.

2. Create a simple calculator that accepts two numbers and an operator (+, -, \*, /) from the user.

3. Accept a list of items with prices and calculate the total cost.

4. Accept a string of comma-separated words and print them one by one.

5. Take a sentence as input and print the number of words in it.

**Practice Questions:**

• **HackerRank:** [Say “Hello, World!” With Python](https://www.hackerrank.com/challenges/py-hello-world/problem)

• **CodeChef:** [FLOW007 - Find the Remainder](https://www.codechef.com/problems/FLOW007)

• **GeeksforGeeks:** [Input and Output](https://www.geeksforgeeks.org/input-output-in-python/)

• **LeetCode:** [Palindrome Number](https://leetcode.com/problems/palindrome-number/)

• **HackerRank:** [Python If-Else](https://www.hackerrank.com/challenges/python-if-else/problem)

**Loops**

• **Real-World Example:** Looping through a range of numbers to find prime numbers or even numbers.

1. Write a program to print all prime numbers between 1 and 100.

2. Create a program that calculates the factorial of a given number.

3. Print all Fibonacci numbers less than 1000.

4. Find the sum of all even numbers between 1 and 1000.

5. Generate the first N terms of the Fibonacci sequence.

**Practice Questions:**

• **HackerRank:** [Loops](https://www.hackerrank.com/challenges/python-loops/problem)

• **LeetCode:** [Fibonacci Number](https://leetcode.com/problems/fibonacci-number/)

• **GeeksforGeeks:** [Sum of N Natural Numbers](https://practice.geeksforgeeks.org/problems/sum-of-n-naturals/1)

• **CodeChef:** [SUMOFPRO](https://www.codechef.com/problems/SUMOFPRO)

• **GeeksforGeeks:** [Sum of Digits](https://www.geeksforgeeks.org/sum-of-digits-until-it-reduces-to-a-single-digit/)

**Conditionals**

• **Real-World Example:** Checking if a person is eligible to vote based on their age.

1. Check if a number is even or odd.

2. Determine whether a person is eligible to vote (age >= 18).

3. Check if a year is a leap year.

4. Write a program to calculate the grade of a student based on marks.

5. Implement a basic ATM withdrawal system where the user inputs the amount to withdraw.

**Practice Questions:**

• **LeetCode:** [Valid Palindrome](https://leetcode.com/problems/valid-palindrome/)

• **HackerRank:** [Nested Lists](https://www.hackerrank.com/challenges/nested-list/problem)

• **GeeksforGeeks:** [Even or Odd](https://www.geeksforgeeks.org/python-program-to-check-if-a-number-is-even-or-odd/)

• **GeeksforGeeks:** [Leap Year Check](https://www.geeksforgeeks.org/python-program-to-check-if-a-year-is-leap-year-or-not/)

• **CodeChef:** [ATM Problem](https://www.codechef.com/problems/HS08TEST)

**Lists**

• **Real-World Example:** Managing a list of items in an inventory and performing operations such as sorting and removing items.

1. Create a list of numbers and sort them in ascending order.

2. Reverse a list.

3. Remove duplicates from a list.

4. Create a program to merge two lists into one.

5. Write a program that accepts a list of names and sorts them alphabetically.

**Practice Questions:**

• **LeetCode:** [Find Pivot Index](https://leetcode.com/problems/find-pivot-index/)

• **GeeksforGeeks:** [List Operations](https://www.geeksforgeeks.org/python-list/)

• **CodeChef:** [Find Remainder](https://www.codechef.com/problems/FLOW007)

• **HackerRank:** [Lists](https://www.hackerrank.com/challenges/python-lists/problem)

• **GeeksforGeeks:** [Python List Operations](https://www.geeksforgeeks.org/python-list-append/)

**Strings**

• **Real-World Example:** Creating a program that reverses a user input string or checks if the string is a palindrome.

1. Reverse a string without using the built-in reverse function.

2. Count the number of vowels in a string.

3. Check if a string is a palindrome.

4. Replace spaces in a string with hyphens.

5. Count the frequency of each character in a string.

**Practice Questions:**

• **LeetCode:** [Valid Palindrome](https://leetcode.com/problems/valid-palindrome/)

• **GeeksforGeeks:** [Check Palindrome](https://www.geeksforgeeks.org/python-program-to-check-whether-a-string-is-palindrome-or-not/)

• **CodeChef:** [Palindrome String](https://www.codechef.com/problems/PALINDROME)

• **HackerRank:** [String Split and Join](https://www.hackerrank.com/challenges/python-string-split-and-join/problem)

• **GeeksforGeeks:** [String Manipulation](https://www.geeksforgeeks.org/string-manipulation-in-python/)

**Functions**

• **Real-World Example:** Creating functions for basic arithmetic operations such as addition, subtraction, etc.

1. Write a function to calculate the factorial of a number.

2. Create a function that checks whether a number is prime.

3. Write a function to find the greatest common divisor (GCD) of two numbers.

4. Implement a function to calculate the power of a number.

5. Write a function that checks if a string is an anagram of another string.

**Practice Questions:**

• **LeetCode:** [Fibonacci Number](https://leetcode.com/problems/fibonacci-number/)

• **HackerRank:** [Functions](https://www.hackerrank.com/challenges/py-functions/problem)

• **GeeksforGeeks:** [Factorial Function](https://www.geeksforgeeks.org/python-program-to-find-the-factorial-of-a-number/)

• **CodeChef:** [Factorial Program](https://www.codechef.com/problems/FLOW006)

• **GeeksforGeeks:** [Find GCD](https://www.geeksforgeeks.org/python-program-to-find-gcd-or-hcf-of-two-numbers/)

**2. Milestone 2: Intermediate Python**

**String Manipulations**

• **Real-World Example:** Processing user input for email addresses, phone numbers, or formatting strings for display.

1. Validate an email address (basic format check).

2. Extract domain names from email addresses.

3. Implement a string formatting system for invoices (e.g., aligning values in columns).

4. Write a program to check for anagrams between two strings.

5. Create a program to replace all vowels in a string with a specified character.

**Practice Questions:**

• **LeetCode:** [Valid Palindrome](https://leetcode.com/problems/valid-palindrome/)

• **GeeksforGeeks:** [String Manipulation](https://www.geeksforgeeks.org/string-manipulation-in-python/)

• **CodeChef:** [Reverse String](https://www.codechef.com/problems/REVERSE)

• **HackerRank:** [String Validators](https://www.hackerrank.com/challenges/string-validators/problem)

• **LeetCode:** [Anagram](https://leetcode.com/problems/group-anagrams/)

**File Handling**

• **Real-World Example:** Reading, writing, and modifying files (e.g., processing logs, CSV files, or saving user data).

1. Write a program to read from a text file and count the occurrences of each word.

2. Create a program that saves user data (name, age, email) into a CSV file.

3. Implement a program that extracts specific data from a CSV file and displays it.

4. Create a log file where each action taken by a user is logged with timestamps.

5. Implement a program that merges multiple text files into one.

**Practice Questions:**

• **HackerRank:** [Python File I/O](https://www.hackerrank.com/challenges/py-file-io/problem)

• **LeetCode:** [Read N Characters Given Read4](https://leetcode.com/problems/read-n-characters-given-read4/)

• **GeeksforGeeks:** [File Handling in Python](https://www.geeksforgeeks.org/file-handling-python/)

• **CodeChef:** [File Handling Practice](https://www.codechef.com/problems/READFILE)

• **GeeksforGeeks:** [Reading CSV File](https://www.geeksforgeeks.org/working-csv-files-python/)

**Error Handling**

• **Real-World Example:** Handling user input errors and exceptions in real-time applications like calculators or data-processing programs.

