

1. String Manipulation

1. Reverse a string.
2. Check if a string is a palindrome.
3. Count occurrences of each character in a string.
4. Find the first non-repeating character in a string.
5. Compress a string (e.g., "aaaabbbcc" → "a4b3c2").
6. Remove duplicate characters from a string.
7. Check if two strings are anagrams.
8. Find the longest common prefix in an array of strings.
9. Find all permutations of a string.
10. Check if a string contains only unique characters.

2. Number Problems

1. Check if a number is prime.
2. Find all prime numbers in a range.
3. Check if a number is an Armstrong number.
4. Find the factorial of a number.
5. Find the greatest common divisor (GCD) of two numbers.
6. Find the least common multiple (LCM) of two numbers.
7. Generate Fibonacci numbers up to n.
8. Check if a number is a perfect number.
9. Convert a decimal number to binary, octal, and hexadecimal.
10. Find the sum of digits of a number.

3. Array and List Operations

1. Find the largest and smallest numbers in an array.
2. Reverse an array.
3. Find duplicates in an array.
4. Remove duplicates from a list.
5. Rotate an array by k positions.
6. Find the second largest number in an array.

7. Find the subarray with the maximum sum.
8. Merge two sorted arrays.
9. Find the intersection of two arrays.
10. Check if an array is a subset of another array.

4. Matrix Problems

1. Transpose a matrix.
2. Find the sum of diagonal elements in a matrix.
3. Rotate a matrix by 90 degrees.
4. Add two matrices.
5. Multiply two matrices.
6. Check if a matrix is symmetric.
7. Find the largest element in a matrix.
8. Perform spiral traversal of a matrix.
9. Count the number of zeros in a matrix.
10. Check if a matrix is an identity matrix.

5. Searching and Sorting

1. Implement binary search.
2. Implement bubble sort.
3. Implement selection sort.
4. Implement insertion sort.
5. Implement merge sort.
6. Implement quicksort.
7. Find the kth smallest element in an array.
8. Find the peak element in an array.
9. Check if an array is sorted.
10. Perform a linear search.

6. Recursion Problems

1. Solve the Tower of Hanoi.

2. Calculate the nth Fibonacci number.
3. Find the sum of digits of a number using recursion.
4. Generate all subsets of a set using recursion.
5. Find the power of a number using recursion.
6. Solve a maze problem using recursion.
7. Print all permutations of a string using recursion.
8. Reverse a string using recursion.
9. Find the GCD of two numbers using recursion.
10. Flatten a nested list using recursion.

8. File Handling

1. Read a file and count the number of lines, words, and characters.
2. Write a list of strings to a file.
3. Append content to an existing file.
4. Count the frequency of words in a file.
5. Find the largest word in a file.
6. Merge the content of two files into a third file.
7. Remove blank lines from a file.
8. Copy content from one file to another.
9. Find and replace a word in a file.
10. Reverse the content of a file line by line.

9. Miscellaneous Problems

1. Implement a stack using a list.
2. Implement a queue using a list.
3. Find the missing number in an array.
4. Find the majority element in an array.
5. Check if a number is a power of 2.
6. Validate an IP address.
7. Generate all combinations of balanced parentheses.
8. Solve the N-Queens problem.

9. Implement a simple calculator.
10. Simulate a dice roll.

10. Object-Oriented Programming

1. Create a class to represent a bank account with methods to deposit and withdraw money.
2. Implement a class for a stack with push, pop, and display methods.
3. Create a class for a linked list with methods for insertion and deletion.
4. Implement method overloading and overriding.
5. Create a class to represent a 2D point and calculate the distance between two points.
6. Implement a class for a shopping cart.
7. Create a singleton class in Python.
8. Implement a class for polynomial arithmetic.
9. Create a class to handle complex numbers.
10. Design a class for a library management system.