C programming practice

Sure, here are 100 C programming questions covering various topics from basic to advanced:

### Basic Concepts

1. Print "Hello, World!".

2. Define and print variables of different types.

3. Perform arithmetic operations (+, -, \*, /) on two numbers.

4. Check if a number is even or odd.

5. Calculate the factorial of a number.

6. Swap two numbers without using a third variable.

7. Find the largest among three numbers.

8. Check if a character is a vowel or consonant.

9. Calculate the sum of digits of a number.

10. Check if a number is prime or not.

11. Generate Fibonacci series up to n terms.

12. Reverse a number and check if it is a palindrome.

13. Calculate the power of a number using recursion.

14. Calculate the sum of a series: 1 + 1/2 + 1/3 + ... + 1/n.

15. Write a program to find the square root of a number.

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

17. Convert temperature from Celsius to Fahrenheit and vice versa.

18. Calculate the area of a circle, rectangle, and triangle.

19. Check if a number is Armstrong or not.

20. Convert decimal to binary, octal, and hexadecimal.

### Control Statements

21. Write a program to implement a simple calculator (+, -, \*, /).

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

23. Find the largest among n numbers using arrays.

24. Find the factorial of a number using a while loop.

25. Generate a multiplication table of a number.

26. Calculate the sum of natural numbers up to n.

27. Display numbers in reverse order from n to 1.

28. Check if a number is perfect or not.

29. Print Floyd's Triangle.

30. Calculate the GCD (Greatest Common Divisor) of two numbers.

31. Print Pascal's Triangle.

32. Check if a number is a strong number or not.

33. Convert binary to decimal and vice versa.

34. Find the roots of a quadratic equation.

35. Print all prime numbers between two given numbers.

36. Check if a string is palindrome or not.

37. Sort elements of an array in ascending and descending order.

38. Calculate the sum of all prime numbers up to n.

39. Find the LCM (Least Common Multiple) of two numbers.

40. Print the ASCII value of a character.

### Functions

41. Write a function to find the factorial of a number.

42. Check if a number is even or odd using a function.

43. Find the largest among three numbers using a function.

44. Calculate the power of a number using a function.

45. Check if a number is prime or not using a function.

46. Generate Fibonacci series up to n terms using a function.

47. Reverse a number using a function.

48. Calculate the sum of digits of a number using a function.

49. Calculate the sum of two matrices.

50. Find the transpose of a matrix.

### Arrays and Strings

51. Find the sum and average of elements in an array.

52. Find the largest and smallest element in an array.

53. Search for an element in an array.

54. Insert an element into an array at a specified position.

55. Delete an element from an array at a specified position.

56. Copy elements from one array to another.

57. Concatenate two strings.

58. Compare two strings.

59. Convert lowercase to uppercase and vice versa.

60. Check if a string is an anagram or not.

### Pointers

61. Swap two numbers using pointers.

62. Find the factorial of a number using pointers.

63. Reverse a string using pointers.

64. Allocate memory dynamically for an array.

65. Pass arrays to functions using pointers.

### Structures and File Handling

66. Create a structure to store student details.

67. Calculate the average marks of students using structures.

68. Read and write to a file.

69. Copy contents of one file to another.

70. Append data to a file.

### Advanced Topics

71. Implement bubble sort and selection sort.

72. Implement binary search and linear search.

73. Implement quicksort and mergesort.

74. Implement a stack using arrays.

75. Implement a queue using arrays.

### Dynamic Memory Allocation

76. Allocate memory for an integer using malloc.

77. Allocate memory for a character array using calloc.

78. Allocate memory for a structure using malloc.

### Recursion

79. Write a recursive function to calculate factorial.

80. Write a recursive function to calculate Fibonacci series.

### Data Structures

81. Implement a linked list (insertion, deletion, display).

82. Implement a stack (push, pop, display).

83. Implement a queue (enqueue, dequeue, display).

### File Handling

84. Read data from a CSV file and display it.

85. Write data to a CSV file.

### Preprocessor Directives

86. Use #define to define a constant value.

87. Use #include to include header files.

88. Use conditional compilation (#ifdef, #ifndef, #endif).

### Bitwise Operations

89. Perform bitwise AND, OR, XOR operations.

90. Perform bitwise left shift and right shift operations.

### Command Line Arguments

91. Write a program to accept command line arguments.

92. Perform operations based on command line arguments.

### Error Handling

93. Handle runtime errors using try-catch blocks.

94. Handle file I/O errors.

### Multithreading

95. Implement a multithreaded program.

96. Synchronize threads using mutex and semaphore.

### Networking

97. Create a simple client-server program.

98. Implement socket programming.

### Graphics and GUI

99. Create a basic graphics program using a library.

100. Design a simple GUI using a toolkit.

These questions cover a wide range of topics in C programming and will help you strengthen your skills from basic concepts to advanced topics.