Core Java & OOP

1. Reverse a linked list (iterative and recursive).
2. Detect a cycle in a linked list (Floyd’s algorithm).
3. Find the first non-repeating character in a string.
4. Check if two strings are anagrams.
5. Implement LRU Cache using LinkedHashMap.

🔹 Java 8 & Functional Programming

1. Filter and sort a list of employees using streams.
2. Group employees by department and calculate average salary.
3. Use Optional to avoid NullPointerException.

 Flatten a list of lists using flatMap.

🔹 Collections & Algorithms

* 1. Find the top K frequent elements in an array.
  2. Sort a list of objects using a custom comparator.
  3. Remove duplicates from a list using Set.

🔹 Multithreading & Concurrency

* + 1. Implement a thread-safe counter using AtomicInteger.
    2. Solve the producer-consumer problem using wait() and notify().
    3. Use ExecutorService to run tasks in parallel.

Exception Handling & Flow Control

* + - 1. What happens if you return from a try block—will finally execute?
      2. What if System.exit() is called inside try—will finally execute?

🔹 Real-World Scenarios

* + - * 1. Design a rate limiter (e.g., 100 requests per minute).
        2. Build a URL shortener with basic CRUD and redirection.
        3. Cache method results using Spring Boot and Redis.