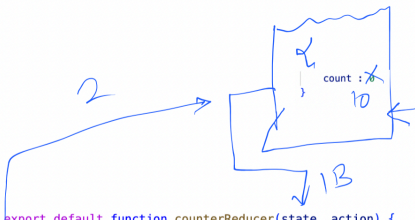
```jsx
import { useState } from "react"

export default function Counter() {
    let [count, setCount] = useState(0); // to introduce state variable in functional
    let [name, setName] = useState("Roger");


    return <div>
        Name: {name} <br />
        Count : {count} <br />
        <button type="button" onClick={() => setCount(count + 1)}>Increment</button>
    </div>
}
```

```jsx
class Counter extends Component {
    state = {
        name : "Roger",
        count: 0
    }

    updateCount() {
        this.setState({
            count: this.state.count + 1
        })
    }

    render() {
        return <div>
            Name: {this.state.name} <br />
            Count: {this.state.count} <br />
            <button type="button" onClick={() => updateCount()}>Increment</button>
        </div>
    }
}
```


```jsx
const initialState = {
    count : 0
}

export default function ReducerComponent() {
    let [state, dispatch] = useReducer(countReducer, initialState);

    return <div>
        Count: {state.count} <br />
        <button type="button" onClick={() => dispatch({type:'INCREMENT', payload: 10})}>Increment</button>
        <button type="button" onClick={() => dispatch({type:'DECREMENT'})}>Decrement</button>
        <button type="button" onClick={() => dispatch({type:'RESET'})}>CLEAR</button>
    </div>
}
```

```jsx
export default function counterReducer(state, action) {
    switch(action.type) {
        case 'INCREMENT':
            return {
                count: state.count + action.payload
            }
        case 'DECREMENT':
            return {
                count: state.count - 1
            }
        case 'RESET':
            return {
                count: 0
            }
        default:
            return state;
    }
}
```




```jsx
const initialState = {
    count : 0
}

export default function ReducerComponent() {
    let [state, dispatch] = useReducer(countReducer, initialState);

    return <div>
        Count: {state.count} <br />
        <button type="button" onClick={() => dispatch({type:'INCREMENT', payload: 10})}>Increment</button>
        <button type="button" onClick={() => dispatch({type:'DECREMENT'})}>Decrement</button>
        <button type="button" onClick={() => dispatch({type:'RESET'})}>CLEAR</button>
    </div>
}
```

```jsx
export default function counterReducer(state, action) {
    switch(action.type) {
        case 'INCREMENT':
            return {
                count: state.count + action.payload
            }
        case 'DECREMENT':
            return {
                count: state.count - 1
            }
        case 'RESET':
            return {
                count: 0
            }
        default:
            return state;
    }
}
```