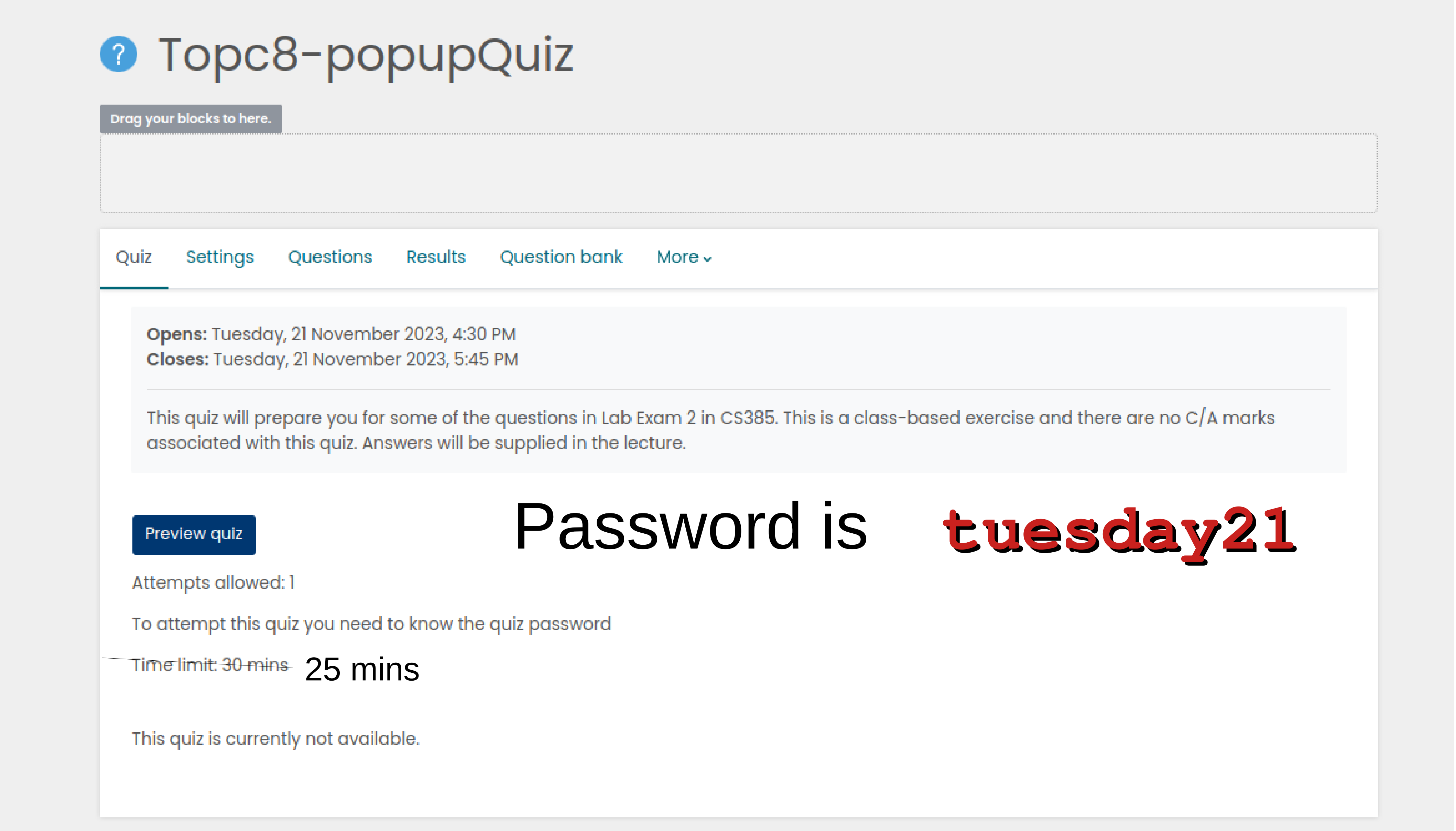
# CS385 Lecture 16

## In-class popup quiz, popup-quiz review, project update, plans for Lecture 17-18

Complete slides for Lecture 16 will be available AFTER the lecture

# Popup Quiz – 25 minutes
# Please work together on this



## Topc8-popupQuiz

**Opens:** Tuesday, 21 November 2023, 4:30 PM
**Closes:** Tuesday, 21 November 2023, 5:45 PM

This quiz will prepare you for some of the questions in Lab Exam 2 in CS385. This is a class-based exercise and there are no C/A marks associated with this quiz. Answers will be supplied in the lecture.

Password is **tuesday21**

Attempts allowed: 1

To attempt this quiz you need to know the quiz password

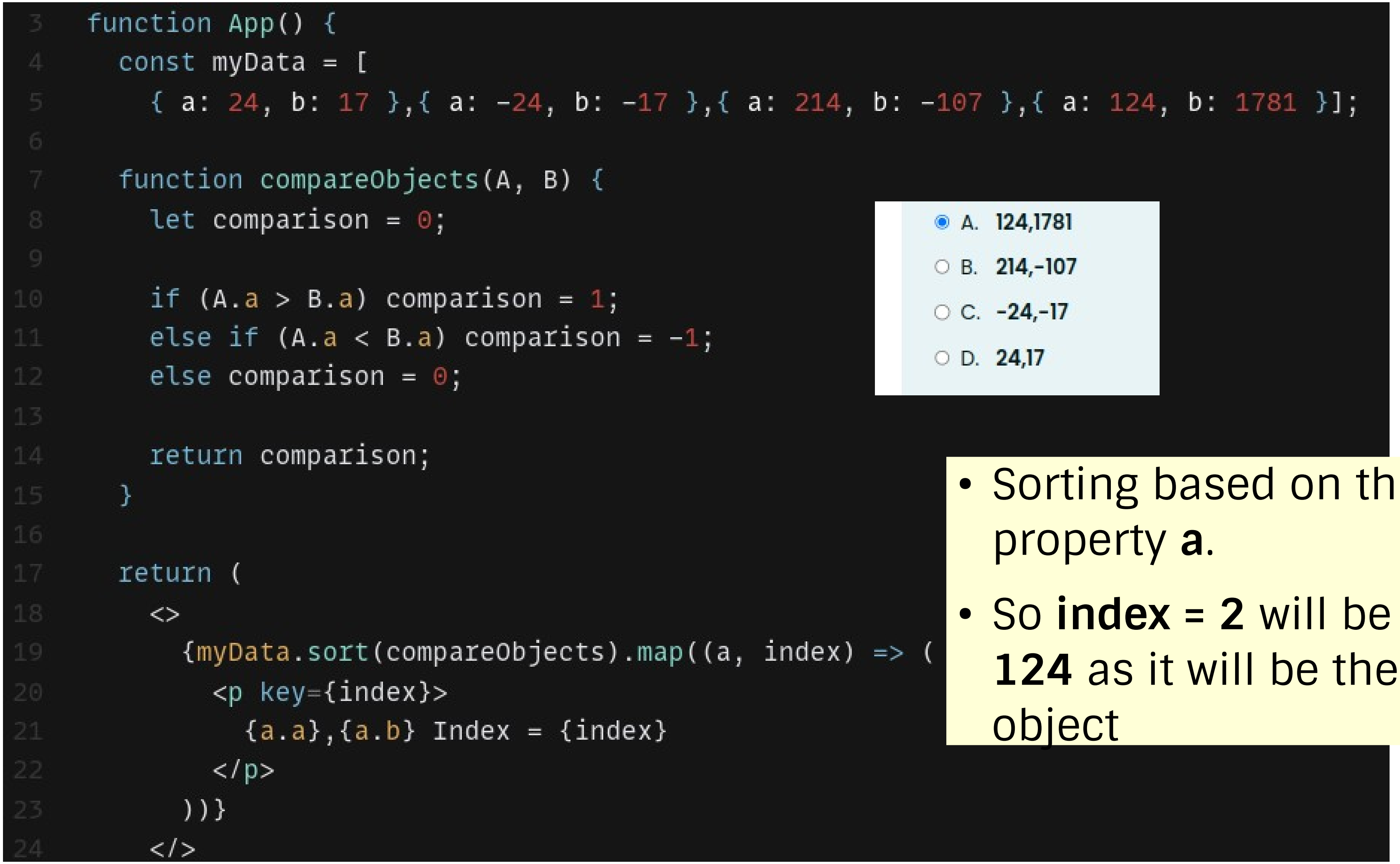
~~Time limit: 30 mins~~ 25 mins

This quiz is currently not available.

