

## **JAVA INTERVIEW QUESTION BANK**

1. Write a program to **reverse a number**.
2. Write a program to find the **Factorial of a number**.
3. Write a program to **check whether a number is palindrome**.
4. Write a program to **find factorial of a number** using recursion.
5. Write a program to **generate Fibonacci series**.
6. Write a program to **check whether a number is prime**.
7. Write a program to **print all prime numbers between 1 and 100**.
8. Write a Java program to **reverse a string**.
9. Write a program to **check whether a string is palindrome**.
10. Write a program to **count vowels and consonants in a string**.
11. Write a program to **find duplicate characters in a string**.
12. Write a program to **count the frequency of characters in a string**.
13. Write a program to **remove duplicate characters from a string**.
14. Write a program to **check if two strings are anagrams**.
15. Write a program to **find the first non-repeated character in a string**.
16. Write a program to **reverse words in a sentence**.
17. Write a program to **count words in a string**.
18. Write a program to **find the largest element in an array**.
19. Write a program to **find the second largest element in an array**.
20. Write a program to **sort an array** (without using built-in methods).
21. Write a program to **remove duplicate elements from an array**.
22. Write a program to **find missing number in an array**.
23. Write a program to **find sum of elements in an array**.
24. Write a program to **reverse an array**.
25. Write a program to **check whether an array is sorted**.
26. Write a program to **find common elements between two arrays**.

- 27.** Write a program to **rotate an array left/right**.
- 28.** Write a program to **demonstrate method overloading**.
- 29.** Write a program to **demonstrate method overriding**.
- 30.** Write a program to **implement an interface**.
- 31.** Write a program using **abstract class and method**.
- 32.** Write a program to **create a singleton class**.
- 33.** Find the **third highest element** in an array.
- 34.** Find the **third smallest element** in an array.
- 35.** Find the **kth highest element** in an array.
- 36.** Find the **kth smallest element** in an array.
- 37.** Find the **second highest element without sorting**.
- 38.** Find the **second smallest element without sorting**.
- 39.** Find the **top 3 maximum elements** in an array.
- 40.** Find the **least frequent element** in an array.
- 41.** Find the **second most frequent element** in an array.
- 42.** Find the **third most frequent element** in an array.
- 43.** Find the **highest occurring character** in a string.
- 44.** Find the **second highest occurring character** in a string.
- 45.** Find the **third highest occurring character** in a string.
- 46.** Find the **kth most frequent character** in a string.
- 47.** Find the **longest word** in a sentence.
- 48.** Find the **second longest word** in a sentence.
- 49.** Find the **third longest word** in a sentence.
- 50.** Find the **kth longest word** in a sentence.
- 51.** Find the **largest word lexicographically** in a string array.
- 52.** Find the **smallest word lexicographically** in a string array.
- 53.** Find the **top 3 repeated words** in a paragraph.
- 54.** Find the **first and second maximum sum pair** in an array.

- 55.** Find the **maximum and second maximum difference** between elements.
- 56.** Find the **third largest number without using collections**.
- 57.** Find the **third largest number using TreeSet**.
- 58.** Find the **first repeating element** in an array.
- 59.** Find the **longest consecutive sequence** in an array.
- 60.** Find the **pair with maximum sum** in an array.
- 61.** Find the **pair whose sum equals a given number**.
- 62.** Find the **duplicate number in an array of size n+1**.
- 63.** Find the **missing and repeating number** in an array.
- 64.** Move **all zeros to the end of the array**.
65. Sort an array of **0s, 1s, and 2s** (Dutch National Flag problem).
- 66.** Find the **longest substring without repeating characters**.
67. Reverse words in a string **without using split()**.
- 68.** Find the **most repeated word** in a paragraph.
69. Check if a string contains **only digits**.
- 70.** Validate **balanced parentheses** using stack.
- 71.** Remove **adjacent duplicate characters** from a string.
- 72.** Print numbers from **1 to 100 without using loops**.