C Programing Gemini Problems

Topic 1: Advanced Number Theory & Series

Easy (Fundamental Logic)

1. Write a program to calculate the **sum of digits** of a number until it becomes a single-digit number (e.g., 98 -> 17 -> 8).
2. Find the **sum of all even factors** of a user-entered number.
3. Check if a number is a **Strong Number** (Sum of factorial of digits = original number).
4. Print all numbers between 1 and N that are divisible by both 3 and 5 but not by 10.
5. Count the number of **trailing zeros** in the factorial of a given number.
6. Write a program to check if a given number is a **power of 2** using loops.Mid (Complex Iteration)
7. Convert a Decimal number to Binary without using arrays (using arithmetic operations).
8. Convert a Binary number to Decimal.
9. Print Floyd's Triangle (1, 2 3, 4 5 6...) up to N rows.
10. Check if a number is an Automorphic Number (ends with the same digits as the number itself, e.g., $5^2 = 25$, $6^2 = 36$).
11. Check if a number is a Harshad Number (divisible by the sum of its digits).
12. Find the LCM of three numbers entered by the user.

Hard (Algorithm Heavy)

13. Write a program to print the Prime Factors of a number (e.g., Input 12 -> Output 2, 2, 3).
14. Convert a decimal number to Roman Numerals.
15. Generate a Spiral Number Pattern for size N (filling a matrix logic using loops).

Topic 2: 1D Arrays

Easy (Basic Operations)

16. Initialize an array of 10 integers and print the sum and average of all elements.
17. Find the maximum and minimum element in an array in a single pass.
18. Search for a specific number in an array and print its index (Linear Search).
19. Count the frequency of a specific number in an array.
20. Create a second array that contains elements of the first array in reverse order.

21. Check if an array is sorted in ascending order (return 1 for yes, 0 for no).

22. Copy all positive elements of one array into a new array.

23. Replace all odd numbers in an array with 0 and print the modified array.

Mid (Array Manipulation)

24. Write a program to remove duplicate elements from an array.

25. Rotate an array to the left by K positions.

26. Merge two sorted arrays into a third sorted array.

27. Find the second largest number in an array without sorting.

28. Implement Bubble Sort to sort array elements in ascending order.

29. Find the missing number in a sequence of integers from 1 to N.

30. Move all zeros to the end of the array while maintaining the order of non-zero elements.

31. Find the intersection of two arrays (common elements).

Hard (Optimization & Sliding Window)

32. Implement Kadane's Algorithm to find the contiguous subarray with the largest sum.

33. Find the length of the longest consecutive sequence of integers in an unsorted array.

34. Find the majority element (an element that appears more than N/2 times).

35. Rearrange an array such that positive and negative numbers are placed alternatively.

Topic 3: Multidimensional Arrays (Matrices)

Easy (Grid Basics)

36. Write a program to add two 3x3 matrices.

37. Print only the main diagonal elements of a square matrix.

38. Calculate the sum of each row and each column separately.

39. Find the Transpose of a given matrix.

40. Check if a matrix is an Identity Matrix.

41. Accept a 2D array and count how many negative numbers are present.

Mid (Matrix Algebra)

42. Write a program to multiply two matrices (check dimensions first).

43. Rotate a matrix 90 degrees clockwise.

44. Sort the elements of each row of the matrix in ascending order.

45. Find the Saddle Point of a matrix (smallest in row, largest in column).

46. Check if a given matrix is Symmetric (Transpose == Original).

47. Calculate the sum of the Upper Triangular and Lower Triangular elements.

Hard (Complex Traversal)

48. Print a matrix in Spiral Order.

49. Calculate the Determinant of a 3x3 matrix.

50. Generate a Magic Square of odd order N (sum of every row, col, diag is same).

Topic 4: String Handling

Easy (Char Processing)

51. Calculate the length of a string without using strlen().

52. Copy one string to another without using strcpy().

53. Concatenate two strings manually.

54. Count the number of vowels, consonants, digits, and spaces in a string.

55. Toggle the case of every character in a string (Upper -> Lower, Lower -> Upper).

56. Check if a string contains only alphanumeric characters.

Mid (Pattern Matching)

57. Check if a string is a Palindrome (case sensitive).

58. Check if two strings are Anagrams of each other.

59. Count the total number of words in a sentence.

60. Remove all occurrences of a specific character from a string.

61. Compress a string using counts (e.g., "aaabbc" -> "a3b2c1").

62. Find the first occurrence of a substring within a main string (substring search).

Hard (Parsing & Logic)

63. Find the length of the longest substring without repeating characters.

64. Reverse the order of words in a given sentence (e.g., "Hello World" -> "World Hello").

65. Implement a function to convert a String to Float (similar to atof).

Topic 5: Functions & Recursion

Easy (Modular Basics)

66. Write a function to calculate the area and circumference of a circle given radius.

67. Create a function checkEven() that returns 1 if even, 0 if odd.

68. Write a function to swap two numbers (Call by Value vs Call by Reference).

69. Calculate Factorial of a number using recursion.

70. Calculate the sum of natural numbers up to N using recursion.

71. Write a function to calculate x^y (power) using recursion.

Mid (Recursive Logic)

72. Find the GCD (HCF) of two numbers using recursion.

73. Print the Nth Fibonacci term using a recursive function.

74. Implement the Ackermann Function (famous for deep recursion).

75. Write a recursive function to calculate the sum of digits of a number.

76. Convert Decimal to Binary using recursion.

77. Solve the Tower of Hanoi problem for N disks.

Hard (Backtracking)

78. Print all permutations of a given string (Backtracking).

79. Write a recursive function to reverse a string in place.

80. Validate a Sudoku board (check if a 9x9 configuration is valid).

Topic 6: Pointers & Memory Management

Easy (Pointer Syntax)

81. Write a program to print the address and value of a variable using pointers.

82. Swap two numbers using pointers.

83. Add two numbers using pointers.

84. Print all elements of an array using pointer arithmetic.

Mid (Pointer Logic)

85. Calculate the length of a string using a pointer.

86. Copy a string from source to destination using pointers.

87. Reverse an array using pointers.

88. Find the largest element in an array using Dynamic Memory Allocation (malloc).

Hard (Advanced Pointers)

89. Implement a calculator using an array of function pointers.

90. Simulate the logic of realloc(): create a dynamic array, resize it, and preserve data.

Topic 7: Structures & File Handling

Easy (Struct Basics)

91. Define a structure Student (Name, Roll, Marks) and read/print data for one student.

92. Write a program to add two Complex numbers using structures.

93. Calculate the difference between two Time periods (Hours, Minutes, Seconds) using structures.

94. Write a program to create a file and write a string entered by the user into it.

Mid (Data Management)

95. Create an array of structures for 5 employees and sort them by Salary.

96. Write a program to copy the contents of one file to another.

97. Count the number of lines and words in a text file.

98. Append data to an existing file without overwriting it.

Hard (System Simulation)

99. Create a mini Bank Management System (Account Number, Name, Balance) with Deposit/Withdraw functions using File I/O.

100. Read a CSV file containing student data and calculate the class average.