11. Stream API 200 Questions [Basic to Moderate]

📝 **Note for Developers**

While solving these Stream API questions, you'll come across many methods, collectors, and functional interfaces that may be new to you.

👉 **Your task:**
Whenever you encounter a new method, **take the time to look it up, understand how it works**, and **add it to your personal notes** — along with a short example or use case.

This habit will not only deepen your understanding of the Stream API but also help you **build a focused revision sheet** aligned with your **interview goals**.

⭐ List - 50 Questions

1. Print all elements of a list using Stream API.
2. Given a list of integers, print only even numbers using streams.
3. Given a list of strings, convert all to uppercase using streams.
4. Given a list of strings, filter those starting with the letter "A".
5. Count how many numbers in a list are greater than 10.
6. Remove all null or empty strings from a list using streams.
7. Sort a list of integers in ascending order using streams.
8. Sort a list of strings in descending order using streams.
9. Find the first element in a list using Stream API.
10. Find the maximum value in a list of integers using streams.
11. Find the minimum value in a list of integers using streams.
12. Count how many elements are in a list using Stream API.
13. Given a list of integers, double each value using `map()` .
14. Filter and print only unique elements from a list using `distinct()` .
15. Concatenate all strings in a list into a single comma-separated string.
16. Convert a list of integers into a list of their squares.
17. Check if any number in a list is divisible by 7 using streams.
18. Check if all strings in a list have length greater than 3.

19. Get the sum of all integers in a list using `reduce()`.
20. Collect only odd numbers from a list into a new list using streams.
21. Given a list of words, create a list of their lengths using `map()`.
22. Given a list of sentences, extract all unique words using `flatMap()`.
23. Find the average of all integers in a list using streams.
24. Given a list of numbers, find the second highest number using streams.
25. Find the second lowest number in a list using streams.
26. Convert a list of strings into a `Map<String, Integer>` where key = string, value = its length.
27. Group a list of strings by their first letter using `Collectors.groupingBy()`.
28. Partition a list of integers into even and odd using `partitioningBy()`.
29. Given a list of integers, find the product of all numbers using `reduce()`.
30. Join all strings in a list with " - " as separator using `Collectors.joining()`.
31. Remove duplicates, sort alphabetically, and collect results into a list.
32. From a list of names, find those that contain the letter "e" and are longer than 4 characters.
33. Limit the stream to the first 5 elements and collect them into a list.
34. Skip the first 3 elements in a list and print the rest.
35. Convert a list of lists into a single flat list of integers using `flatMap()`.
36. Given a list of custom `Employee` objects, extract all unique departments.
37. Given a list of `Employee` objects, sort them by salary (ascending).
38. Given a list of `Employee` objects, find the employee with the highest salary.
39. Group a list of employees by department using `Collectors.groupingBy()`.
40. Calculate the total salary expense per department using streams.
41. Find all strings that are palindromes from a given list.
42. From a list of numbers, create a list of numbers squared but only if they are even.
43. Given a list of sentences, count how many times each word occurs.
44. Flatten a list of arrays into a single list using Stream API.
45. Given a list of strings, remove duplicates ignoring case sensitivity.
46. Find the longest string in a list using Stream API.
47. Find the shortest string in a list using Stream API.
48. Given a list of integers, create a map of number → its cube.

49. Given a list of dates, find the latest and earliest dates using Stream API.
50. Given a list of integers, check if the list is sorted using Stream operations.

⭐ Set - 50 Questions

1. Print all elements of a `Set<Integer>` using Stream API.
2. Convert all strings in a `Set<String>` to uppercase using streams.
3. Filter and print only even numbers from a `Set<Integer>`.
4. Count how many strings in a `Set<String>` start with the letter “A”.
5. Convert a `Set<Integer>` to a sorted list using Stream API.
6. Sort elements of a `Set<String>` alphabetically and print them.
7. Remove all null or empty strings from a `Set<String>` using streams.
8. Find the maximum element in a `Set<Integer>` using streams.
9. Find the minimum element in a `Set<Integer>` using streams.
10. Calculate the sum of all integers in a `Set<Integer>` using `reduce()`.
11. Check if any element in a `Set<Integer>` is divisible by 5 using streams.
12. Check if all elements in a `Set<Integer>` are positive numbers.
13. Convert a `Set<Integer>` to a comma-separated string using `Collectors.joining()`.
14. Create a new `Set<Integer>` containing only odd numbers from an existing set.
15. Convert a `Set<String>` into a `Set<Integer>` representing each string’s length.
16. Count the total number of characters across all strings in a `Set<String>`.
17. Print only distinct lowercase versions of all strings in a `Set<String>`.
18. Convert a `Set<Integer>` to a `List<Integer>` using streams.
19. Limit a stream from a `Set<Integer>` to the first 3 elements and print them.
20. Skip the first 2 elements from a `Set<Integer>` stream and print the rest.
21. Given a `Set<String>`, create a `Map<String, Integer>` (key = string, value = length).
22. Given a `Set<Integer>`, find the average value using Stream API.
23. Given a `Set<Integer>`, find the second highest number using streams.

