◆ Basic Aggregation Tasks

1 Find the total sales (`$sum`) from the `orders` collection.
Hint: Use `$group` to sum `totalPrice`.

2 Find the average price (`$avg`) of all products in the `products` collection.
Hint: Use `$group` to calculate the average of `price`.

3 Find the most expensive (`$max`) and cheapest (`$min`) product in the `products` collection.
Hint: Use `$group` and `$max/$min` on `price`.

4 Get the first (`$first`) and last (`$last`) order placed in the `orders` collection.
Hint: Group by a dummy value and sort by `orderDate`.

◆ Arithmetic & Comparison Operators

5 Calculate the discount price for all products (10% off) using `$multiply` and `$subtract`.
Hint: Use `$project` to create a new field like `discountPrice`.

6 Find all orders where the total price is greater than $500 using `$gt`.
Hint: Use `$match` with `{ totalPrice: { $gt: 500 } }`.

7 Find users who are older than 30 using `$gte`.
Hint: Use `$match` on `age`.

8 Find all products that cost between $50 and $500 using `$gte` and `$lte`.
Hint: Use `$match` with `{ price: { $gte: 50, $lte: 500 } }`.

◆ Boolean Operators

9 Find all orders that are either 'pending' or 'shipped' using `$or`.
Hint: Use `$match` with `{ $or: [ { status: "pending" }, { status: "shipped" } ] }`.

10 Find all users who are NOT using a Gmail email using `$not`.
Hint: Use `$match` and `$regexMatch` with `$not`.

◆ String Manipulation

11 Concatenate first name and last name (`$concat`) to create a `fullName` field in users.
Hint: Use `$project`.

12 Convert all product names to uppercase (`$toUpper`).
Hint: Use `$project` to modify `productName`.

13 Extract the domain from users' emails using `$substr`.
Hint: Find `@` position and use `$substr`.

◆ Array Operators

14 Find the number of orders per user using `$group` and `$count`.
Hint: Group by `userId`.

15 Find the second ordered product using `$arrayElemAt`.
Hint: Sort `orders` by `orderDate` and use `$arrayElemAt`.

16 Get the first 3 products using `$slice`.
Hint: Use `$project` and `$slice`.

◆ Date Manipulation

17 Extract the year from all order dates using `$year`.
Hint: Use `$project` on `orderDate`.

18 Convert order date into a custom string format (`$dateToString`).
Hint: Use `$project` with `{ format: "%Y-%m-%d", date: "$orderDate" }`.

◆ Conditional & Pipeline Tasks

19 Create a `statusLabel` field that says 'High Value' for orders above $500 and 'Low Value' otherwise using `$cond`.
Hint: Use `$project` and `$cond`.

20 Sort products by price in descending order (`$sort`).
Hint: Use `{ price: -1 }` in `$sort`.