# CS385 Lecture 16

Answers to the popup quiz will be discussed here

# Topic 8 – Question 1



```
function App() {
  const myData = [
    { a: 24, b: 17 },{ a: -24, b: -17 },{ a: 214, b: -107 },{ a: 124, b: 1781 }];

  function compareObjects(A, B) {
    let comparison = 0;

    if (A.a > B.a) comparison = 1;
    else if (A.a < B.a) comparison = -1;
    else comparison = 0;

    return comparison;
  }

  return (
    <>
      {myData.sort(compareObjects).map((a, index) => (
        <p key={index}>
          {a.a},{a.b} Index = {index}
        </p>
      ))}
    </>
```

● A. 124,1781

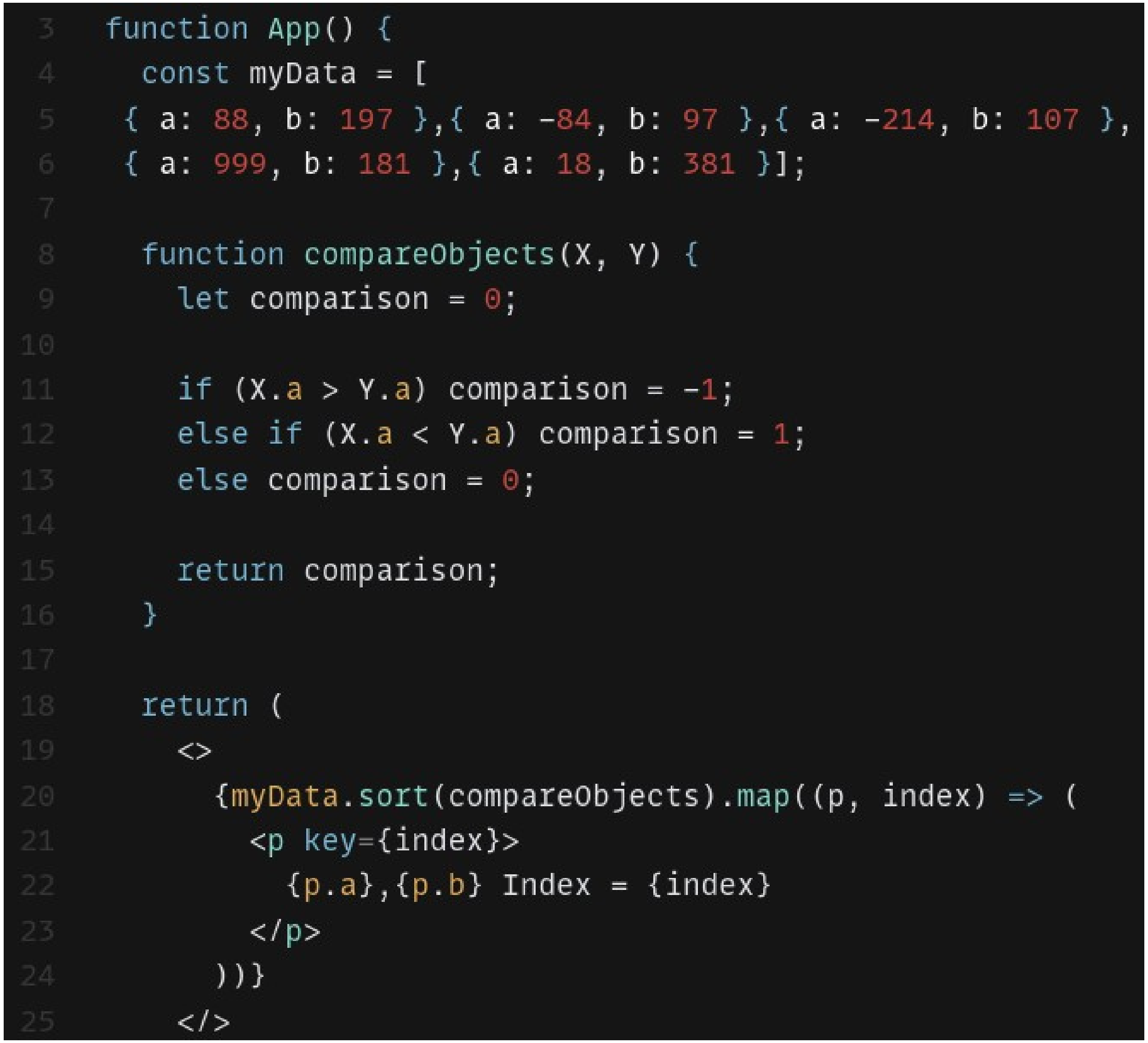
○ B. 214,-107

○ C. -24,-17

○ D. 24,17

- Sorting based on the property **a**.
- So **index = 2** will be **a = 124** as it will be the 3rd object

# Topic 8 – Question 2



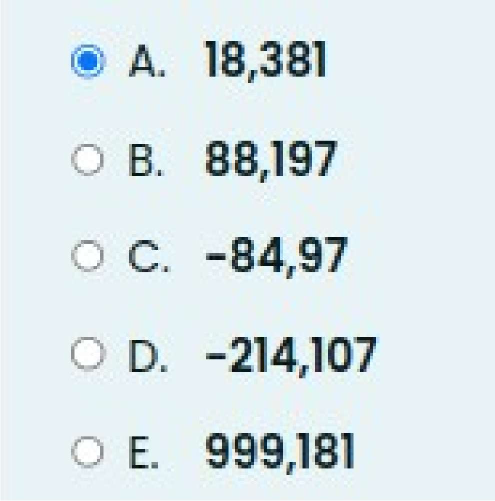
```
function App() {
  const myData = [
 { a: 88, b: 197 },{ a: -84, b: 97 },{ a: -214, b: 107 },
 { a: 999, b: 181 },{ a: 18, b: 381 }];

  function compareObjects(X, Y) {
    let comparison = 0;

    if (X.a > Y.a) comparison = -1;
    else if (X.a < Y.a) comparison = 1;
    else comparison = 0;

    return comparison;
  }

  return (
    <>
      {myData.sort(compareObjects).map((p, index) => (
        <p key={index}>
          {p.a},{p.b} Index = {index}
        </p>
      ))}
    </>
```



● A. 18,381

○ B. 88,197

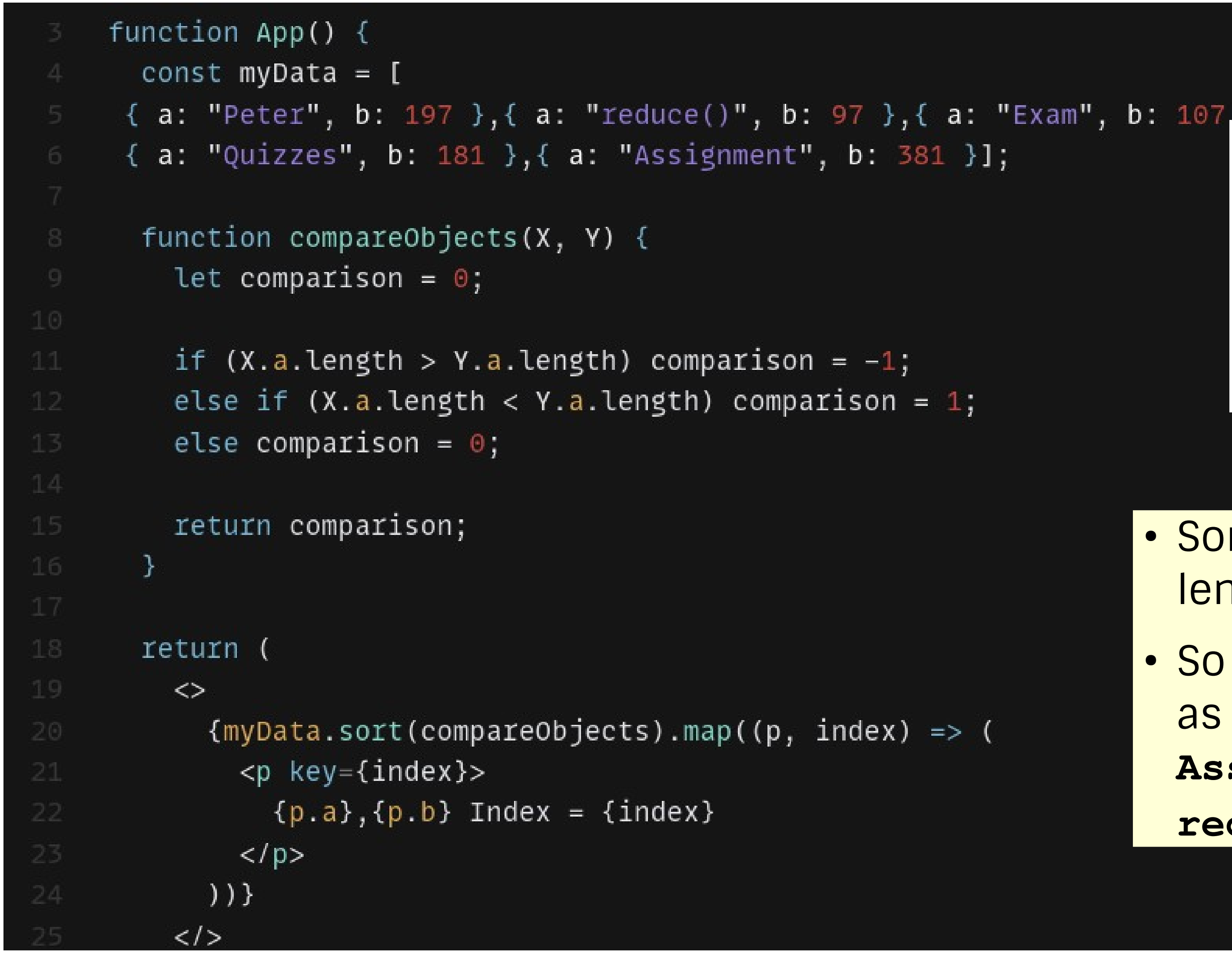
○ C. -84,97

○ D. -214,107

○ E. 999,181

- Sorting based on the property **a** (DESCENDING).
- So **index = 2** will be **a = 18** as it will be the 3rd object. 999, 88, then 18

