

```
export default function countReducer(state, action) {
  switch(action.type) {
    case 'INCREMENT':
      return {
        count: state.count + action.payload
      }
    case 'DECREMENT':
      // assume no payload
      return {
        count: state.count - 1
      }
    case 'RESET':
      return {
        count: 0
      }
    default: return state;
  }
}
```

```
export default function Counter() {
  let [state, dispatch] = useReducer(countReducer, {count: 0});
  return (
    <div>
      Count: {state.count} <br />
      <button type='button' onClick={() => dispatch({type: 'INCREMENT', payload: 5})}>
        Increment
      </button>
      <button type='button' onClick={() => dispatch({type: 'DECREMENT'})}>
        DECREMENT
      </button>
      <button type='button' onClick={() => dispatch({type: 'RESET'})}>
        Reset
      </button>
    </div>
  )
}
```

{count: 0}

HEAP

1A

```
export default function countReducer(state, action) {
  switch(action.type) {
    case 'INCREMENT':
      return {
        count: state.count + action.payload
      }
    case 'DECREMENT':
      // assume no payload
      return {
        count: state.count - 1
      }
    case 'RESET':
      return {
        count: 0
      }
    default: return state;
  }
}
```

1B

```
export default function Counter() {
  let [state, dispatch] = useReducer(countReducer, {count: 0});
  return (
    <div>
      Count: {state.count} <br />
      <button type='button' onClick={() => dispatch({type: 'INCREMENT', payload: 5})}>
        Increment
      </button>
      <button type='button' onClick={() => dispatch({type: 'DECREMENT'})}>
        DECREMENT
      </button>
      <button type='button' onClick={() => dispatch({type: 'RESET'})}>
        Reset
      </button>
    </div>
  )
}
```

Action

{count: 5}

HEAP

1A

```
export default function countReducer(state, action) {
  switch(action.type) {
    case 'INCREMENT':
      return {
        count: state.count + action.payload
      }
    case 'DECREMENT':
      // assume no payload
      return {
        count: state.count - 1
      }
    case 'RESET':
      return {
        count: 0
      }
    default: return state;
  }
}
```

1B

```
export default function Counter() {
  let [state, dispatch] = useReducer(countReducer, {count: 0});
  return (
    <div>
      Count: {state.count} <br />
      <button type='button' onClick={() => dispatch({type: 'INCREMENT', payload: 5})}>
        Increment
      </button>
      <button type='button' onClick={() => dispatch({type: 'DECREMENT'})}>
        DECREMENT
      </button>
      <button type='button' onClick={() => dispatch({type: 'RESET'})}>
        Reset
      </button>
    </div>
  )
}
```

{count: 4}

HEAP