Great approach! Here are structured problems for each React Hook so you can practice them effectively:

**🔵 1. useState Problem (Counter with Step)**

**Task:**  
Create a counter component that:

* Starts from 0
* Has buttons to **increase** and **decrease** the count
* Allows the user to set a custom step value (e.g., +5, -5) via an input field

👉 **Bonus:** Show a message when the count reaches 10: "You've reached 10!"

**🟢 2. useEffect Problem (Fetch Data & Cleanup)**

**Task:**  
Build a component that:

* Fetches and displays user data from https://jsonplaceholder.typicode.com/users/1 when it mounts
* Displays a "Loading..." message before data loads
* Cleans up any ongoing requests when the component unmounts

👉 **Bonus:** Add a button to **refetch the data** when clicked

**🟡 3. useContext Problem (Theme Toggle)**

**Task:**

* Create a **ThemeContext** that provides "light" or "dark" mode
* Use useContext in a child component to display the **current theme**
* Add a button to toggle between light and dark mode

👉 **Bonus:** Change the **background color** of the page based on the theme

**🟠 4. useRef Problem (Focus & Persisting Values)**

**Task:**

* Create an **input field** that automatically gets **focused** when the page loads
* Store the **number of times** the user clicked a button, but **do not trigger re-renders**

👉 **Bonus:** Show the number of clicks without making the component re-render

**🔴 5. useReducer Problem (Todo List)**

**Task:**

* Build a **todo list** using useReducer
* The reducer should handle **adding**, **removing**, and **marking tasks as done**
* Display the tasks in a list

👉 **Bonus:** Add a "Clear Completed" button to remove only completed tasks

**🔵 6. useCallback & useMemo Problem (Expensive Calculation)**

**Task:**

* Create a **number input field** where users enter a value
* Display the **factorial** of the entered number
* Use useMemo to **optimize** the factorial calculation
* Use useCallback to optimize a **button click handler**

👉 **Bonus:** Show how many times the factorial function was re-calculated

**🟣 7. Custom Hook Problem (UseWindowSize)**

**Task:**

* Create a **custom hook** called useWindowSize that:
  + Returns the **current width and height** of the window
  + Updates the values when the window is resized
* Use this hook in a component to display the screen size

👉 **Bonus:** Show a message **"Mobile View"** if the width is less than 768px

**Try these out! Let me know if you need hints or explanations. 🚀**