# Topic 8 – Question 3



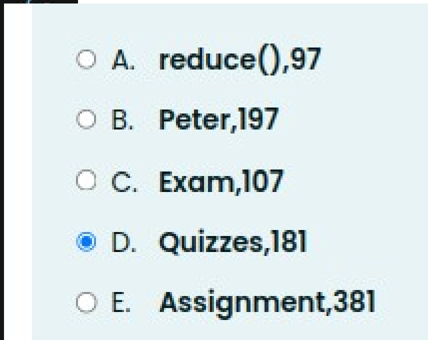
```
function App() {
  const myData = [
 { a: "Peter", b: 197 },{ a: "reduce()", b: 97 },{ a: "Exam", b: 107 },
 { a: "Quizzes", b: 181 },{ a: "Assignment", b: 381 }];

  function compareObjects(X, Y) {
    let comparison = 0;

    if (X.a.length > Y.a.length) comparison = -1;
    else if (X.a.length < Y.a.length) comparison = 1;
    else comparison = 0;

    return comparison;
  }

  return (
    <>
      {myData.sort(compareObjects).map((p, index) => (
        <p key={index}>
          {p.a},{p.b} Index = {index}
        </p>
      ))}
    </>
```



○ A. reduce(),97

○ B. Peter,197

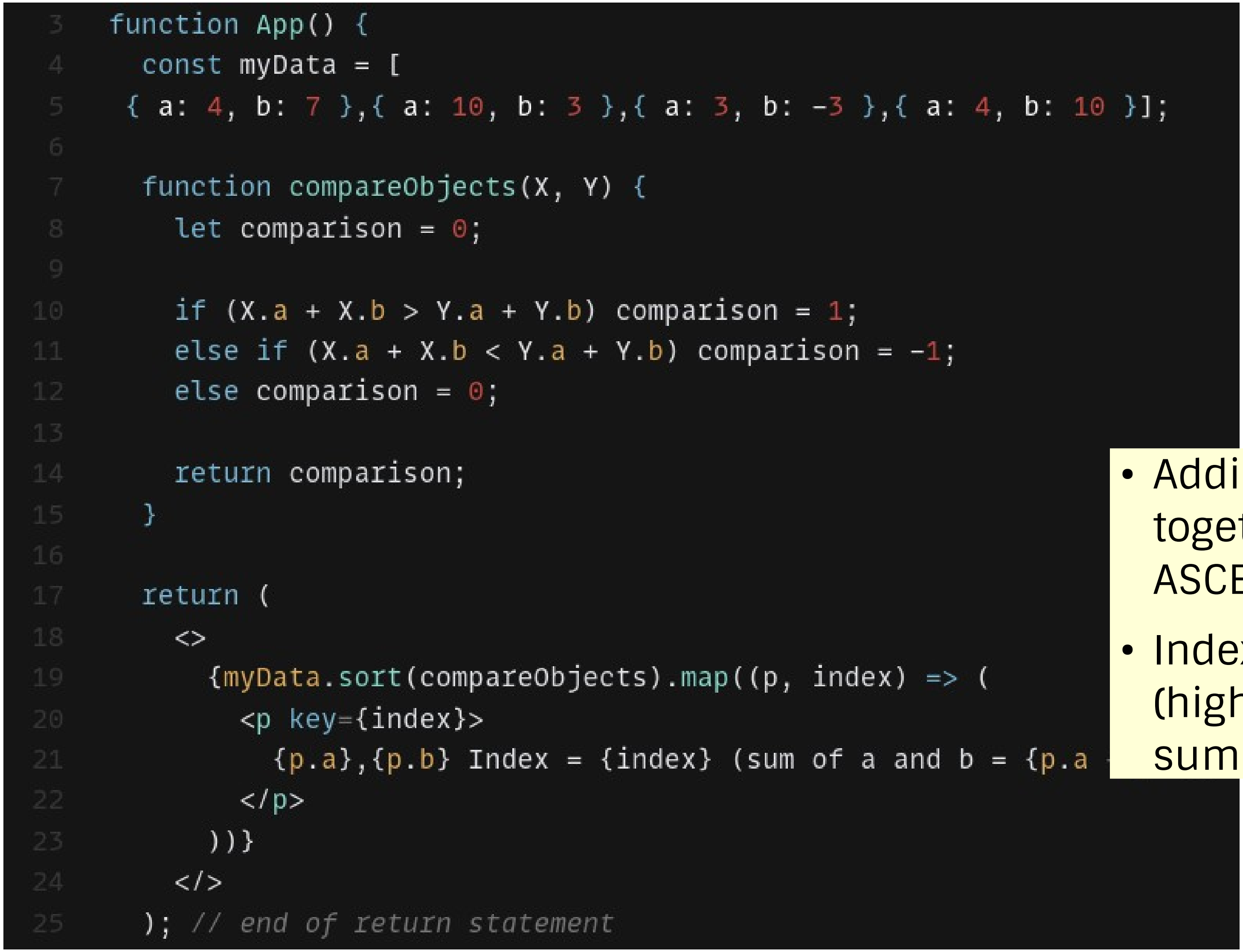
○ C. Exam,107

● D. Quizzes,181

○ E. Assignment,381

- Sorting based on the string length of **a** (DESCENDING).
- So **index = 2** will be **a = Quizzes** as it will be the 3rd object. **`Assignment, reduce(),Quizzes,`**

# Topic 8 – Question 4 (“trick”)



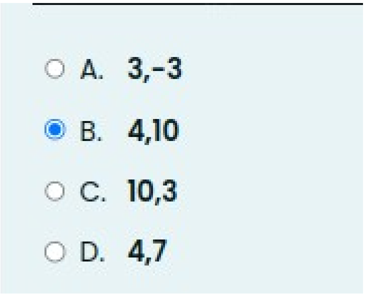
```
function App() {
  const myData = [
 { a: 4, b: 7 },{ a: 10, b: 3 },{ a: 3, b: -3 },{ a: 4, b: 10 }];

  function compareObjects(X, Y) {
    let comparison = 0;

    if (X.a + X.b > Y.a + Y.b) comparison = 1;
    else if (X.a + X.b < Y.a + Y.b) comparison = -1;
    else comparison = 0;

    return comparison;
  }

  return (
    <>
      {myData.sort(compareObjects).map((p, index) => (
        <p key={index}>
          {p.a},{p.b} Index = {index} (sum of a and b = {p.a
        </p>
      ))}
    </>
  ); // end of return statement
```



○ A. 3,-3

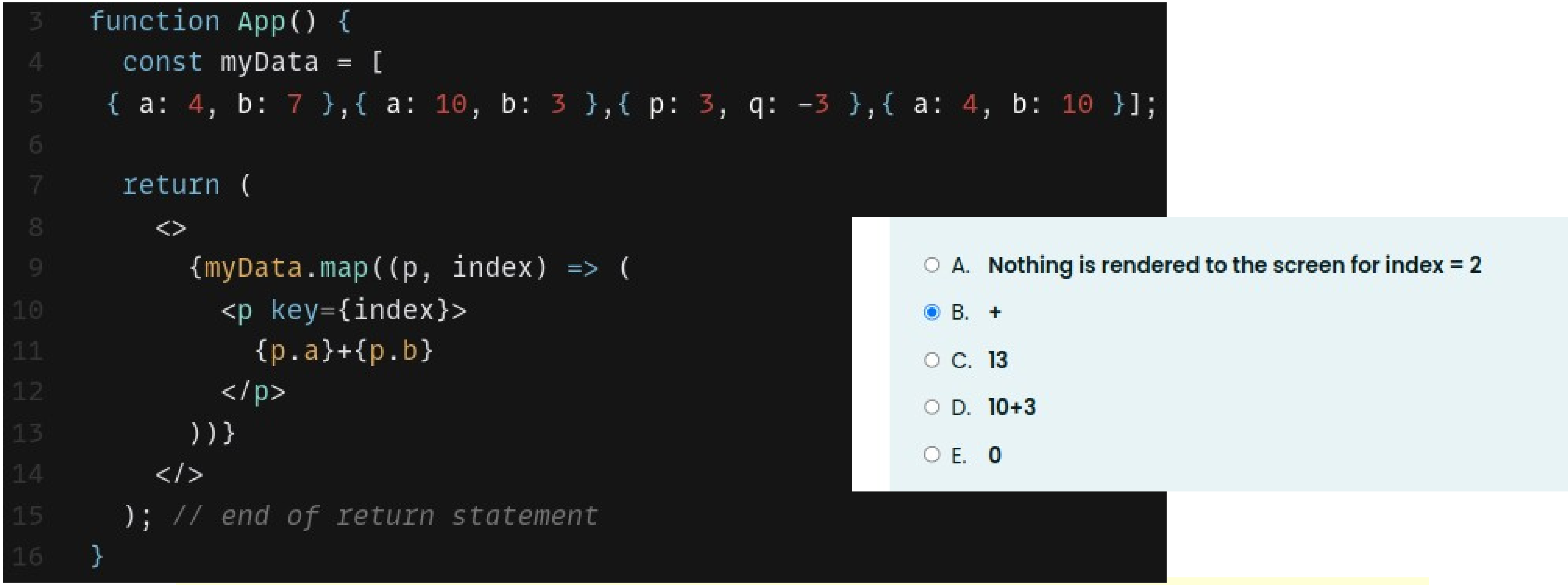
● B. 4,10

○ C. 10,3

○ D. 4,7

- Adding both properties a and b together – then sort in ASCENDING order
- Index = 3 is the last element (highest sum of a and b) where sum is 14