1. Implement a program that asks for user input and handles invalid input gracefully (e.g., non-numeric input for age).

2. Write a program that catches division by zero errors and provides feedback to the user.

3. Handle file not found errors when trying to open a file.

4. Implement a calculator program that handles incorrect operator input.

5. Create a program that catches value errors while trying to convert user input.

**Practice Questions:**

• **GeeksforGeeks:** [Exception Handling in Python](https://www.geeksforgeeks.org/python-exception-handling/)

• **HackerRank:** [Exceptions](https://www.hackerrank.com/challenges/python-exceptions/problem)

• **LeetCode:** [Divide Two Integers](https://leetcode.com/problems/divide-two-integers/)

• **CodeChef:** [Error Handling Practice](https://www.codechef.com/problems/ERRHAND)

• **GeeksforGeeks:** [Try Except Block](https://www.geeksforgeeks.org/python-try-except/)

**Object-Oriented Programming (OOP)**

• **Real-World Example:** Modeling real-world entities using classes and objects (e.g., creating a bank account or employee class).

1. Create a class representing a BankAccount with methods to deposit, withdraw, and check balance.

2. Implement a class for a Book with attributes like title, author, and price. Add methods to display book details.

3. Design a class Car with attributes such as make, model, and mileage, and methods for acceleration and braking.

4. Create a class Employee with attributes such as name, salary, and department, and implement a method to increase salary.

5. Implement an inheritance model where Manager inherits from Employee, adding extra functionalities (like managing a team).

**Practice Questions:**

• **LeetCode:** [Design Parking System](https://leetcode.com/problems/design-parking-system/)

• **GeeksforGeeks:** [OOP Concepts in Python](https://www.geeksforgeeks.org/python-object-oriented-programming-oops-concept/)

• **CodeChef:** [OOP Practice](https://www.codechef.com/problems/CLASSES)

• **HackerRank:** [Classes](https://www.hackerrank.com/challenges/py-class-1/problem)

• **GeeksforGeeks:** [Inheritance in Python](https://www.geeksforgeeks.org/inheritance-in-python/)

**3. Milestone 3: Advanced Python and DSA Basics**

**Key Concepts Covered:**

• Recursion

• Lambda Functions

• Functional Programming

• Core Data Structures (Arrays, Stacks, Queues, Linked Lists, Trees)

• Algorithms (Sorting, Searching, Dynamic Programming)

**Recursion**

• **Real-World Example:** Solving complex problems by breaking them down into smaller subproblems (e.g., solving a maze, calculating Fibonacci numbers).

1. Write a recursive function to calculate the factorial of a number.

2. Implement a recursive function to solve the Fibonacci sequence.

3. Solve the Towers of Hanoi problem recursively.

4. Write a recursive program to find the greatest common divisor (GCD) of two numbers.

5. Implement a recursive function to calculate the sum of digits of a number.

**Practice Questions:**

• **LeetCode:** [Climbing Stairs](https://leetcode.com/problems/climbing-stairs/)

• **GeeksforGeeks:** [Recursion Examples](https://www.geeksforgeeks.org/recursion/)

• **HackerRank:** [Recursion Fibonacci](https://www.hackerrank.com/challenges/python-recursion/problem)

• **CodeChef:** [Recursion Practice](https://www.codechef.com/problems/RECFACT)

• **GeeksforGeeks:** [Towers of Hanoi](https://www.geeksforgeeks.org/c-program-for-tower-of-hanoi/)

**Lambda Functions**

• **Real-World Example:** Implementing small anonymous functions for operations like sorting or filtering lists.

1. Use a lambda function to sort a list of tuples based on the second element.

2. Write a lambda function that checks if a number is even or odd.

3. Use lambda and filter() to return only even numbers from a list.

4. Sort a list of strings by their length using a lambda function.

5. Write a lambda function that adds 10 to a given number.

**Practice Questions:**

• **GeeksforGeeks:** [Lambda Functions](https://www.geeksforgeeks.org/python-lambda-anonymous-functions-filter-map-reduce/)

• **LeetCode:** [Sort Characters By Frequency](https://leetcode.com/problems/sort-characters-by-frequency/)

• **HackerRank:** [Lambda Expressions](https://www.hackerrank.com/challenges/python-lambda/problem)

• **CodeChef:** [Lambda Function Practice](https://www.codechef.com/problems/LAMBDA)

• **GeeksforGeeks:** [Lambda Function Examples](https://www.geeksforgeeks.org/python-lambda-function/)

**Core Data Structures (Arrays, Stacks, Queues, Linked Lists, Trees)**

• **Real-World Example:** Managing tasks or orders in a system using data structures like queues, processing items using stacks.

1. Implement a queue and simulate a checkout line.

2. Write a program that reverses a list using a stack.

3. Implement a linked list and perform insertion and deletion at the beginning, middle, and end.

4. Create a binary search tree and implement basic operations (insert, search, delete).

5. Implement a stack that supports both push and pop operations and keeps track of the minimum element.

**Practice Questions:**

• **LeetCode:** [Implement Queue using Stacks](https://leetcode.com/problems/implement-queue-using-stacks/)

• **GeeksforGeeks:** [Linked List Operations](https://www.geeksforgeeks.org/data-structures/linked-list/)

• **CodeChef:** [Data Structure Practice](https://www.codechef.com/problems/DSA1)

• **HackerRank:** [Queue using Two Stacks](https://www.hackerrank.com/challenges/queue-using-two-stacks/problem)

• **GeeksforGeeks:** [Binary Search Tree](https://www.geeksforgeeks.org/binary-search-tree-data-structure/)

**Algorithms (Sorting, Searching, Dynamic Programming)**

• **Real-World Example:** Implementing search algorithms in a product catalog, or dynamic programming for calculating the minimum number of operations.

1. Implement Bubble Sort and compare its performance with Quick Sort.

2. Use binary search to find the position of an element in a sorted array.

3. Implement dynamic programming to solve the Knapsack problem.

4. Use dynamic programming to find the longest common subsequence between two strings.

5. Implement Dijkstra’s algorithm for finding the shortest path in a graph.

**Practice Questions:**

• **LeetCode:** [Knapsack Problem](https://leetcode.com/problems/knapsack/)

• **HackerRank:** [Sorting Algorithms](https://www.hackerrank.com/domains/tutorials/10-days-of-javascript)

• **CodeChef:** [Dynamic Programming Practice](https://www.codechef.com/problems/DP)

• **GeeksforGeeks:** [Dijkstra’s Algorithm](https://www.geeksforgeeks.org/implementing-dijkstra-algorithm-using-priority-queue/)

• **LeetCode:** [Longest Common Subsequence](https://leetcode.com/problems/longest-common-subsequence/)

This plan gives you a comprehensive set of exercises and real-world examples for each milestone, which should help you build a solid foundation for solving LeetCode problems. Would you like to focus on specific areas or need additional support on any particular topic?

**Milestone 1: Python Basics**

**1. Variables & Data Types**

1. Assign values to variables of different types (integer, float, string).

2. Perform arithmetic operations with integers and floats.

3. Swap two variables without using a third variable.

4. Convert between data types (string to integer, float to string).

5. Write a program that calculates the area of a circle using radius.

6. Check whether a variable is of a specific data type.

7. Parse input from the user and convert it to the appropriate type.

8. Calculate the sum of squares of two numbers entered by the user.

9. Check if a number is positive, negative, or zero.

10. Create a program that outputs a sentence using concatenation of variables.

11. Add, subtract, multiply, and divide two floating-point numbers.

12. Implement type conversion for a user’s input.

13. Store multiple values in a tuple and access individual elements.

14. Use is and == to compare two variables.

15. Handle division by zero error using exception handling.

16. Create a program that checks if a given number is an integer or float.

17. Input and output a complex number in Python.

18. Create variables to store the area, length, and breadth of a rectangle.

19. Find the average of three numbers.

20. Check if a number is divisible by another number.

21. Find the largest of three numbers.

22. Print the ASCII value of a character entered by the user.

23. Create variables to store temperature and convert it between Celsius and Fahrenheit.

24. Calculate the volume of a cylinder using input dimensions.

25. Implement a program that checks if two strings are equal or not.

**Platform Questions for Variables and Data Types:**

• **LeetCode:** [Number of 1 Bits](https://leetcode.com/problems/number-of-1-bits/)

