Problem 1: Temperature Converter (Parent-Child Communication)

Objective:

Create an application where a parent component (`TemperatureApp`) manages the temperature state, and two child components (`CelsiusInput` and `FahrenheitInput`) display and update it.

Requirements:

1. The parent component holds the temperature state in **Celsius**.
2. The `CelsiusInput` component displays and allows editing the temperature in Celsius.
3. The `FahrenheitInput` component converts and displays the temperature in Fahrenheit (formula: `F = (C * 9/5) + 32`).
4. Changes in either input should update the state in the parent and reflect in both components.

Expected Behavior:

- Typing in `CelsiusInput` updates `FahrenheitInput` in real-time.
- Typing in `FahrenheitInput` converts the value back to Celsius and updates both inputs.

Hint:

- Use `onTemperatureChange` callbacks passed from the parent to child components.

Problem 2: Shopping Cart (Sibling Components Sharing State)

Objective:

Build a shopping cart system where:

- A `ProductList` component displays products (name, price, "Add to Cart" button).

- A `Cart` component shows the items added and the total price.

Requirements:

1. The cart state (items, total) should be managed in a **parent component** (`ShoppingApp`).
2. Clicking "Add to Cart" in `ProductList` updates the cart state in the parent.
3. The `Cart` component receives the cart data as props and displays it.
4. Implement a "Remove Item" button in the `Cart` component to modify the state.

Expected Output:

```
ProductList:
- [ ] Product A ($10) [Add to Cart]
- [ ] Product B ($20) [Add to Cart]

Cart:
- Product A ($10) [Remove]
- Product B ($20) [Remove]
Total: $30
```

Hint:

- Lift state up to `ShoppingApp` and pass `addToCart`/`removeFromCart` functions as props.