24. Given a `Set<Integer>`, find the second smallest number using streams.
25. Convert a `Set<List<Integer>>` into a flat `List<Integer>` using `flatMap()`.
26. Convert a `Set<String>` to a `TreeSet` sorted in reverse order using streams.
27. Partition a `Set<Integer>` into even and odd using `Collectors.partitioningBy()`.
28. Group strings in a `Set<String>` by their length using `Collectors.groupingBy()`.
29. Create a map of the first character to all strings starting with that character.
30. Remove all strings longer than 5 characters from a `Set<String>` using streams.
31. Find all strings in a `Set` that contain a digit using streams and regex.
32. Given two sets, find their intersection using Stream API.
33. Given two sets, find their union using Stream API.
34. Given two sets, find their difference (elements only in first set).
35. Given a set of numbers, find all numbers greater than the average value.
36. Find all palindromic strings in a `Set<String>` using streams.
37. Count how many elements in a `Set<String>` contain the letter “e”.
38. Join all strings from a `Set<String>` with “ - ” separator.
39. Find the longest string in a `Set<String>`.
40. Find the shortest string in a `Set<String>`.
41. Given a set of integers, create a map of integer → its square.
42. Given a set of integers, create a map of integer → whether it’s even or odd.
43. Flatten a `Set<Set<Integer>>` into a single `Set<Integer>`.
44. Given a set of numbers, find all prime numbers using streams.
45. Convert a `Set<String>` of comma-separated values into a flat `Set<String>`.
46. Filter out duplicate elements ignoring case sensitivity (e.g., "Java" and "java").
47. Given a set of words, count how many start and end with the same letter.
48. Remove all elements that contain special characters using Stream API.
49. Given a set of integers, sort them in descending order and collect to a list.
50. Given a set of file paths, extract only unique file extensions using streams.

⭐ Map - 50 Questions

1. Print all key-value pairs of a `Map<String, Integer>` using Stream API.
2. Print only the keys from a map using streams.
3. Print only the values from a map using streams.
4. Filter and print entries where the value is greater than 50.
5. Filter and print entries where the key starts with “A”.
6. Convert all keys to uppercase using streams.
7. Convert all values to their squares using streams.
8. Count how many entries have even values.
9. Collect keys of entries with value > 10 into a list.
10. Collect values of entries with key length > 3 into a set.
11. Convert all map entries into a list of strings formatted as “key=value”.
12. Find the maximum value in a map using streams.
13. Find the minimum value in a map using streams.
14. Check if any entry has a value equal to 100.
15. Check if all values in a map are positive.
16. Sum all integer values in a map using `mapToInt().sum()`.
17. Create a new map with only entries whose values are odd.
18. Create a map that swaps keys and values (value → key).
19. Convert a `Map<String, Integer>` into a `List<Integer>` of just values using streams.
20. Convert a `Map<String, Integer>` into a `Set<String>` of just keys using streams.
21. Given a `Map<String, Integer>`, sort entries by key in ascending order.
22. Given a `Map<String, Integer>`, sort entries by value in descending order.
23. Find the entry with the highest value in a map using streams.
24. Find the entry with the smallest key (alphabetically).
25. Collect keys of entries sorted by value into a list.
26. Convert a `Map<String, List<Integer>>` into a flat list of all integers using `flatMap()`.
27. Create a new `Map<String, Integer>` where the value is doubled.