• **HackerRank:** [Python Data Types](https://www.hackerrank.com/challenges/python-data-types/problem)

• **GeeksforGeeks:** [Python Variables](https://www.geeksforgeeks.org/python-variables/)

• **CodeChef:** [Celsius to Fahrenheit](https://www.codechef.com/problems/TEMPCONV)

• **GeeksforGeeks:** [Sum of Numbers](https://www.geeksforgeeks.org/python-program-to-sum-of-squares-of-two-numbers/)

**2. Loops & Conditionals**

1. Print numbers from 1 to 10 using a for loop.

2. Print even numbers from 1 to 100 using a for loop.

3. Create a program that counts from 1 to 50, but prints “Fizz” for multiples of 3 and “Buzz” for multiples of 5.

4. Implement a while loop to display all prime numbers less than 100.

5. Write a program that calculates the factorial of a number using a loop.

6. Write a program that sums all numbers from 1 to n using a while loop.

7. Use if and elif to check if a number is divisible by both 3 and 5.

8. Write a program that prints whether a number is prime or not.

9. Check if a year is a leap year or not.

10. Print a multiplication table for a given number using a for loop.

11. Find the largest of three numbers using if statements.

12. Write a program to check if a given string is a palindrome.

13. Calculate the number of vowels and consonants in a given string.

14. Reverse a string using a loop.

15. Implement a simple calculator using a loop and conditionals.

16. Find the smallest number in a list using a loop.

17. Write a program to find the sum of digits of a number.

18. Implement a program that prints the Fibonacci sequence up to the nth term.

19. Write a program to count the number of occurrences of a character in a string.

20. Write a program to print the factorial of a number recursively.

21. Check if a number is even or odd using an if statement.

22. Print a pattern of stars using a for loop (e.g., pyramid pattern).

23. Find the length of a string without using the built-in len() function.

24. Write a program to find the second largest number in a list.

25. Create a number guessing game using loops and conditionals.

**Platform Questions for Loops and Conditionals:**

• **LeetCode:** [Palindrome Number](https://leetcode.com/problems/palindrome-number/)

• **HackerRank:** [Loops](https://www.hackerrank.com/challenges/python-loops/problem)

• **GeeksforGeeks:** [Python If-Else Statements](https://www.geeksforgeeks.org/python-if-else/)

• **CodeChef:** [Number Guessing](https://www.codechef.com/problems/NUMGUESS)

• **GeeksforGeeks:** [Prime Number Checker](https://www.geeksforgeeks.org/python-program-to-check-whether-a-number-is-prime-or-not/)

**3. Lists & Strings**

1. Create a list with 5 elements and print the first and last elements.

2. Append an element to a list and print the updated list.

3. Reverse a list using a built-in function.

4. Find the sum of all elements in a list.

5. Count the occurrences of an element in a list.

6. Sort a list of integers in ascending order.

7. Remove duplicates from a list without using set().

8. Write a program to merge two lists.

9. Write a function to check if an item exists in a list.

10. Find the maximum and minimum values in a list.

11. Write a program that accepts a string and prints each character one by one.

12. Reverse a string without using the reverse() method.

13. Concatenate two strings.

14. Implement a program that converts a string to lowercase and uppercase.

15. Write a program to remove whitespaces from a string.

16. Check if a string is a palindrome.

17. Count the number of vowels in a string.

18. Find the length of a string without using the built-in len() function.

19. Check if a string is an anagram of another string.

20. Find the index of a character in a string.

21. Implement string slicing to extract substrings.

22. Convert a string to a list and then back to a string.

23. Join a list of strings into one string with a space separator.

24. Split a string into a list of words.

25. Replace all spaces in a string with underscores.

**Platform Questions for Lists and Strings:**

• **LeetCode:** [Reverse String](https://leetcode.com/problems/reverse-string/)

• **GeeksforGeeks:** [Python List Operations](https://www.geeksforgeeks.org/python-list-append/)

• **HackerRank:** [String Manipulation](https://www.hackerrank.com/challenges/string-validators/problem)

• **CodeChef:** [String Comparison](https://www.codechef.com/problems/STRINGCMP)

• **LeetCode:** [Anagram](https://leetcode.com/problems/group-anagrams/)

**Milestone 2: Intermediate Python (Continued)**

**2. Nested Data Structures (Lists, Dictionaries, Sets)**

1. Create a nested list and access elements from it.

2. Create a dictionary and access values using keys.

3. Add a new key-value pair to a dictionary.

4. Write a program to count the occurrences of each character in a string using a dictionary.

5. Implement a program to find common elements between two lists using dictionaries.

6. Merge two dictionaries.

7. Create a list of lists, and find the maximum element in each inner list.

8. Write a program to create a dictionary with key-value pairs representing names and ages.

9. Implement a nested dictionary and extract specific information.

10. Write a program to check if a key exists in a dictionary.

11. Find the frequency of words in a sentence using a dictionary.

12. Iterate over a list of tuples and create a dictionary from it.

13. Find the average of a list of numbers using nested loops.

14. Flatten a dictionary of lists into a single list.

15. Write a program to get the most common character in a string.

16. Convert a dictionary into a list of tuples.

17. Write a function that counts how many times each number occurs in a list of lists.

18. Implement a program to merge multiple lists into a single list.

19. Use a set to remove duplicate elements from a list.

20. Create a program that finds all unique characters in a string using a set.

21. Create a dictionary and implement a function to update its values.

22. Create a set of numbers and add more numbers to the set.

23. Find the union and intersection of two sets.

24. Check if an element exists in a set.

25. Write a program to calculate the intersection of two lists using sets.

**Platform Questions for Nested Data Structures:**

• **LeetCode:** [Group Anagrams](https://leetcode.com/problems/group-anagrams/)

• **HackerRank:** [Nested Lists](https://www.hackerrank.com/challenges/nested-list/problem)

• **GeeksforGeeks:** [Flatten Nested List](https://www.geeksforgeeks.org/python-program-to-flatten-a-nested-list/)

• **CodeChef:** [Dictionary Usage](https://www.codechef.com/problems/DICTHELP)

• **GeeksforGeeks:** [Dictionary Iteration](https://www.geeksforgeeks.org/iterate-through-a-dictionary-in-python/)

**3. Recursion**

1. Write a recursive function to calculate the factorial of a number.

2. Implement a recursive function to find the nth Fibonacci number.

3. Write a program to print numbers from 1 to n using recursion.

4. Implement a recursive function to reverse a string.

5. Solve the Tower of Hanoi problem using recursion.

6. Implement a program to calculate the sum of elements in a list recursively.

7. Write a recursive function to find the greatest common divisor (GCD) of two numbers.

8. Create a recursive function to count the number of elements in a list.

9. Implement a recursive function to find the length of a string.

10. Write a program to check if a string is a palindrome using recursion.

11. Implement a program to calculate the power of a number using recursion.

12. Write a function that prints all elements of a list in reverse order using recursion.

13. Solve the problem of summing digits of a number using recursion.

14. Write a recursive function that prints the Fibonacci sequence up to n.

15. Implement a recursive function to search for an element in a list.

16. Solve the problem of generating all subsets of a given set recursively.

17. Write a program to find the minimum element in a list using recursion.

18. Implement a recursive function that computes the sum of squares of numbers from 1 to n.

19. Create a recursive function to find the factorial of a number iteratively and recursively.

20. Implement a recursive program to calculate the length of a list.

21. Write a function that reverses a list using recursion.

22. Solve a problem of searching a specific element in a nested list using recursion.

23. Implement a program that calculates the sum of numbers in a matrix recursively.

24. Create a function that performs a binary search on a sorted list using recursion.

25. Write a program to print the sum of all elements in a nested list recursively.

**Platform Questions for Recursion:**

• **LeetCode:** [Climbing Stairs](https://leetcode.com/problems/climbing-stairs/)

