Java 8 Stream API – Practice Questions

1. Basic Stream Operations

1 Create a stream from a list of integers and print all elements.
2 Convert a list of strings to uppercase using map().
3 Filter all even numbers from a list.
4 Print the square of each number from a list.
5 Create a stream from an array and print all elements.
6 Count how many elements are in a list using streams.
7 Find the first element of a list using streams.
8 Skip the first 3 numbers from a list and print the rest.
9 Limit the stream to first 5 elements from a list.
10 Remove duplicates from a list using distinct().

2. Intermediate Stream Operations

1 Find the maximum and minimum number in a list.
2 Calculate the sum of numbers using reduce().
3 Find the average of numbers in a list.
4 Collect elements into a Set using collect().
5 Convert a list of strings into a single string using joining().
6 Count the number of words in a sentence using streams.
7 Convert a list of numbers into a Map where key = number, value = square.
8 Filter strings starting with a specific letter from a list.
9 Check if all numbers in a list are even using allMatch().
10 Check if any number in a list is greater than 100 using anyMatch().

3. Sorting and Advanced Collectors

1 Sort a list of integers in ascending order.
2 Sort a list of integers in descending order.
3 Sort a list of strings alphabetically.
4 Find the top 3 highest numbers in a list.
5 Group a list of strings by their length.
6 Group employees by department (custom object).
7 Count how many employees belong to each department.
8 Find the employee with the maximum salary.
9 Find the employee with the minimum salary.
10 Partition a list of numbers into even and odd using partitioningBy().

4. Reduction and Mapping

1 Find the product of all numbers in a list using reduce().
2 Find the longest string in a list.
3 Find the shortest string in a list.
4 Convert a list of integers to their binary string representation.
5 Flatten a list of lists into a single list using flatMap().
6 Find the frequency of each word in a string.

7. Find the frequency of each character in a string.
8. Remove null values from a list using streams.
9. Create a comma-separated string from a list of integers.
10. Find the sum of squares of all numbers in a list.

5. Real-World Scenarios

1. Given a list of employees, find all names who earn more than 50,000.
2. Group students by grade and count them.
3. Find the oldest and youngest employee.
4. Get a list of unique departments from employee objects.
5. Find the second-highest salary from a list of employees.
6. From a list of transactions, find the total transaction value.
7. Find the average salary of each department.
8. Get a list of employee names sorted by salary.
9. Partition employees into two groups: salary above 50,000 and below.
10. Find the top 2 employees in each department by salary.

6. Parallel Streams

1. Use parallelStream() to print numbers in parallel.
2. Compare execution time of stream() vs parallelStream() for a large list.
3. Find the sum of numbers from 1 to 1,000,000 using parallel streams.
4. Sort a list using parallel streams.
5. Demonstrate race conditions when using mutable objects in parallel streams.

7. Expert Level

1. Implement a word frequency counter using streams and Collectors.groupingBy().
2. Find the most repeated word in a string using streams.
3. Find the least repeated word in a string using streams.
4. Given a CSV file of employees (name, dept, salary), use streams to group by department and calculate average salary.
5. Build a custom collector to count vowels in a list of strings.
6. Implement pagination (skip + limit) on a list using streams.
7. Write a program to reverse words in a string using streams.
8. Extract all unique characters from a list of strings using flatMap().
9. Find the longest word in a sentence using streams.
10. Write a program that simulates SQL’s GROUP BY and HAVING using streams.