Backend live coding Logic Question :

1. **Flatten a Nested Array (poin 30)**
   Desc: Write a function that flattens an arbitrarily nested array of arrays.
   Example: Input: `[1, [2, [3, 4]], 5]`; Output: `[1, 2, 3, 4, 5]`

2. **Find All Pairs with a Given Sum (poin 30)**
   Desc : Given an array of integers, find all unique pairs that add up to a specific target sum.
   Example: Input: `[1, 2, 3, 4, 3]`, Target: `6`; Output: `[[2, 4], [3, 3]]`

3. **Check if Two Strings are Anagrams (poin 15)**
   Desc: Write a function that takes two strings and checks if they are anagrams of each other.
   Example: Input: `"listen"` and `"silent"`; Output: `true`

4. **Find the First Non-Repeating Character (poin 30)**
   Desc: Given a string, find the first character that does not repeat. Return the character, or `null` if all characters repeat.
   Example: Input: `"swiss"`; Output: `"w"`

5. **Sum of Two Numbers Equals Target (15)**
   Desc: Given an array of integers and a target integer, return indices of the two numbers that add up to the target.
   Example: Input: `[2, 7, 11, 15]`, Target: `9`; Output: `[0, 1]` (because `2 + 7 = 9`)

Backend Homework Interview Question

1. **Create a Middleware for Logging without using any library (poin 10)**
   The middleware should log each incoming request's details to the console and then pass control to the next middleware.

2. **Create a Recursive Function to Read Nested Directories (poin 40)**
   Use the `fs.promises` API to handle reading asynchronously, and ensure errors are caught and logged.

3. **Create a Rate Limiter Middleware (poin 20)**
   For example, limit users to 10 requests per minute. If they exceed this, respond with a `429 Too Many Requests` status.

4. **Implement Caching with Memory (poin 40)**
   Cache the response for each unique request for 1 minute. If the same request comes within that time frame, return the cached response.

5. **Data Validation and Sanitization with Zod or Joi (poin 10)**
   Return a response indicating success if data is valid, or return an error if validation fails.

Backend Scraping Interview Question

- web scraping menggunakan bun.js dan selenium
    1. navigate https://www.logammulia.com/id/harga-emas-hari-ini
    2. ambil Harga emas Batangan hari ini
    3. simpan data hasil scraping ke dalam db postgres

- buat api untuk hitung simulasi Harga emas Batangan menggunakan bun.js berdasarkan Harga emas yang sudah disimpan di db
    1. method : post
    2. Request :
        - berat emas per batang dalam satuan gram (decimal)
        - jumlah batang yang dibeli (int)
        - flag include pajak (Boolean)
    3. response :
        - Harga dasar emas (decimal)
        - total Harga emas (decimal)