• **HackerRank:** [Recursion and Backtracking](https://www.hackerrank.com/challenges/recursion-coding/problem)

• **GeeksforGeeks:** [Recursion Practice](https://www.geeksforgeeks.org/recursion/)

• **CodeChef:** [Factorial Recursion](https://www.codechef.com/problems/FCTR)

• **LeetCode:** [Permutations](https://leetcode.com/problems/permutations/)

**Milestone 3: Advanced Python and DSA Basics**

**1. Dynamic Programming**

1. Implement a program to calculate the Fibonacci number using dynamic programming.

2. Solve the problem of finding the maximum sum subarray using dynamic programming.

3. Solve the problem of coin change with a given amount using dynamic programming.

4. Implement the Longest Common Subsequence (LCS) problem.

5. Solve the Knapsack problem using dynamic programming.

6. Solve the problem of finding the minimum number of steps to reach the end of an array.

7. Write a program that finds the longest increasing subsequence in a given sequence.

8. Implement the problem of counting the number of ways to climb stairs (DP approach).

9. Solve the problem of finding the number of distinct ways to reach the top of a staircase.

10. Solve the problem of the longest palindrome substring using dynamic programming.

11. Write a program that calculates the minimum path sum in a grid using DP.

12. Implement a dynamic programming solution to solve the subset sum problem.

13. Solve the problem of finding the maximum profit in a stock buying and selling scenario.

14. Write a program that counts the number of distinct ways to partition a set using DP.

15. Solve the problem of maximum product subarray using dynamic programming.

16. Implement the problem of counting unique binary search trees using DP.

17. Write a program that finds the minimum number of coins needed to make change.

18. Solve the problem of word break using dynamic programming.

19. Implement the problem of finding the number of ways to reach the end of a grid.

20. Write a program to calculate the minimum edit distance between two strings using DP.

21. Solve the problem of the longest common substring using dynamic programming.

22. Implement the problem of finding the number of ways to make change for a given value.

23. Solve the rod cutting problem using dynamic programming.

24. Implement the problem of calculating the maximum sum of non-adjacent numbers.

25. Write a program that finds the maximum product of two integers in an array.

**Platform Questions for Dynamic Programming:**

• **LeetCode:** [House Robber](https://leetcode.com/problems/house-robber/)

• **GeeksforGeeks:** [Longest Common Subsequence](https://www.geeksforgeeks.org/longest-common-subsequence-dp-4/)

• **HackerRank:** [Coin Change Problem](https://www.hackerrank.com/challenges/coin-change/problem)

• **CodeChef:** [Knapsack Problem](https://www.codechef.com/problems/KNAPSACK)

• **LeetCode:** [Climbing Stairs](https://leetcode.com/problems/climbing-stairs/)

**2. Algorithms (Searching & Sorting)**

1. Implement the binary search algorithm on a sorted list.

2. Write a program to implement the bubble sort algorithm.

3. Implement the insertion sort algorithm.

4. Write a program that implements the merge sort algorithm.

5. Implement the quicksort algorithm.

6. Write a program to implement the selection sort algorithm.

7. Implement the heap sort algorithm.

8. Write a program to find the kth largest element in an unsorted list.

9. Implement linear search on an unsorted list.

10. Write a program to search for an element in a rotated sorted array using binary search.

11. Solve the problem of sorting a list of strings based on their length.

12. Implement the counting sort algorithm.

13. Implement a binary search tree (BST) and perform basic operations.

14. Write a program that sorts a list of numbers using radix sort.

15. Implement the shell sort algorithm.

16. Write a program that finds the majority element in an array.

17. Implement the partition function of quicksort.

18. Write a program that sorts a list of tuples based on the second element of the tuple.

19. Solve the problem of finding the first missing positive integer in an unsorted array.

20. Implement an algorithm to find the intersection of two sorted arrays.

21. Write a program to merge two sorted arrays into one.

22. Implement a program to find the median of two sorted arrays.

23. Solve the problem of finding the closest pair in an unsorted list.

24. Write a program to implement the top-down approach of merge sort.

25. Implement the counting sort algorithm to sort a list of integers.

**Platform Questions for Searching & Sorting:**

• **LeetCode:** [Binary Search](https://leetcode.com/problems/binary-search/)

• **HackerRank:** [Sorting](https://www.hackerrank.com/domains/tutorials/30-days-of-code)

• **GeeksforGeeks:** [Sorting Algorithms](https://www.geeksforgeeks.org/sorting-algorithms/)

• **CodeChef:** [Sorting](https://www.codechef.com/problems/SORTING)

• **LeetCode:** [Merge Intervals](https://leetcode.com/problems/merge-intervals/)

Here are 25 questions for each of the concepts in **Milestone 1: Python Basics** from various coding platforms to help you practice and build a strong foundation in Python.

**Milestone 1: Python Basics**

**1. Variables & Data Types**

1. **LeetCode:** [Python Data Types](https://leetcode.com/problems/python-data-types/)

• Description: Work with basic Python data types and their conversions.

2. **HackerRank:** [Python Integers](https://www.hackerrank.com/challenges/python-integers/problem)

• Description: Work with integer values and arithmetic in Python.

3. **GeeksforGeeks:** [Check if Two Numbers are Equal](https://www.geeksforgeeks.org/check-two-numbers-equal-python/)

• Description: Compare two numbers using Python.

4. **LeetCode:** [Valid Number](https://leetcode.com/problems/valid-number/)

• Description: Validate if a string represents a valid number.

5. **CodeChef:** [Variables Input](https://www.codechef.com/problems/INPSTR)

• Description: Work with user input and print variables.

6. **HackerRank:** [String to Integer](https://www.hackerrank.com/challenges/python-string-to-integer/problem)

• Description: Convert a string into an integer in Python.

7. **LeetCode:** [Convert a Number to Hexadecimal](https://leetcode.com/problems/convert-a-number-to-hexadecimal/)

• Description: Convert an integer to a hexadecimal string.

8. **GeeksforGeeks:** [Sum of Two Numbers](https://www.geeksforgeeks.org/python-program-to-find-sum-of-two-numbers/)

• Description: Perform basic arithmetic operations.

9. **HackerRank:** [Basic Data Types](https://www.hackerrank.com/challenges/python-basic-data-types/problem)

• Description: Work with basic data types in Python.

10. **CodeChef:** [Type Conversion](https://www.codechef.com/problems/TYPES)

• Description: Practice type conversion in Python.

11. **LeetCode:** [Find the Difference of Two Strings](https://leetcode.com/problems/find-the-difference-of-two-strings/)

• Description: Compare two strings and determine their difference.

12. **HackerRank:** [Floating Point Precision](https://www.hackerrank.com/challenges/python-floating-point/problem)

• Description: Work with floating point numbers and precision.

13. **GeeksforGeeks:** [Swap Two Numbers](https://www.geeksforgeeks.org/swap-two-numbers-in-python-without-using-third-variable/)

• Description: Swap values of two numbers.

14. **LeetCode:** [Palindrome Number](https://leetcode.com/problems/palindrome-number/)

• Description: Check if an integer is a palindrome.

15. **CodeChef:** [Multiple of 3](https://www.codechef.com/problems/MUL3)

• Description: Check if a number is divisible by 3.

16. **LeetCode:** [Roman to Integer](https://leetcode.com/problems/roman-to-integer/)

• Description: Convert a Roman numeral to an integer.

17. **HackerRank:** [Python Division](https://www.hackerrank.com/challenges/python-division/problem)

• Description: Work with integer division and float division.

18. **GeeksforGeeks:** [Number of Digits](https://www.geeksforgeeks.org/count-digits-in-an-integer-in-python/)

• Description: Find the number of digits in an integer.

