PART1 - REDUX

1.) Can you provide a summary of what is happening in this function code?

The countReducer function is a Redux reducer that checks if the action type is 'increment.' If the action type is 'increment,' it returns a new state object with the value incremented by 1. If the action type is not 'increment,' it does not make any changes

2.) Add one action that tells the reducer to reduce the state value by 1

```
const countReducer = (state = initialState,
action) => {
   if (action.type === 'decrement') {
     return {
       value: state.value - 1,
     };
   }
   return state;
 };
```

3.) Add one action that tells the reducer to reset the state

```
const countReducer = (state = initialState,
action) => {
   if (action.type === 'reset') {
     return {
       value: 0,
     };
   }
   return state;
 };
```

PART 2:

1.) Can you provide a brief summary of what is happening on line 34,39..

Within the classInfo function, we are creating a variable studentsCount and a updater function setStudentsCount and we are initiating the studentCounts to 0 using the useState hook.

This means that studentsCount = 0.

If any changes in studentsCount occurs, the setStudentCount will trigger a re-render and update this student count.

2a.) Pseudocode will look something similar to this.
I first need to create a function to handle adding student so i will call it handleStudentCount and then the student count has been set to 0 using useState. So now when the Add student button is clicked, it will trigger the handle StudentCount function.. Where i want to iterate through the array and check if student.present = true. If true, then it will increase the student count by 1, if not, it will keep it at current value.

```
const classInfo= () => {
 let [studentCount, setStudentCount] = useState(0);

 // function to handle adding a student
 const handleStudentCount = () => {

 // Iterate through the array and check if student is
present probably using a for loop

 students.forEach(function(student) {
   // Check if the student is present
   if (student.present === true) {
```

```
    // // If student is present, increment studentCount
by 1
    setStudentCount(studentCount + 1);
  }
  else {
    // If student is not present, do nothing
    setStudentCount(studentCount);
  }

 return (
  <div>
    <p>Student Count: {studentCount}</p>
    <button onClick={(handleStudentCount)}>Add
Student</button>
  </div>
```

b.) How do you ensure the function is triggered when the button is clicked,
I can ensure the function is triggered by making sure that the handleStudentCount function is placed within the button tag.

```
<button onClick={(handleStudentCount)}>Add
Student</button>
```

Then I can see if this re-renders and updates on the UI,
I also test this with react library or jest. By checking if an event is fired when the Add Student button is clicked.

c.) How will you update the state with the result of your function.
If student is present, then that means present ==== true

```
setStudentCount(studentCount + 1);
```

Else- means no change to be made to the student count

```
setStudentCount(studentCount);
```

Part 3:

A change of code was made on line 174- can you briefly explain what to do.

1.) The action.payload must be declared in the action object in this Redux setup. If it is not defined, then it shows as state.value + undefined which may give NaN- not a number . However, if action.payload is defined as a number, the state.value will increment by the value of action.payload, triggering a re-render in the UI.

2.) If we want to use the dispatch and not the useState

I will need to ensure i have npm installed react-redux
then we need to set up the action, store.js and the reducer.js. For REDUX
I will need to import useDispatch and UseSelector form react-redux
Then for the handleCLick function, i will dispatch Increment

```
const classInfo= () => {

   // I will have useSelector to get the studentCount
from the store
 const studentCount = useSelector(state => state.value);
 // then i have to useDispatch to dispatch the action to
the reducer
 const dispatch = useDispatch();
 // then i have to create a function to handle the
studentCount
 const handleStudentCount = () => {
   // then iterate through the array and check if student
is present
   for each student, if present === true {
     // Increment studentCount  by dispatching the action
'increment'
     dispatch(increment());
   }}
```

```
 return (
   <div>
     <p>Student Count: {studentCount}</p>
     <button onClick={(handleStudentCount)}>Add
Student</button>
   </div>
```

3.) Figure 4- Figure 4 is more suited for increment action because our initialState has an object of key value pair {value:0} which is where the state is declared. This means the state.value = 0. If the payload is defined in the action, then this function in figure 4 will run and will increment the state.value by the payload each time the button is clicked.

```
if (action.type === 'increment') {
   return {
     value: state.value + action.payload,
   };
 }
```

If payload is added to the action, then when dispatching, you can dispatch Incrment as:

```
 dispatch({
   type: 'increment',
   payoad: 1, // or any other value you want to pass
 });
```

Figure 5 however uses a payload which is undefined, therefore the and the payload remains at NaN(Not a number) regardless of how often the button is clicked.