# Topic 8 – Question 5 “trick”



```
3   function App() {
4     const myData = [
5    { a: 4, b: 7 },{ a: 10, b: 3 },{ p: 3, q: -3 },{ a: 4, b: 10 }];
6
7     return (
8       <>
9         {myData.map((p, index) => (
10          <p key={index}>
11            {p.a}+{p.b}
12          </p>
13        ))}
14      </>
15    ); // end of return statement
16  }
```

○ A. Nothing is rendered to the screen for index = 2

● B. +

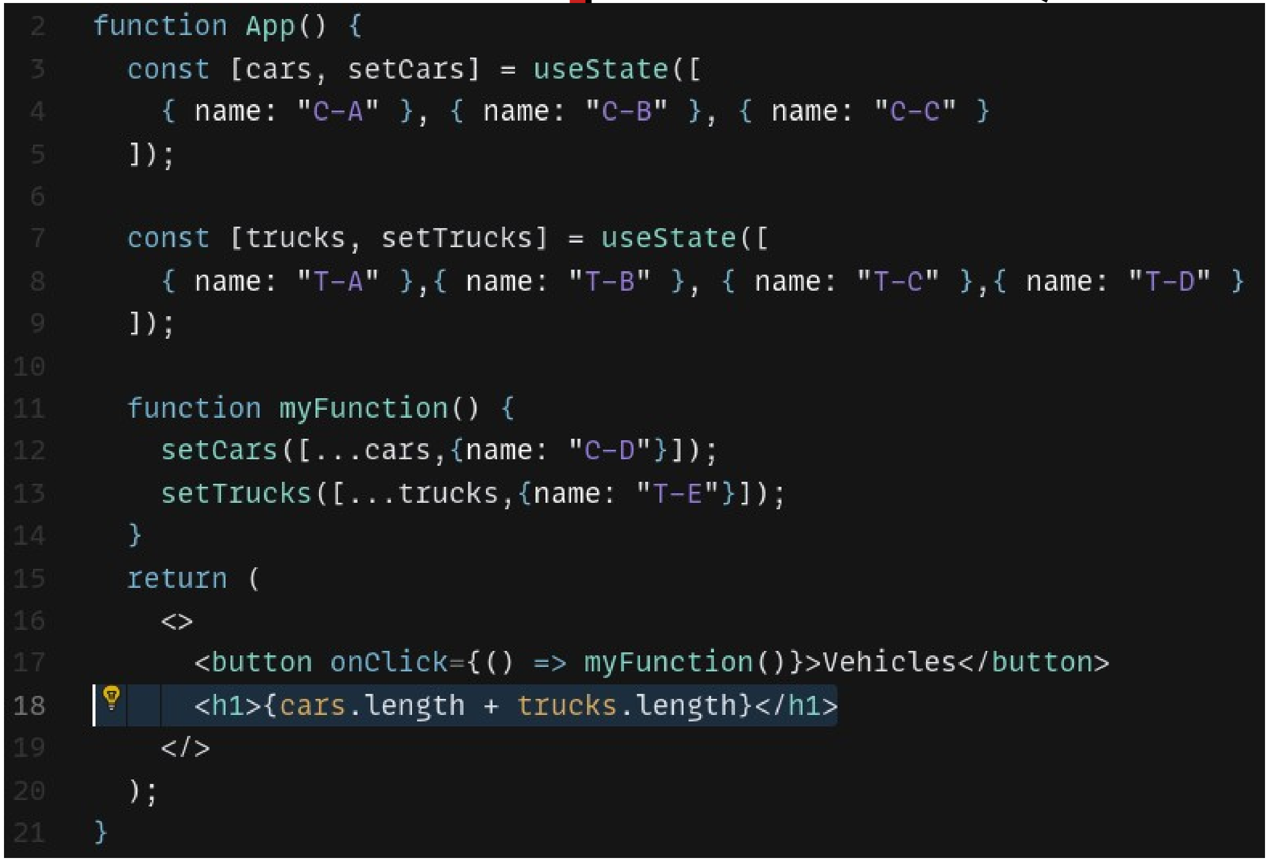
○ C. 13

○ D. 10+3

○ E. 0

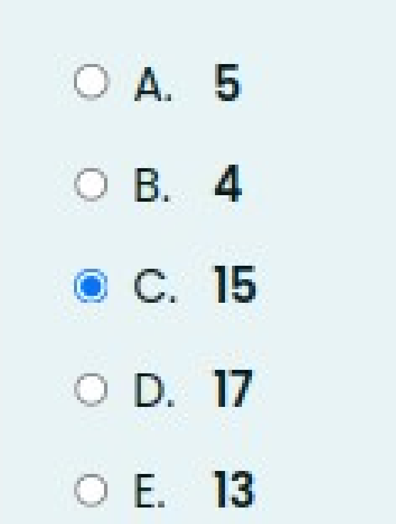
- **Index = 2**, this will be for the element **{p:3,q-3}** – notice the properties are a and b within the line 11
- So react skips the properties but the + sign is rendered as this is a character

# Topic 8 – Question 6



```
2   function App() {
3     const [cars, setCars] = useState([
4       { name: "C-A" }, { name: "C-B" }, { name: "C-C" }
5     ]);
6
7     const [trucks, setTrucks] = useState([
8       { name: "T-A" },{ name: "T-B" }, { name: "T-C" },{ name: "T-D" }
9     ]);
10
11    function myFunction() {
12      setCars([...cars,{name: "C-D"}]);
13      setTrucks([...trucks,{name: "T-E"}]);
14    }
15    return (
16      <>
17        <button onClick={() => myFunction()}>Vehicles</button>
18        <h1>{cars.length + trucks.length}</h1>
19      </>
20    );
21  }
```

- What is rendered to the screen from Line 18 when the button "Vehicles" has been clicked or pressed FOUR times?



○ A. 5

○ B. 4

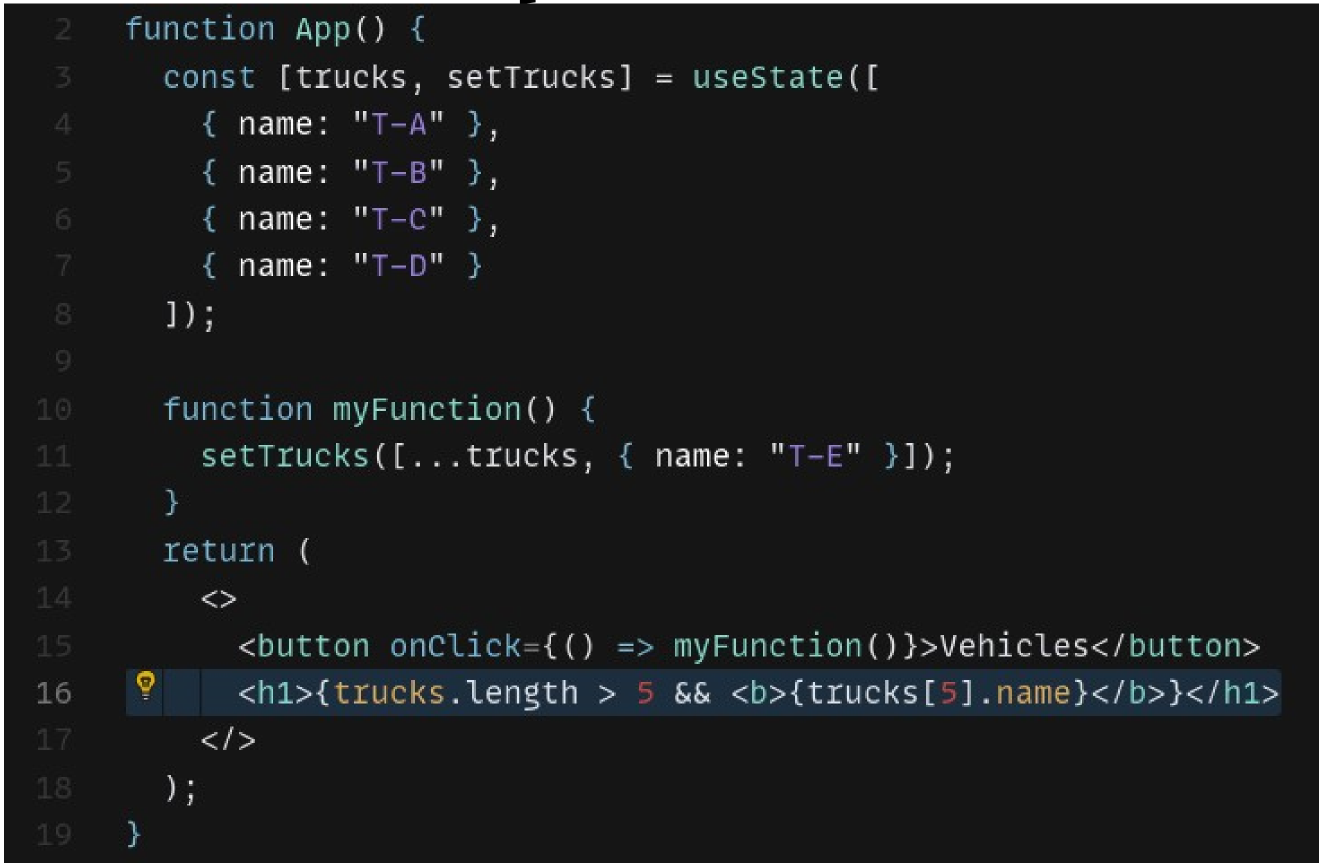
● C. 15

○ D. 17

○ E. 13

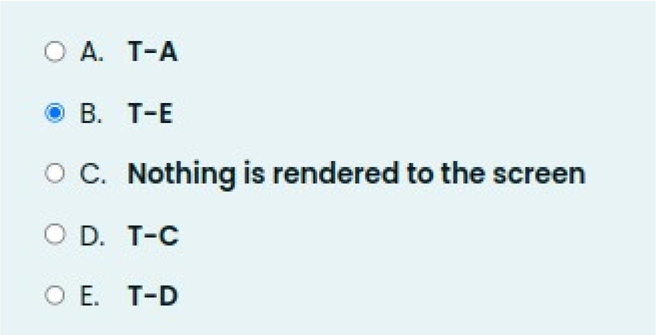
**1st time – cars has length 4, trucks length 5,**
**2nd (5,6)**
**3rd (6,7)**
**4th (7,8) = 15**