19. **LeetCode:** [Count Primes](https://leetcode.com/problems/count-primes/)

• Description: Count how many prime numbers exist less than a given number.

20. **HackerRank:** [Arithmetic Operators](https://www.hackerrank.com/challenges/python-arithmetic-operators/problem)

• Description: Perform basic arithmetic operations.

21. **GeeksforGeeks:** [Sum of Digits](https://www.geeksforgeeks.org/sum-of-digits-until-it-becomes-single-digit/)

• Description: Sum digits of a number until it becomes a single digit.

22. **CodeChef:** [Check Even or Odd](https://www.codechef.com/problems/EO)

• Description: Check if a number is even or odd.

23. **LeetCode:** [Sum of Square Numbers](https://leetcode.com/problems/sum-of-square-numbers/)

• Description: Find if a number can be expressed as the sum of squares of two integers.

24. **HackerRank:** [List Manipulation](https://www.hackerrank.com/challenges/python-list/problem)

• Description: Practice list manipulation and basic operations.

25. **CodeChef:** [Checking Prime](https://www.codechef.com/problems/PRIME1)

• Description: Check if a number is prime.

**2. Input/Output**

1. **LeetCode:** [Count and Say](https://leetcode.com/problems/count-and-say/)

• Description: Output the count and say sequence for a given number.

2. **GeeksforGeeks:** [Read User Input](https://www.geeksforgeeks.org/taking-input-in-python/)

• Description: Learn how to take user input in Python.

3. **HackerRank:** [Python Input](https://www.hackerrank.com/challenges/python-input/problem)

• Description: Practice taking multiple inputs in a single line.

4. **LeetCode:** [Print Integer in Binary](https://leetcode.com/problems/convert-integer-to-the-sum-of-two-no-zero-integers/)

• Description: Convert a number to its binary form.

5. **GeeksforGeeks:** [Write to File](https://www.geeksforgeeks.org/reading-and-writing-files-in-python/)

• Description: Practice reading and writing data to files.

6. **HackerRank:** [Python Output](https://www.hackerrank.com/challenges/python-print/problem)

• Description: Output the given text or data to the console.

7. **LeetCode:** [Fizz Buzz](https://leetcode.com/problems/fizz-buzz/)

• Description: Print numbers 1 to N, but for multiples of 3 print “Fizz” and for multiples of 5 print “Buzz”.

8. **CodeChef:** [Print a Number](https://www.codechef.com/problems/PRINTNUM)

• Description: Print a number based on the given input.

9. **GeeksforGeeks:** [Taking Multiple Inputs](https://www.geeksforgeeks.org/taking-multiple-inputs-from-user-in-python/)

• Description: Take multiple inputs from the user in a single line.

10. **HackerRank:** [Print Function](https://www.hackerrank.com/challenges/python-print/problem)

• Description: Output the given number in the desired format.

11. **LeetCode:** [Reverse Words in a String](https://leetcode.com/problems/reverse-words-in-a-string/)

• Description: Reverse words in a string and print them.

12. **GeeksforGeeks:** [String Formatting](https://www.geeksforgeeks.org/format-strings-python/)

• Description: Learn string formatting for user output.

13. **HackerRank:** [Output Formatting](https://www.hackerrank.com/challenges/python-division/problem)

• Description: Format output and control precision.

14. **LeetCode:** [Palindrome Check](https://leetcode.com/problems/valid-palindrome/)

• Description: Check if a string is a palindrome.

15. **GeeksforGeeks:** [Count Words in a Sentence](https://www.geeksforgeeks.org/python-program-to-count-words-in-a-string/)

• Description: Count the number of words in a string.

16. **HackerRank:** [Array Manipulation](https://www.hackerrank.com/challenges/array-manipulation/problem)

• Description: Output the manipulated array after performing operations.

17. **LeetCode:** [Print the Matrix](https://leetcode.com/problems/spiral-matrix/)

• Description: Output elements of a matrix in a spiral order.

18. **CodeChef:** [Sum of Digits](https://www.codechef.com/problems/SUMD)

• Description: Print the sum of digits of a number.

19. **HackerRank:** [List Comprehensions](https://www.hackerrank.com/challenges/list-comprehensions/problem)

• Description: Practice list comprehensions and their use in formatting output.

20. **GeeksforGeeks:** [Input and Output](https://www.geeksforgeeks.org/input-output-functions-python/)

• Description: Learn advanced input/output functions in Python.

21. **LeetCode:** [Longest Substring Without Repeating Characters](https://leetcode.com/problems/longest-substring-without-repeating-characters/)

• Description: Print the length of the longest substring without repeating characters.

22. **GeeksforGeeks:** [Python Format String](https://www.geeksforgeeks.org/python-format-function/)

• Description: Format strings in Python for user output.

23. **LeetCode:** [Merge Two Sorted Lists](https://leetcode.com/problems/merge-two-sorted-lists/)

• Description: Print the merged list of two sorted lists.

24. **CodeChef:** [Pattern Printing](https://www.codechef.com/problems/PRTPAT)

• Description: Print a specific pattern using loops.

25. **HackerRank:** [Formatted Output](https://www.hackerrank.com/challenges/python-division/problem)

• Description: Format numbers as output in a specific format.

**3. Loops**

1. **LeetCode:** [Pascal’s Triangle](https://leetcode.com/problems/pascals-triangle/)

• Description: Print Pascal’s triangle up to a certain number of rows using loops.

2. **GeeksforGeeks:** [Sum of Natural Numbers](https://www.geeksforgeeks.org/python-program-to-find-sum-of-natural-numbers/)

• Description: Write a program to find the sum of natural numbers.

3. **HackerRank:** [Loops in Python](https://www.hackerrank.com/challenges/python-loops/problem)

• Description: Practice loops and iteration over numbers.

4. **LeetCode:** [Print Numbers in Reverse](https://leetcode.com/problems/counting-bits/)

• Description: Print numbers in reverse using a loop.

5. **CodeChef:** [Even Odd Count](https://www.codechef.com/problems/EO)

• Description: Count the number of even and odd numbers from a list.

6. **GeeksforGeeks:** [Prime Numbers using Loop](https://www.geeksforgeeks.org/python-program-to-print-prime-numbers-in-a-range/)

• Description: Print prime numbers using loops.

7. **HackerRank:** [Print the Squares](https://www.hackerrank.com/challenges/python-loops/problem)

• Description: Print squares of numbers from 1 to n using a loop.

8. **LeetCode:** [Happy Number](https://leetcode.com/problems/happy-number/)

• Description: Check if a number is happy using loops.

9. **GeeksforGeeks:** [Fibonacci Series using Loop](https://www.geeksforgeeks.org/python-program-for-fibonacci-series/)

• Description: Print the Fibonacci series using loops.

10. **CodeChef:** [Pattern Printing](https://www.codechef.com/problems/PRTPAT)

• Description: Print a specified pattern using loops.

11. **HackerRank:** [Even Numbers](https://www.hackerrank.com/challenges/py-hello-world/problem)

• Description: Print all even numbers between a given range.

12. **LeetCode:** [Find the Largest Element](https://leetcode.com/problems/maximum-subarray/)

• Description: Find the largest element in a list using a loop.

13. **GeeksforGeeks:** [Count Digits in a Number](https://www.geeksforgeeks.org/count-number-of-digits-in-an-integer-in-python/)

• Description: Count the number of digits in a number using loops.

14. **HackerRank:** [Multiplication Tables](https://www.hackerrank.com/challenges/python-multiplication/problem)

• Description: Print the multiplication table using loops.

15. **LeetCode:** [Reverse Array](https://leetcode.com/problems/reverse-array/)

• Description: Reverse an array using loops.

16. **GeeksforGeeks:** [Sum of Elements in List](https://www.geeksforgeeks.org/python-program-to-find-the-sum-of-elements-in-list/)

• Description: Sum elements of a list using a loop.

