Exercises for Map, Filter, and Reduce in JavaScript

1. **Use `map` to get an array of product titles**
    - Extract only the titles of all products.
    - **Sample Output:** `["Sony LED 40 inch", "Mobile", "Bike"]`

2. **Use `filter` to get all products that have variations in black color**
    - Find products that have at least one variation with the color "black".
    - **Sample Output:** `[ { id: 101, title: "Sony LED 40 inch", ... }, { id: 102, title: "Mobile", ... }, { id: 103, title: "Bike", ... } ]`

3. **Use `reduce` to calculate the total stock of all products**
    - Sum the quantity of all variations across all products.
    - **Sample Output:** `20`

4. **Use `map` and `reduce` to get the average rating of each product**
    - Consider only the reviews where `status` is `true`.
    - Calculate the average rating for each product.
    - **Sample Output:** `[ { title: "Sony LED 40 inch", averageRating: 4.5 }, { title: "Mobile", averageRating: 4.0 }, { title: "Bike", averageRating: 4.0 } ]`

5. **Use `filter` to get products that have at least one review with a rating of 5.0**
    - Find products that contain at least one review with a 5-star rating.
    - **Sample Output:** `[ { id: 101, title: "Sony LED 40 inch", ... } ]`

6. **Use `map` to format variations with product name**
    - Create a new array where each product contains only the title and its variations (color, price, quantity).
    - **Sample Output:** `[ { title: "Sony LED 40 inch", variations: [ { color: "black", price: 50000, quantity: 5 }, ... ] } ]`

7. **Use `reduce` to get the total revenue if all items were sold**

    - Calculate the total revenue by multiplying price and quantity for each variation, then summing it for all products.
    - **Sample Output:** `850000`

8. **Use `filter` to get all products that have more than 5 items in any variation**

    - Find products where at least one variation has a quantity greater than 5.
    - **Sample Output:** `[ { id: 101, title: "Sony LED 40 inch", ... } ]`

9. **Use `map` to get a summary of each product with total variations and total reviews**

    - Create an array where each product has the title, total variations count, and total reviews count.
    - **Sample Output:** `[ { title: "Sony LED 40 inch", totalVariations: 3, totalReviews: 3 }, ... ]`

10. **Use `reduce` to find the product with the highest total stock**

    - Identify the product with the maximum sum of all variation quantities.
    - **Sample Output:** `{ title: "Sony LED 40 inch", totalStock: 14}`