# Topic 8 – Question 7



```
 2  function App() {
 3    const [trucks, setTrucks] = useState([
 4      { name: "T-A" },
 5      { name: "T-B" },
 6      { name: "T-C" },
 7      { name: "T-D" }
 8    ]);
 9
10    function myFunction() {
11      setTrucks([...trucks, { name: "T-E" }]);
12    }
13    return (
14      <>
15        <button onClick={() => myFunction()}>Vehicles</button>
16        <h1>{trucks.length > 5 && <b>{trucks[5].name}</b>}</h1>
17      </>
18    );
19  }
```

- What is rendered to the screen from Line 16 within the application below after the Vehicles button has been pressed or clicked 5 times?



○ A. T-A

● B. T-E

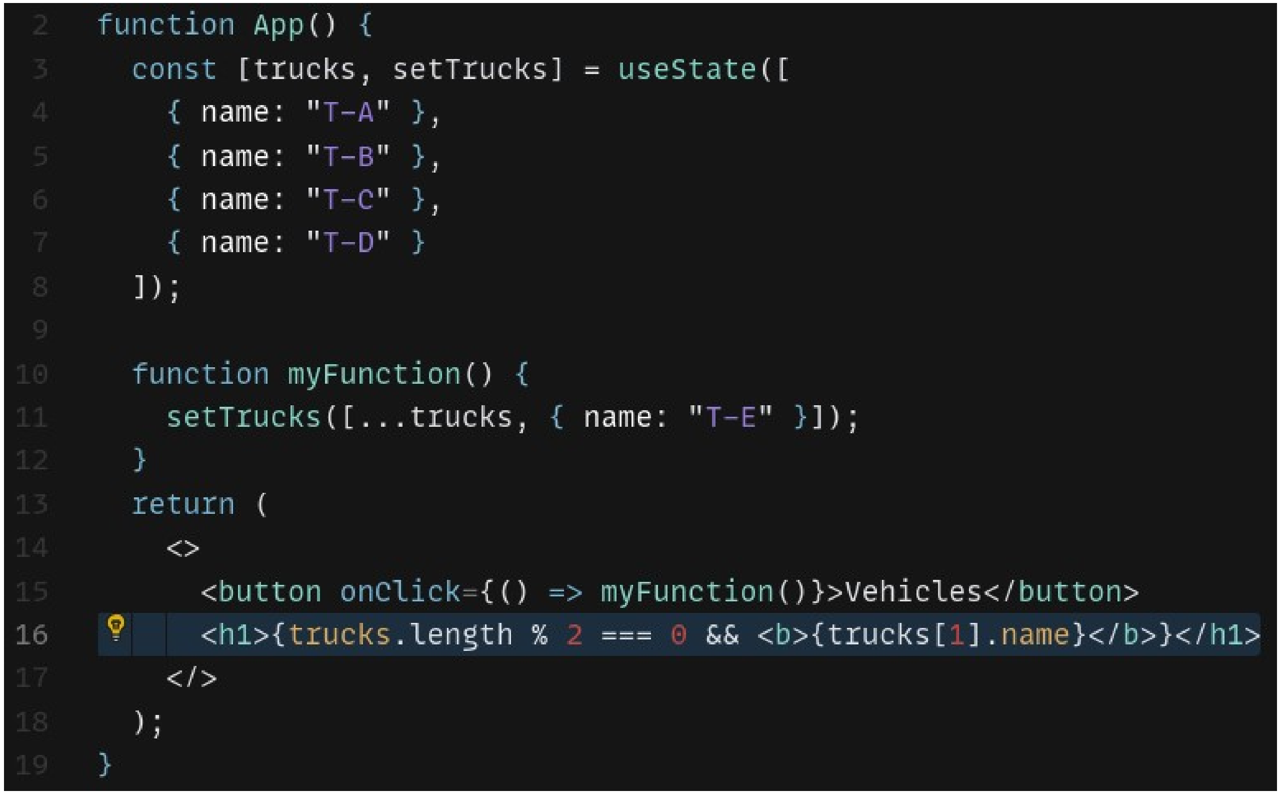
○ C. Nothing is rendered to the screen

○ D. T-C

○ E. T-D

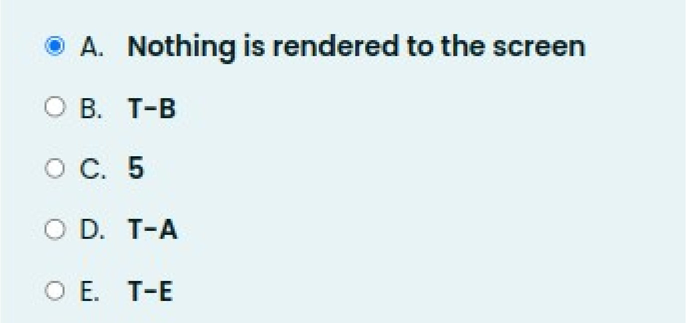
**Originally, this array has length 4 – so in the first setTrucks – the length goes to 5.**
**Then the next one it is 6 – “T-E” is set to the name here.**
**It stays in position six then regardless of the number of clicks following**

# Topic 8 – Question 8 (“tricky”?)



```
 2  function App() {
 3    const [trucks, setTrucks] = useState([
 4      { name: "T-A" },
 5      { name: "T-B" },
 6      { name: "T-C" },
 7      { name: "T-D" }
 8    ]);
 9
10    function myFunction() {
11      setTrucks([...trucks, { name: "T-E" }]);
12    }
13    return (
14      <>
15        <button onClick={() => myFunction()}>Vehicles</button>
16        <h1>{trucks.length % 2 === 0 && <b>{trucks[1].name}</b>}</h1>
17      </>
18    );
19  }
```

- What is rendered to the screen, from Line 16, when the application below runs and the button Vehicles is pressed or clicked THREE Times?



● A. Nothing is rendered to the screen

○ B. T-B

○ C. 5

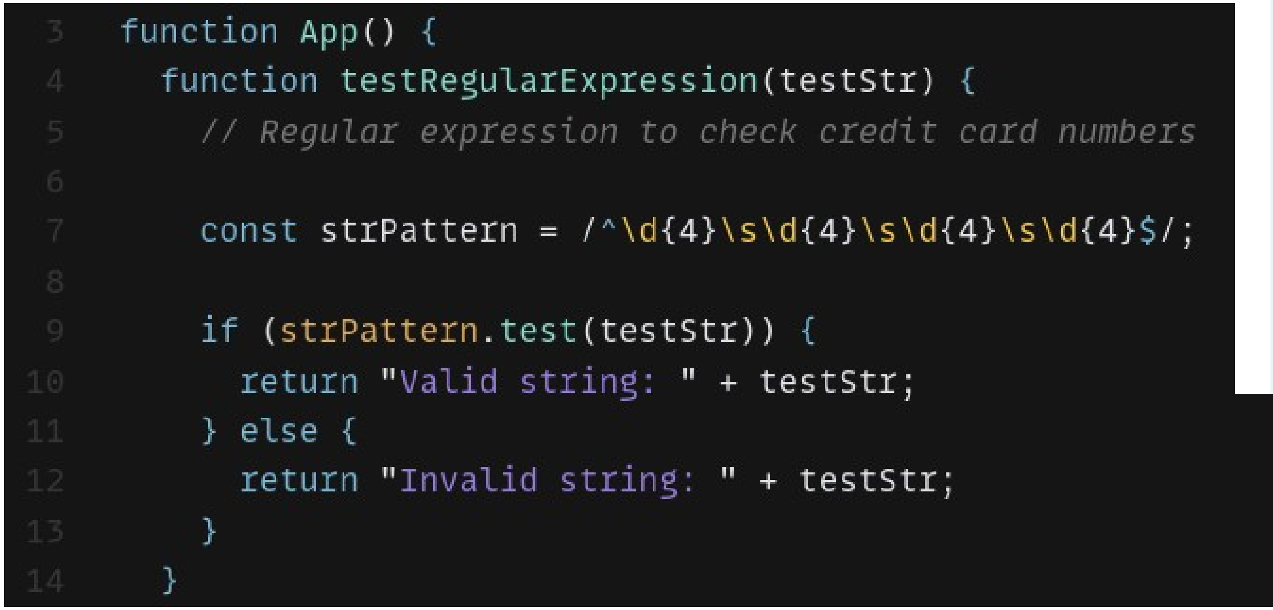
○ D. T-A

○ E. T-E

**Line 16 only executes when the length of the trucks array is even. Then we will always render the 2nd element name.**
**1st click – length = 5**
**2nd click – length = 6**
**3rd click – length = 7**