17. **CodeChef:** [Factorial Calculation](https://www.codechef.com/problems/FACTORIAL)

• Description: Calculate the factorial of a number using a loop.

18. **HackerRank:** [Range of Numbers](https://www.hackerrank.com/challenges/python-range/problem)

• Description: Work with the range function in loops.

19. **LeetCode:** [Find Maximum Sum Subarray](https://leetcode.com/problems/maximum-subarray/)

• Description: Use loops to find the maximum sum subarray.

20. **GeeksforGeeks:** [Reverse a String](https://www.geeksforgeeks.org/python-program-to-reverse-a-string/)

• Description: Reverse a string using loops.

21. **HackerRank:** [Sum of N Numbers](https://www.hackerrank.com/challenges/python-loops/problem)

• Description: Find the sum of N numbers using a loop.

22. **LeetCode:** [Palindrome Check using Loop](https://leetcode.com/problems/valid-palindrome/)

• Description: Check if a string is a palindrome using loops.

23. **GeeksforGeeks:** [GCD using Loop](https://www.geeksforgeeks.org/python-program-to-find-gcd-or-hcf-of-two-numbers/)

• Description: Calculate the GCD of two numbers using loops.

24. **LeetCode:** [Missing Number](https://leetcode.com/problems/missing-number/)

• Description: Find the missing number in an array using loops.

25. **CodeChef:** [Sum of Odd Numbers](https://www.codechef.com/problems/SUMODD)

• Description: Find the sum of odd numbers in a range using loops.

**4. Conditionals**

1. **LeetCode:** [Fizz Buzz](https://leetcode.com/problems/fizz-buzz/)

• Description: Print numbers from 1 to N, but for multiples of 3 print “Fizz” and for multiples of 5 print “Buzz”.

2. **HackerRank:** [Python If-Else](https://www.hackerrank.com/challenges/py-if-else/problem)

• Description: Implement if-else conditions in Python.

3. **GeeksforGeeks:** [Largest of Three Numbers](https://www.geeksforgeeks.org/python-program-to-find-largest-of-three-numbers/)

• Description: Find the largest of three numbers using conditional statements.

4. **LeetCode:** [Valid Palindrome](https://leetcode.com/problems/valid-palindrome/)

• Description: Check if a string is a valid palindrome using conditions.

5. **CodeChef:** [Age Eligibility](https://www.codechef.com/problems/AGEELIG)

• Description: Determine eligibility for a service based on age using if-else conditions.

6. **GeeksforGeeks:** [Check if Number is Even or Odd](https://www.geeksforgeeks.org/python-program-to-check-whether-a-number-is-even-or-odd/)

• Description: Check if a number is even or odd using conditions.

7. **LeetCode:** [Even Odd Separation](https://leetcode.com/problems/separate-even-and-odd-values-in-array/)

• Description: Separate even and odd values in an array using conditional statements.

8. **HackerRank:** [Odd or Even](https://www.hackerrank.com/challenges/odd-or-even/problem)

• Description: Determine if a number is odd or even.

9. **GeeksforGeeks:** [Grade Classification](https://www.geeksforgeeks.org/python-program-to-classify-student-grade/)

• Description: Classify a student’s grade based on marks using conditions.

10. **CodeChef:** [Find Maximum of Three Numbers](https://www.codechef.com/problems/MAXNUM)

• Description: Find the largest of three numbers using conditional statements.

11. **LeetCode:** [Palindrome Number](https://leetcode.com/problems/palindrome-number/)

• Description: Check if a number is a palindrome using conditionals.

12. **GeeksforGeeks:** [Check Leap Year](https://www.geeksforgeeks.org/python-program-to-check-if-a-year-is-leap-year-or-not/)

• Description: Check if a given year is a leap year using conditionals.

13. **HackerRank:** [Simple If Else](https://www.hackerrank.com/challenges/simple-if-else/problem)

• Description: Implement a simple if-else structure in Python.

14. **LeetCode:** [Sum of Even and Odd Indices](https://leetcode.com/problems/sum-of-even-and-odd-indices/)

• Description: Compute the sum of elements at even and odd indices in a list using conditional checks.

15. **CodeChef:** [Check Divisibility by 5](https://www.codechef.com/problems/ODD)

• Description: Check if a number is divisible by 5 using conditionals.

16. **HackerRank:** [Grading System](https://www.hackerrank.com/challenges/grading/problem)

• Description: Apply a grading system to marks using conditional statements.

17. **LeetCode:** [Range Sum Query](https://leetcode.com/problems/range-sum-query/)

• Description: Implement a range sum query using conditionals.

18. **GeeksforGeeks:** [Vowel or Consonant](https://www.geeksforgeeks.org/python-program-check-character-vowel-consonant/)

• Description: Check if a character is a vowel or consonant using conditionals.

19. **LeetCode:** [Reverse Integer](https://leetcode.com/problems/reverse-integer/)

• Description: Reverse an integer using conditionals.

20. **GeeksforGeeks:** [Armstrong Number](https://www.geeksforgeeks.org/python-program-to-check-whether-a-number-is-armstrong-or-not/)

• Description: Check if a number is an Armstrong number using conditionals.

21. **HackerRank:** [Perfect Number](https://www.hackerrank.com/challenges/python-perfect-number/problem)

• Description: Check if a number is a perfect number using conditionals.

22. **LeetCode:** [Majority Element](https://leetcode.com/problems/majority-element/)

• Description: Find the majority element in a list using conditionals.

23. **GeeksforGeeks:** [Check Alphabetic or Numeric](https://www.geeksforgeeks.org/python-program-to-check-whether-a-character-is-alphabet-or-digit/)

• Description: Check if a character is an alphabet or a digit using conditionals.

24. **HackerRank:** [Greater Than or Equal to](https://www.hackerrank.com/challenges/greater-than-or-equal-to/problem)

• Description: Compare two numbers and check if one is greater than or equal to the other.

25. **CodeChef:** [Password Strength Check](https://www.codechef.com/problems/PASSCHECK)

• Description: Check the strength of a password based on conditions.

**5. Lists**

1. **LeetCode:** [Remove Duplicates from Sorted Array](https://leetcode.com/problems/remove-duplicates-from-sorted-array/)

• Description: Remove duplicates from a sorted list.

2. **GeeksforGeeks:** [Find the Largest Element in a List](https://www.geeksforgeeks.org/python-program-to-find-largest-element-in-a-list/)

• Description: Find the largest element in a list.

3. **HackerRank:** [List Operations](https://www.hackerrank.com/challenges/python-list/problem)

• Description: Perform various operations on lists such as insertion, deletion, and access.

4. **LeetCode:** [Two Sum](https://leetcode.com/problems/two-sum/)

• Description: Find two numbers in a list that add up to a given target.

5. **CodeChef:** [List Manipulation](https://www.codechef.com/problems/AMLIST)

• Description: Manipulate lists by performing various operations.

6. **GeeksforGeeks:** [Remove Odd Numbers](https://www.geeksforgeeks.org/python-remove-all-odd-numbers-from-list/)

• Description: Remove all odd numbers from a list.

7. **LeetCode:** [Find All Numbers Disappeared in an Array](https://leetcode.com/problems/find-all-numbers-disappeared-in-an-array/)

• Description: Find the numbers missing from an array.

8. **HackerRank:** [List Comprehensions](https://www.hackerrank.com/challenges/list-comprehensions/problem)

• Description: Practice list comprehensions for creating lists based on conditions.

9. **GeeksforGeeks:** [Find Minimum Element in List](https://www.geeksforgeeks.org/python-program-to-find-smallest-element-in-a-list/)

• Description: Find the minimum element in a list.

10. **LeetCode:** [Rotate Array](https://leetcode.com/problems/rotate-array/)

• Description: Rotate a list by a given number of steps.

11. **GeeksforGeeks:** [Find Frequency of Elements](https://www.geeksforgeeks.org/python-program-to-find-frequency-of-elements-in-list/)

• Description: Count the frequency of elements in a list.

