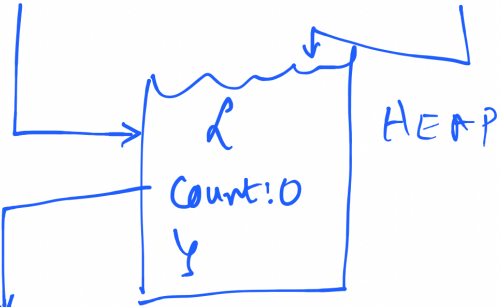


```
export default function countReducer(state, action) {
  switch (action.type) {
    case 'INCREMENT':
      return {
        count: state.count + action.payload
      }
    case 'DECREMENT':
      return {
        count: state.count - 1
      }
    case 'RESET':
      return {
        count: 0
  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

```
let initialState = {
  count: 0
}
let [state, dispatch] = useReducer(countReducer, initialState);
```

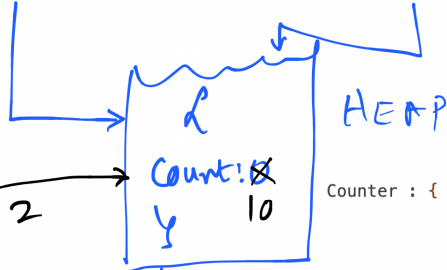
Counter : { state.count }



```
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;
  }
}
```

```
let initialState = {
  count: 0
}
let [state, dispatch] = useReducer(countReducer, initialState);
```

Counter : { state.count }



```
<button type='button'
  onClick={() => dispatch({type: 'INCREMENT', payload: 10})}>
  +
</button>
```

1A