**So nothing is rendered to the screen**

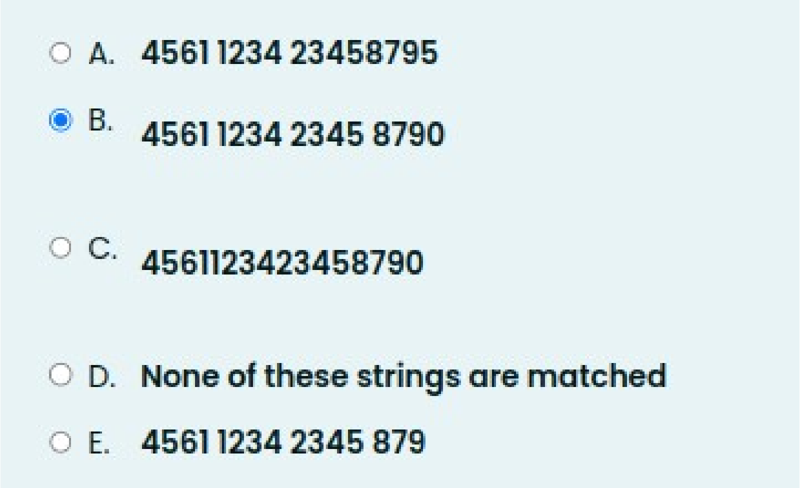
# Topic 8 – Question 9



```
function App() {
  function testRegularExpression(testStr) {
    // Regular expression to check credit card numbers

    const strPattern = /^\d{4}\s\d{4}\s\d{4}\s\d{4}$/;

    if (strPattern.test(testStr)) {
      return "Valid string: " + testStr;
    } else {
      return "Invalid string: " + testStr;
    }
  }
```



○ A. 4561 1234 23458795

● B. 4561 1234 2345 8790

○ C. 4561123423458790

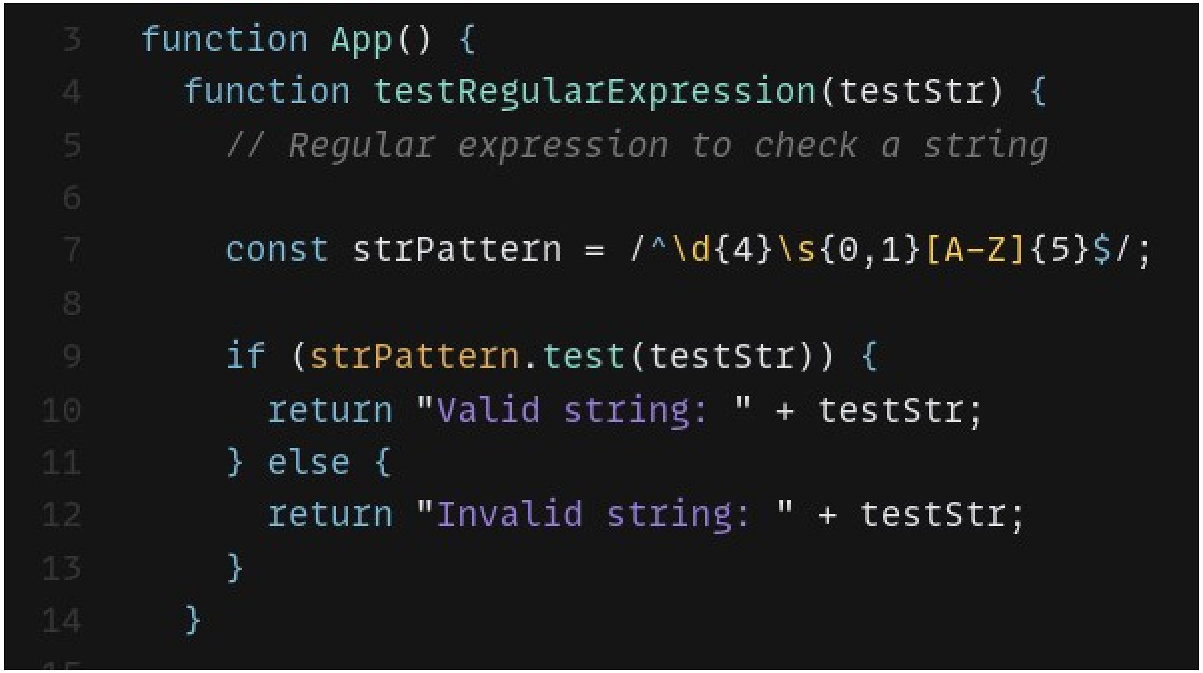
○ D. None of these strings are matched

○ E. 4561 1234 2345 879

**Option B is the only option where there are spaces between the four blocks of digits.**

- Given the following regular expression in Javascript - which ONE of the following strings will return **`"Valid String"`** when the function **`testRegularExpression`** is used below.

# Topic 8 – Question 10

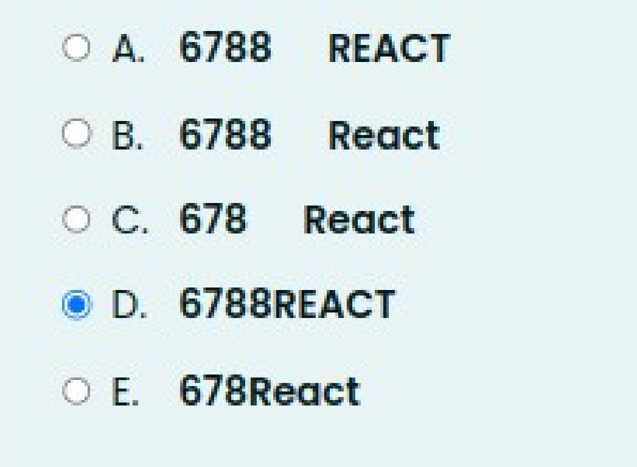


```
function App() {
  function testRegularExpression(testStr) {
    // Regular expression to check a string

    const strPattern = /^\d{4}\s{0,1}[A-Z]{5}$/;

    if (strPattern.test(testStr)) {
      return "Valid string: " + testStr;
    } else {
      return "Invalid string: " + testStr;
    }
  }
```

- Given the following code to check or test a regular expression. Which ONE of the strings below will return **"Valid string"** when the function shown below is used.