12. **HackerRank:** [List Filter](https://www.hackerrank.com/challenges/filter-list/problem)

• Description: Filter a list based on certain conditions.

13. **LeetCode:** [Merge Intervals](https://leetcode.com/problems/merge-intervals/)

• Description: Merge overlapping intervals in a list.

14. **CodeChef:** [List Sorting](https://www.codechef.com/problems/LISTSORT)

• Description: Sort a list of elements in increasing or decreasing order.

15. **GeeksforGeeks:** [Sum of List Elements](https://www.geeksforgeeks.org/python-program-to-find-sum-of-elements-in-list/)

• Description: Find the sum of elements in a list.

16. **LeetCode:** [Insert Interval](https://leetcode.com/problems/insert-interval/)

• Description: Insert an interval into a sorted list of intervals.

17. **HackerRank:** [List Subtraction](https://www.hackerrank.com/challenges/list-subtraction/problem)

• Description: Subtract one list from another.

18. **GeeksforGeeks:** [Check List for Even Numbers](https://www.geeksforgeeks.org/python-check-whether-list-contains-even-numbers-or-not/)

• Description: Check if a list contains any even numbers.

19. **LeetCode:** [Find Peak Element](https://leetcode.com/problems/find-peak-element/)

• Description: Find a peak element in an array.

20. **HackerRank:** [Sum of Even Numbers in List](https://www.hackerrank.com/challenges/sum-even-numbers/problem)

• Description: Sum all the even numbers in a list.

21. **GeeksforGeeks:** [List Reversal](https://www.geeksforgeeks.org/python-program-to-reverse-a-list/)

• Description: Reverse the order of elements in a list.

22. **LeetCode:** [Find the Intersection of Two Arrays](https://leetcode.com/problems/intersection-of-two-arrays/)

• Description: Find the intersection of two lists.

23. **GeeksforGeeks:** [Largest Sum Contiguous Subarray](https://www.geeksforgeeks.org/python-program-to-find-largest-sum-contiguous-subarray/)

• Description: Find the largest sum of a contiguous subarray.

24. **HackerRank:** [Nested Lists](https://www.hackerrank.com/challenges/nested-list/problem)

• Description: Practice handling nested lists.

25. **CodeChef:** [Find Unique Elements in List](https://www.codechef.com/problems/UNIQUE)

• Description: Find unique elements in a list.

**6. Strings**

1. **LeetCode:** [Reverse String](https://leetcode.com/problems/reverse-string/)

• Description: Reverse a given string.

2. **HackerRank:** [String Split and Join](https://www.hackerrank.com/challenges/python-string-split-and-join/problem)

• Description: Split a string into words and join them back with a specific delimiter.

3. **GeeksforGeeks:** [String Palindrome](https://www.geeksforgeeks.org/python-program-to-check-whether-a-string-is-palindrome-or-not/)

• Description: Check if a string is a palindrome.

4. **LeetCode:** [Longest Common Prefix](https://leetcode.com/problems/longest-common-prefix/)

• Description: Find the longest common prefix among a list of strings.

5. **CodeChef:** [Palindrome String Check](https://www.codechef.com/problems/PALINDROME)

• Description: Check if a given string is a palindrome.

6. **GeeksforGeeks:** [Count Vowels in String](https://www.geeksforgeeks.org/python-program-to-count-vowels-in-string/)

• Description: Count the number of vowels in a string.

7. **HackerRank:** [String Introduction](https://www.hackerrank.com/challenges/python-strings-introduction/problem)

• Description: Basic string manipulation such as indexing, slicing, and concatenation.

8. **LeetCode:** [String Compression](https://leetcode.com/problems/string-compression/)

• Description: Implement string compression using counts of repeated characters.

9. **GeeksforGeeks:** [Remove Punctuation from String](https://www.geeksforgeeks.org/python-program-to-remove-punctuation-from-string/)

• Description: Remove all punctuation characters from a string.

10. **LeetCode:** [Valid Anagram](https://leetcode.com/problems/valid-anagram/)

• Description: Check if two strings are anagrams.

11. **HackerRank:** [String Formatting](https://www.hackerrank.com/challenges/python-string-formatting/problem)

• Description: Use string formatting techniques to display numbers and strings.

12. **GeeksforGeeks:** [Count Occurrences of Substring](https://www.geeksforgeeks.org/python-count-occurrences-of-substring-in-string/)

• Description: Count the occurrences of a substring within a string.

13. **LeetCode:** [Find All Anagrams in a String](https://leetcode.com/problems/find-all-anagrams-in-a-string/)

• Description: Find all anagrams of a string within another string.

14. **GeeksforGeeks:** [Replace Spaces with Hyphen](https://www.geeksforgeeks.org/python-replace-all-spaces-with-hyphen/)

• Description: Replace all spaces in a string with hyphens.

15. **HackerRank:** [String Mutation](https://www.hackerrank.com/challenges/python-mutation/problem)

• Description: Modify a string based on some conditions.

16. **LeetCode:** [Check if String Contains All Vowels](https://leetcode.com/problems/check-if-a-string-contains-all-vowels/)

• Description: Check if a string contains all the vowels.

17. **CodeChef:** [Reverse Each Word in a Sentence](https://www.codechef.com/problems/REVERSE)

• Description: Reverse each word in a sentence, but maintain the word order.

18. **GeeksforGeeks:** [Remove Duplicates from String](https://www.geeksforgeeks.org/python-remove-duplicates-from-string/)

• Description: Remove duplicate characters from a string.

19. **LeetCode:** [Count and Say](https://leetcode.com/problems/count-and-say/)

• Description: Generate the n-th term of the “Count and Say” sequence.

20. **HackerRank:** [String Concatenation](https://www.hackerrank.com/challenges/python-string-concatenation/problem)

• Description: Concatenate two strings with some specific rules.

21. **GeeksforGeeks:** [Check if String is Subset of Another](https://www.geeksforgeeks.org/python-check-if-string-is-subset-of-another/)

• Description: Check if a string is a subset of another string.

22. **LeetCode:** [Implement strStr](https://leetcode.com/problems/implement-strstr/)

• Description: Implement the strStr() function that finds the first occurrence of a substring in a string.

23. **HackerRank:** [String Validation](https://www.hackerrank.com/challenges/python-string-validation/problem)

• Description: Check if a string contains only alphanumeric characters and if it is in uppercase.

24. **GeeksforGeeks:** [Count Substrings with Given Prefix](https://www.geeksforgeeks.org/python-count-substrings-with-given-prefix/)

• Description: Count how many substrings of a string start with a given prefix.

25. **CodeChef:** [String Reversal](https://www.codechef.com/problems/REVERSESTR)

• Description: Reverse a given string.

**7. Functions**

1. **LeetCode:** [Climbing Stairs](https://leetcode.com/problems/climbing-stairs/)

• Description: Calculate the number of ways to climb to the top of a staircase using dynamic programming.

2. **HackerRank:** [Functions in Python](https://www.hackerrank.com/challenges/python-functions/problem)

• Description: Write functions to handle simple mathematical problems.

3. **GeeksforGeeks:** [Sum of Natural Numbers](https://www.geeksforgeeks.org/python-program-to-find-sum-of-natural-numbers/)

• Description: Write a function to find the sum of natural numbers up to n.

4. **LeetCode:** [Maximum Subarray](https://leetcode.com/problems/maximum-subarray/)

• Description: Find the contiguous subarray with the largest sum.

5. **GeeksforGeeks:** [Factorial of a Number](https://www.geeksforgeeks.org/python-program-to-find-factorial-of-a-number/)

• Description: Write a function to find the factorial of a number.

6. **HackerRank:** [Recursive Function](https://www.hackerrank.com/challenges/recursive-digit-sum/problem)

• Description: Implement a recursive function to compute the recursive digit sum.