28. Partition a map into two maps: one with even values and one with odd values.
29. Group map entries by the length of their keys using `Collectors.groupingBy()`.
30. Find the average of all numeric values in a map.
31. Merge two maps using streams (keys unique, sum values if duplicate).
32. Convert a list of `Map.Entry<String, Integer>` into a map using collectors.
33. Given a map of names to salaries, increase all salaries by 10%.
34. Find all keys whose corresponding values are prime numbers.
35. Create a comma-separated string of all “key=value” pairs.
36. Given a `Map<String, String>`, find all entries where value contains a given substring.
37. Convert a `Map<String, String>` into another map with key as uppercase and value as lowercase.
38. From a map of `String -> List<Integer>`, compute the sum of list elements for each key.
39. Given a map of `Department -> List<Employee>`, find total employees across all departments.
40. From a map of product → price, filter products costing more than 1000.
41. Given a map of country → population, find top 3 countries by population using streams.
42. Find all keys that have duplicate values in a map.
43. Flatten a `Map<String, List<String>>` into a single list of all values.
44. Create a map of word → frequency from a list of words using Stream API.
45. Convert a JSON-like string list (e.g., “key:value”) into a `Map<String, String>` using streams.
46. From a map of `String -> Integer`, find sum of all values grouped by key’s first letter.
47. Create a new map keeping only the first N entries from a sorted stream.
48. Given a map of student → marks, find all students who scored above average.
49. Given a nested map (e.g., `Map<String, Map<String, Integer>>`), flatten it into a list of all inner entries.
50. Given a map of file extension → list of files, count total files across all extensions.

★ Arrays - 50 Questions

1. Print all elements of an integer array using Stream API.
2. Print all strings from a string array using Stream API.
3. Convert all strings in an array to uppercase using streams.
4. Filter and print only even numbers from an integer array.
5. Filter and print only strings that start with the letter “A”.
6. Count how many numbers in an array are greater than 10.
7. Count how many strings in an array have length > 5.
8. Sort an integer array in ascending order using streams.
9. Sort a string array in descending (reverse alphabetical) order using streams.
10. Find the maximum number in an integer array using streams.
11. Find the minimum number in an integer array using streams.
12. Calculate the sum of all integers in an array using streams.
13. Calculate the average of all integers in an array using streams.
14. Convert an integer array into a list using streams.
15. Convert a string array into a set using streams.
16. Remove duplicate numbers from an integer array using `distinct()`.
17. Remove null or empty strings from a string array using streams.
18. Check if any element in an integer array is divisible by 5.
19. Check if all numbers in an integer array are positive.
20. Find the first element of a string array using streams.
21. Convert an integer array into an array of squares using `map()`.
22. Convert a string array into an array of their lengths using streams.
23. Filter and collect all even numbers into a new integer array.
24. Filter and collect all strings longer than 3 characters into a new array.
25. Join all strings in an array into a single comma-separated string.
26. Sort an array of strings by their length using streams.
27. Sort an array of integers in descending order using streams.
28. Find the second largest number in an integer array using streams.
29. Find the second smallest number in an integer array using streams.
30. Remove duplicate strings ignoring case sensitivity using streams.
31. Convert a 2D integer array into a flat list of integers using `flatMapToInt()`.
32. Flatten a 2D string array into a single list of strings using streams.

33. Convert an array of words into a map of word → length using streams.
34. Count how many times each string appears in an array using `Collectors.groupingBy()`.
35. From an array of integers, count how many are even vs odd using `partitioningBy()`.
36. Find all prime numbers in an integer array using Stream API.
37. Reverse-sort an array of strings by their length using streams.
38. Find the longest string in a string array using streams.
39. Find the shortest string in a string array using streams.
40. Create a map from an array of strings where key = first letter, value = list of words starting with that letter.
41. Merge two integer arrays into one without duplicates using streams.
42. Merge two string arrays and sort the result alphabetically.
43. Convert an array of integers to a comma-separated string of only odd numbers.
44. Find all palindromic strings in a string array using Stream API.
45. Given an array of strings, convert it to lowercase and remove duplicates.
46. Given an array of integers, group them by remainder when divided by 3.
47. Convert an array of integers into a `Map<Integer, Boolean>` indicating if each number is even.
48. Limit an integer stream to first 5 numbers and collect to an array.
49. Skip the first 3 elements from an integer array and print the rest.
50. Given an array of file names, extract only unique file extensions using Stream API.

Document By ***Suyash***😇