Panel 1

B   A

```js
export default function countReducer(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;
    }
}
```

```js
let initialState = {
    count : 0
}
export default function CounterComponent() {
    let [state, dispatch] = useReducer(countReducer, initialState);

    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>
    )
}
```

C

HEAP
```
{
    count : 0
}
```

Panel 2

A

```js
export default function countReducer(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;
    }
}
```

```js
let initialState = {
    count : 0
}
export default function CounterComponent() {
    let [state, dispatch] = useReducer(countReducer, initialState);

    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>
    )
}
```

B

2

HEAP
```
{
    count : 0  5
}
```

Panel 3

A

```js
export default function countReducer(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;
    }
}
```

```js
let initialState = {
    count : 0
}
export default function CounterComponent() {
    let [state, dispatch] = useReducer(countReducer, initialState);

    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>
    )
}
```

?

HEAP
```
{
    count : 0  5
}
```
4