# ### Basic Level (1-30)

1. Hello World Program

2. Variables and Data Types

3. Simple Calculator (Addition, Subtraction, Multiplication, Division)

4. Even or Odd Number Check

5. Factorial of a Number

6. Fibonacci Series

7. Reverse a String

8. Palindrome Check

9. Sum of Elements in an Array

10. Largest Element in an Array

11. Average of Elements in an Array

12. **Count Vowels in a String**

13. Prime Number Check

14. Matrix Addition

15. Bubble Sort

16. Selection Sort

17. Insertion Sort

18. Linear Search in an Array

19. Binary Search in an Array (sorted array)

20. Find Duplicate Elements in an Array

21. Merge Two Arrays

22. Compute Power of a Number

23. Calculate GCD and LCM of Two Numbers

24. Calculate Area and Circumference of a Circle

25. Generate Random Numbers within a Range

26. Convert Decimal to Binary

27. Convert Binary to Decimal

28. Print Pattern (e.g., pyramid pattern)

29. Find Factorial of a Large Number using BigInteger

30. Check Leap Year

## ### Intermediate Level (31-70)

31. Armstrong Number Check

32. Reverse a Number

33. Check if String is Anagram

34. Count Occurrences of Each Character in a String

35. Calculate Sum of Natural Numbers using Recursion

36. Print Fibonacci Series using Recursion

37. Print Pascal's Triangle

38. Reverse Words in a Sentence

39. Remove Duplicate Elements from an Array

40. Check if Array is Sorted (ascending or descending)

41. Find Median of Two Sorted Arrays

42. Find Missing Number in an Array of 1 to N

43. Rotate Array Elements to the Right

44. Check if Array contains a Subarray with Sum Zero

45. Check if Linked List is Palindrome

46. Implement Stack using Array

47. Implement Queue using Linked List

48. Implement Binary Search Tree (Insert, Search, Delete)

49. Implement Singly Linked List (Insert, Delete, Search)

50. Implement Doubly Linked List (Insert, Delete, Search)

51. Check if Linked List has Cycle

52. Reverse a Linked List

53. Merge Two Sorted Linked Lists

54. Calculate Factorial of a Number using Recursion

55. Calculate Power of a Number using Recursion

56. Implement Bubble Sort Algorithm for Linked List

57. Implement Quick Sort Algorithm for Arrays

58. Implement Depth First Search (DFS) Algorithm

59. Implement Breadth First Search (BFS) Algorithm

60. Implement Dijkstra's Algorithm for Shortest Path

61. Implement Bellman-Ford Algorithm for Shortest Path

62. Implement Kruskal's Algorithm for Minimum Spanning Tree

63. Implement Prim's Algorithm for Minimum Spanning Tree

64. Check if Graph is Bipartite using BFS/DFS

65. Check if Graph is Connected using DFS/BFS

66. Find Strongly Connected Components in a Graph

67. Calculate Maximum Flow in a Network using Ford-Fulkerson Algorithm

68. Implement Floyd-Warshall Algorithm for All Pairs Shortest Path

69. Implement KMP (Knuth-Morris-Pratt) Algorithm for Pattern Searching

70. Implement AVL Tree (Insert, Delete, Search)

## ### Advanced Level (71-100)

71. Implement Trie (Prefix Tree)

72. Implement Red-Black Tree (Insert, Delete, Search)

73. Implement A\* Algorithm for Pathfinding

74. Implement Min-Heap and Heap Sort

75. Implement Topological Sorting of a Directed Acyclic Graph (DAG)

76. Implement B+ Tree (Insert, Delete, Search)

77. Implement Skip List (Insert, Delete, Search)

78. Implement Johnson's Algorithm for All Pairs Shortest Path

79. Implement Edmonds-Karp Algorithm for Maximum Flow

80. Implement Dinic's Algorithm for Maximum Flow

81. Implement Hungarian Algorithm for Bipartite Matching

82. Implement Karger's Algorithm for Minimum Cut

83. Implement Ford-Fulkerson Algorithm for Maximum Bipartite Matching

84. Implement Aho-Corasick Algorithm for String Searching

85. Implement Rabin-Karp Algorithm for String Searching

86. Implement Boyer-Moore Algorithm for String Searching

87. Implement Ford-Fulkerson Algorithm for Maximum Flow with Capacity Scaling

88. Implement Karatsuba Algorithm for Fast Multiplication of Large Numbers

89. Implement Pollard's Rho Algorithm for Integer Factorization

90. Implement Miller-Rabin Primality Test for Large Numbers

91. Implement RSA Encryption and Decryption

92. Implement SHA-256 Hashing Algorithm

93. Implement AVL Tree with Threaded Nodes

94. Implement B-Tree (Insert, Delete, Search)

95. Implement B\* Tree (Insert, Delete, Search)

96. Implement Treap (Binary Search Tree with Random Priority)

97. Implement Ternary Search Tree (Insert, Delete, Search)

98. Implement Suffix Array Construction Algorithm

99. Implement Burrows-Wheeler Transform

100. Implement LZ77 Compression Algorithm

These questions cover a wide range of Java programming concepts and algorithms. You can choose questions based on your current level of expertise and gradually move towards more advanced topics as you gain proficiency. Happy coding!