○ A. 6788 REACT

○ B. 6788 React

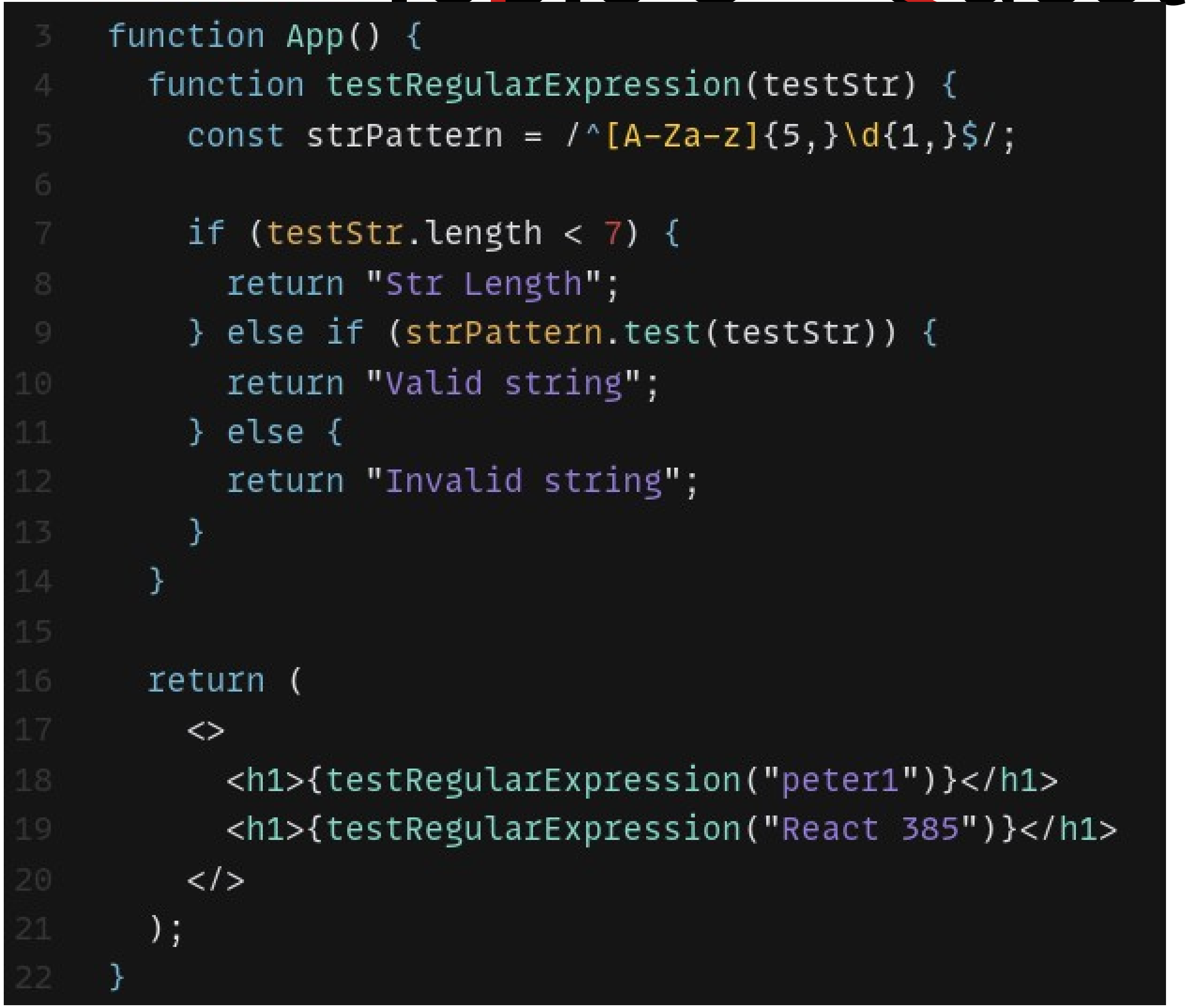
○ C. 678 React

● D. 6788REACT

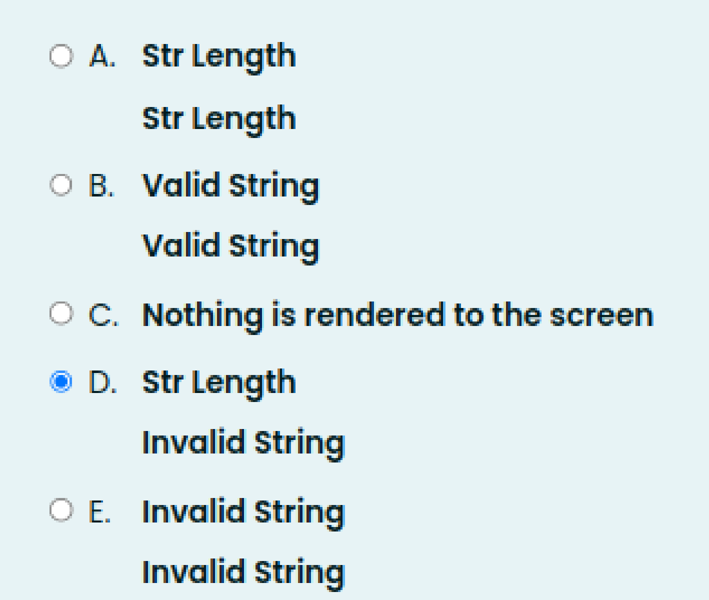
○ E. 678React

**We need 4 digits then zero or one spaces then exactly 5 upper case letters.**
**Option D above is the only string that will fit this regular expression**

# Topic 8 – Question 11



```
3   function App() {
4     function testRegularExpression(testStr) {
5       const strPattern = /^[A-Za-z]{5,}\d{1,}$/;
6
7       if (testStr.length < 7) {
8         return "Str Length";
9       } else if (strPattern.test(testStr)) {
10        return "Valid string";
11      } else {
12        return "Invalid string";
13      }
14    }
15
16    return (
17      <>
18        <h1>{testRegularExpression("peter1")}</h1>
19        <h1>{testRegularExpression("React 385")}</h1>
20      </>
21    );
22  }
```



○ A. Str Length
Str Length

○ B. Valid String
Valid String

○ C. Nothing is rendered to the screen

● D. Str Length
Invalid String

○ E. Invalid String
Invalid String

- What output, from the options below, is rendered to the screen when the code below tests the strings using regular expression and string functions?

**We need 5 or more upper or lower case letters then at least one digit until the end of the string.**
**So peter1 fails on the string length check on line 7**

**And React 385 has a space so it’s invalid**

# Lab Exam 2 – Dec 1st 2023 - 11:00 – 12:00

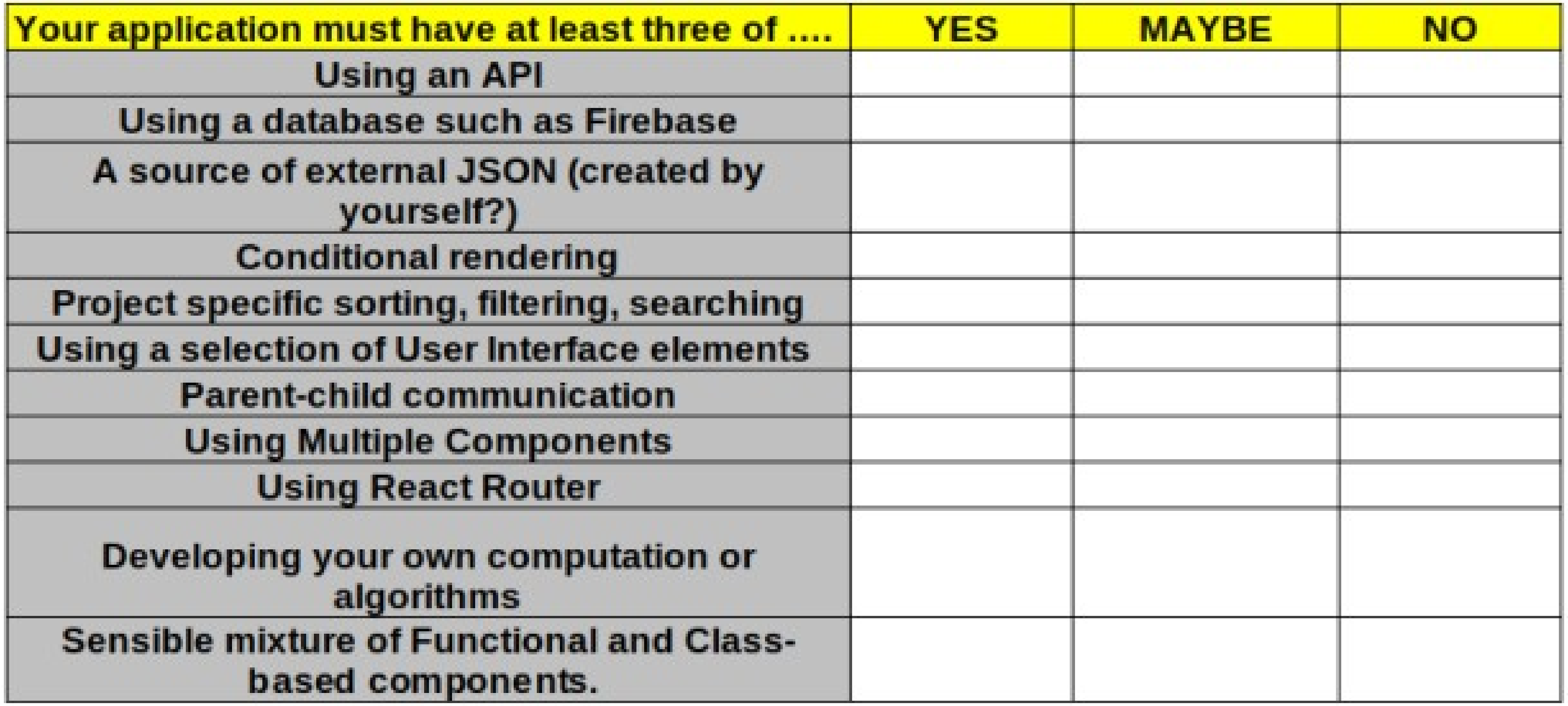
- **DEMO or PRACTICE Lab Exam 2 will be available after November 24th 2023**
- The DEMO Lab Exam 2 is optional – but it is advised that you take this exam in good time before Lab Exam 2
- **Lab Exam 2 will include ALL MATERIAL up to today – Lecture 16 (including that of Lab Exam 1) – it's impossible to separate the material.**

# CS385 Lecture 16

# Lab 7 - 24th November 2023

- **I’ll be meeting all teams** – a demo (even basic) should be available
- **Team attendance is monitored.**
- Your demonstrator mentor will also be meeting with you – make sure to ask questions about your code, project implementation, etc.

