***Discuss how each hooks works***

UseContext is a way to manage state globally, it can be used togething with the useState hook to share state between deeply nested components more easily than with useState alone. When using the context, it requires 3 steps: creating, providing, and consuming the context, integrating the context into the application consider that it adds complexitity. UseReducer is a hook similar to useState, it allows for custom logic. If you find yourself keeping track of multiple pieces of state that rely on complex logic, useReducer may be useful. UseReducer accepts two arguments useReducer(<reducer>, <initialState>).

***Discuss how each would help in dealing with state of a larger React app***

When using both useContext and useReducer in a larger application they help to avoid prop drilling, with useReducer you create state and dispatch, useReducer uses reducer function and intial state. Using stateContext and DispatchContext we can make them available to all child components. You can access state and dispatch out of stateContext and dispatchContext with useContext hook.

***Discuss how you would re-factor the ToDo in using one or both of these hooks***

Using these concepts in the to do list would help refactor the syntax to avoid prop drilling, and passing custom functions with ease. It can make the code look cleaner and more readable while also making the application easier to test.