

```

export default function CartReducer(state, action) {
  switch (action.type) {
    case 'ADD_TO_CART':
      let total = state.total + action.payload.amount
      return {
        items: [...state.items, action.payload],
        total
      }
    case 'INCREMENT':
      let items = state.items;
      items.forEach(item => {
        if (item.id === action.payload) {
          item.qty++;
          item.amount = item.price * item.qty;
        }
      });
      // need to compute total
      return {
        items,
        total: state.total
      }
    case 'CLEAR_CART':
      return {
        items: [],
        total: 0.0
      }
    default:
      return state;
  }
}

```

B

```

const initialState = {
  items: [],
  total: 0.0
}
let [state, dispatch] = useReducer(CartReducer, initialState);

```

A B

items: [],  
total: 0.0

```

let item = {id, name, price, amount: price, qty: 1}
dispatch({type: 'ADD_TO_CART', payload: item})

```

```

export default function CartReducer(state, action) {
  switch (action.type) {
    case 'ADD_TO_CART':
      let total = state.total + action.payload.amount
      return {
        items: [...state.items, action.payload],
        total
      }
    case 'INCREMENT':
      let items = state.items;
      items.forEach(item => {
        if (item.id === action.payload) {
          item.qty++;
          item.amount = item.price * item.qty;
        }
      });
      // need to compute total
      return {
        items,
        total: state.total
      }
    case 'CLEAR_CART':
      return {
        items: [],
        total: 0.0
      }
    default:
      return state;
  }
}

```

```

const initialState = {
  items: [],
  total: 0.0
}
let [state, dispatch] = useReducer(CartReducer, initialState);

```

items: [],  
total: 0.0

```

let item = {id, name, price, amount: price, qty: 1}
dispatch({type: 'ADD_TO_CART', payload: item})

```

①

Action

②