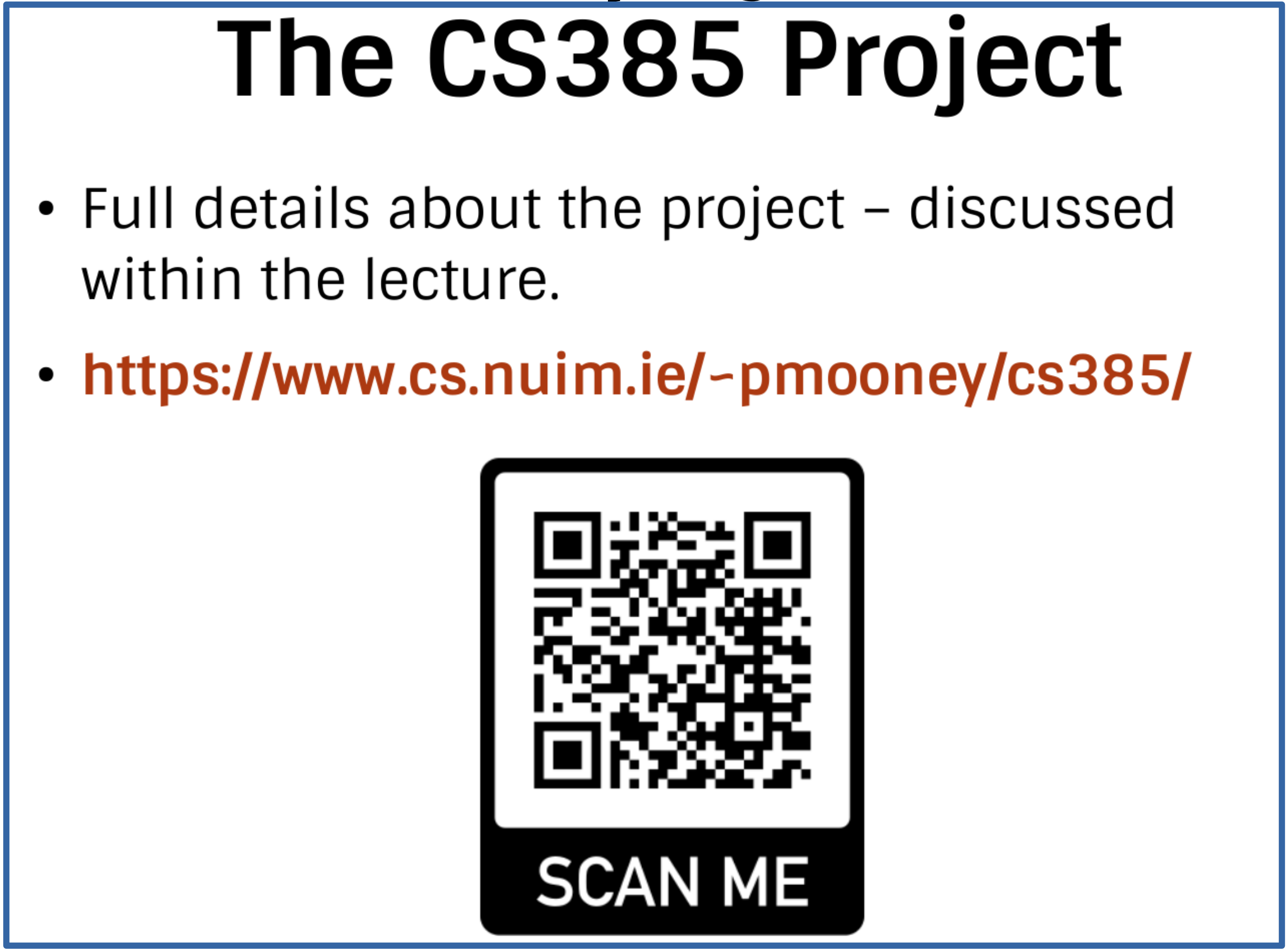
# Start thinking about the functionalities (your project needs at least three….)



| Your application must have at least three of …. | YES | MAYBE | NO |
|---|---|---|---|
| Using an API | | | |
| Using a database such as Firebase | | | |
| A source of external JSON (created by yourself?) | | | |
| Conditional rendering | | | |
| Project specific sorting, filtering, searching | | | |
| Using a selection of User Interface elements | | | |
| Parent-child communication | | | |
| Using Multiple Components | | | |
| Using React Router | | | |
| Developing your own computation or algorithms | | | |
| Sensible mixture of Functional and Class-based components. | | | |

- In next few lectures we'll cover **React Router** and an **optional Firebase lecture** will be provided. **Development of your own algorithms** can include the code necessary to perform validation of forms or other input

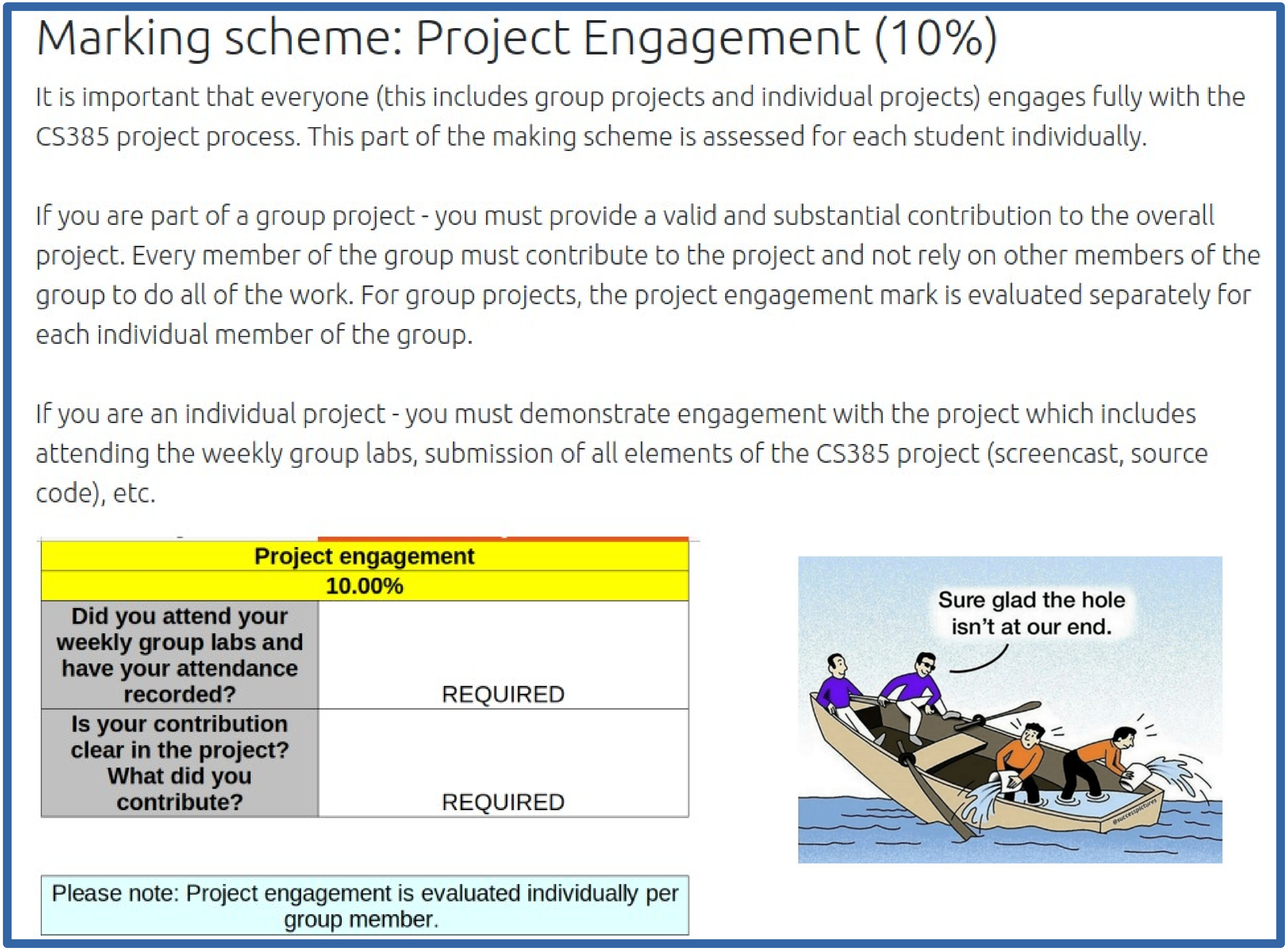
# Before you ask that question……. did you check the project website?



## The CS385 Project

- Full details about the project – discussed within the lecture.
- https://www.cs.nuim.ie/~pmooney/cs385/

# Remember .....



## Marking scheme: Project Engagement (10%)

It is important that everyone (this includes group projects and individual projects) engages fully with the CS385 project process. This part of the making scheme is assessed for each student individually.

If you are part of a group project - you must provide a valid and substantial contribution to the overall project. Every member of the group must contribute to the project and not rely on other members of the group to do all of the work. For group projects, the project engagement mark is evaluated separately for each individual member of the group.

If you are an individual project - you must demonstrate engagement with the project which includes attending the weekly group labs, submission of all elements of the CS385 project (screencast, source code), etc.

| Project engagement | |
|---|---|
| 10.00% | |
| Did you attend your weekly group labs and have your attendance recorded? | REQUIRED |
| Is your contribution clear in the project? What did you contribute? | REQUIRED |

Please note: Project engagement is evaluated individually per group member.

# Lecture 17 and 18
# 28th November 2023

- **Working with dates and timestamps** in React Javascript
- **Removing duplicates from arrays** of objects
- **Summarising arrays** of objects
- **`JSON.Stringify`**
- **An API with parameters** (changes to useEffect)
- Possibly, React router…
- **Discussion of Lab Exam 2**

# Lecture 19 and 20
# 5th December 2023

- **React Router** (if not covered in Lecture 17/18)
- **Overview of Firebase** – a full OPTIONAL video lecture will be provided (with source code)
- **Working with objects** – equality among objects, the need for ===, (Lab Exam 3)
- **Using** `getOwnPropertyNames` in Javascript (Lab Exam 3)
- **Nested map functions** (map functions inside map functions)

# See you all on Friday, in the project Lab