7. **LeetCode:** [Power of Two](https://leetcode.com/problems/power-of-two/)

• Description: Write a function to check if a number is a power of two.

8. **CodeChef:** [Sum of Digits](https://www.codechef.com/problems/ENGLISH)

• Description: Write a function to find the sum of digits of a number.

9. **GeeksforGeeks:** [Check Prime Number](https://www.geeksforgeeks.org/python-program-to-check-whether-a-number-is-prime-or-not/)

• Description: Write a function to check if a number is prime.

10. **HackerRank:** [Exponentiation](https://www.hackerrank.com/challenges/python-exponentiation/problem)

• Description: Write a function that performs exponentiation.

11. **LeetCode:** [Reverse Integer](https://leetcode.com/problems/reverse-integer/)

• Description: Implement a function that reverses an integer.

12. **CodeChef:** [Find GCD](https://www.codechef.com/problems/PGCD)

• Description: Write a function to compute the greatest common divisor of two numbers.

13. **GeeksforGeeks:** [Find LCM](https://www.geeksforgeeks.org/python-program-to-find-lcm-of-two-numbers/)

• Description: Write a function to compute the least common multiple of two numbers.

14. **HackerRank:** [Python Print Function](https://www.hackerrank.com/challenges/python-print/problem)

• Description: Write a function that prints the numbers from 1 to n.

15. **LeetCode:** [Sum of Two Integers](https://leetcode.com/problems/sum-of-two-integers/)

• Description: Write a function to calculate the sum of two integers.

16. **GeeksforGeeks:** [Find Fibonacci Number](https://www.geeksforgeeks.org/python-program-to-find-fibonacci-number-using-recursion/)

• Description: Write a function to find the nth Fibonacci number.

17. **HackerRank:** [Calculate Factorial](https://www.hackerrank.com/challenges/python-factorial/problem)

• Description: Implement a function to calculate the factorial of a number.

18. **LeetCode:** [Add Digits](https://leetcode.com/problems/add-digits/)

• Description: Write a function to add digits of a number until a single digit is achieved.

19. **GeeksforGeeks:** [Reverse List Using Recursion](https://www.geeksforgeeks.org/python-reverse-list-using-recursion/)

• Description: Reverse a list using recursion.

20. **LeetCode:** [Missing Number](https://leetcode.com/problems/missing-number/)

• Description: Find the missing number in an array using a function.

21. **HackerRank:** [Python Closures](https://www.hackerrank.com/challenges/python-closures/problem)

• Description: Implement closures in Python.

22. **GeeksforGeeks:** [Count Vowels and Consonants](https://www.geeksforgeeks.org/python-count-vowels-and-consonants-in-string/)

• Description: Count vowels and consonants in a string using a function.

23. **LeetCode:** [Length of Last Word](https://leetcode.com/problems/length-of-last-word/)

• Description: Write a function to find the length of the last word in a string.

24. **GeeksforGeeks:** [Find Common Elements](https://www.geeksforgeeks.org/python-find-common-elements-in-three-lists/)

• Description: Write a function to find the common elements between three lists.

25. **LeetCode:** [Valid Parentheses](https://leetcode.com/problems/valid-parentheses/)

• Description: Write a function to check if parentheses are balanced in a string.

**8. Dictionaries**

1. **LeetCode:** [Two Sum](https://leetcode.com/problems/two-sum/)

• Description: Use a dictionary to solve the two-sum problem efficiently.

2. **GeeksforGeeks:** [Count Occurrences of Element in List](https://www.geeksforgeeks.org/python-count-occurrences-of-an-element-in-a-list/)

• Description: Count the occurrences of an element in a list using a dictionary.

3. **HackerRank:** [Dictionary Operations](https://www.hackerrank.com/challenges/collections-ordereddict/problem)

• Description: Perform various operations on dictionaries, such as adding, updating, and deleting entries.

4. **LeetCode:** [Group Anagrams](https://leetcode.com/problems/group-anagrams/)

• Description: Use a dictionary to group anagrams from a list of strings.

5. **GeeksforGeeks:** [Dictionary to List Conversion](https://www.geeksforgeeks.org/python-program-convert-dictionary-into-list/)

• Description: Convert a dictionary into a list.

6. **HackerRank:** [Merge Two Dictionaries](https://www.hackerrank.com/challenges/merge-dictionaries/problem)

• Description: Merge two dictionaries in Python.

7. **LeetCode:** [Intersection of Two Arrays II](https://leetcode.com/problems/intersection-of-two-arrays-ii/)

• Description: Find the intersection of two arrays using dictionaries.

8. **GeeksforGeeks:** [Nested Dictionary Access](https://www.geeksforgeeks.org/nested-dictionary-in-python/)

• Description: Access elements in a nested dictionary.

9. **HackerRank:** [Student Marks Dictionary](https://www.hackerrank.com/challenges/python-dictionary-task/problem)

• Description: Use a dictionary to store and process student marks.

10. **LeetCode:** [Roman to Integer](https://leetcode.com/problems/roman-to-integer/)

• Description: Convert a Roman numeral string to an integer using a dictionary for mapping.

11. **GeeksforGeeks:** [Dictionary Key Value Swap](https://www.geeksforgeeks.org/python-swap-key-values-in-dictionary/)

• Description: Swap the keys and values in a dictionary.

12. **LeetCode:** [Sort Dictionary by Values](https://leetcode.com/problems/sort-items-by-groups-according-to-strings/)

• Description: Sort a dictionary by its values.

13. **HackerRank:** [Default Dictionaries](https://www.hackerrank.com/challenges/collections-defaultdict-tutorial/problem)

• Description: Work with Python’s defaultdict to handle missing keys.

14. **GeeksforGeeks:** [Check If Key Exists in Dictionary](https://www.geeksforgeeks.org/python-check-if-key-exists-in-dictionary/)

• Description: Check if a key exists in a dictionary.

15. **LeetCode:** [Map Sum Pairs](https://leetcode.com/problems/map-sum-pairs/)

• Description: Implement a key-value map with sum operations.

16. **CodeChef:** [Frequency of Elements in Dictionary](https://www.codechef.com/problems/FREQ)

• Description: Find the frequency of elements using a dictionary.

17. **GeeksforGeeks:** [Find the Frequency of Words](https://www.geeksforgeeks.org/python-find-frequency-of-words-in-list/)

• Description: Count the frequency of words in a list using a dictionary.

18. **LeetCode:** [Top K Frequent Elements](https://leetcode.com/problems/top-k-frequent-elements/)

• Description: Find the top k frequent elements in a list using a dictionary.

19. **HackerRank:** [Counting Characters in a String](https://www.hackerrank.com/challenges/counting-characters-in-a-string/problem)

• Description: Count occurrences of characters in a string using a dictionary.

20. **GeeksforGeeks:** [Sort a Dictionary by Keys](https://www.geeksforgeeks.org/sort-python-dictionary-by-key/)

• Description: Sort a dictionary by its keys.

21. **LeetCode:** [Isomorphic Strings](https://leetcode.com/problems/isomorphic-strings/)

• Description: Check if two strings are isomorphic using dictionaries.

22. **HackerRank:** [Dictionary Merging](https://www.hackerrank.com/challenges/collections-counter/problem)

• Description: Merge two dictionaries using Python’s Counter.

23. **GeeksforGeeks:** [Convert Dictionary to JSON](https://www.geeksforgeeks.org/python-convert-dictionary-to-json/)

• Description: Convert a Python dictionary into a JSON object.

24. **LeetCode:** [Find All Duplicates in an Array](https://leetcode.com/problems/find-all-duplicates-in-an-array/)

• Description: Use a dictionary to find duplicates in an array.

25. **CodeChef:** [Phonebook Dictionary](https://www.codechef.com/problems/PHONENUM)

• Description: Implement a phonebook system using a dictionary.

With these additional questions, we’ve now covered all concepts within **Milestone 1: Python Basics**. If you need further assistance or additional