Assignment Two

Please read:

- This is your second assignment which you need to submit on the LMS. This is an assignment on Functional Programming.

- Late Submission: The deadline to submit this assignment is 19th February 2024, 6:00PM IST.

What you have to do?

1. Here is a template of CodeSandbox of where you have to write your solutions - CodeSandbox. Fork this codeSandbox and write your solutions here.
2. There are multiple folders, one for each exercise. You need to write your code for each exercise under the comment line `// Your ES6 code here`
3. DO NOT change or write any code in the `.test.js` files.
4. In this CodeSandbox, tests have been provided. You just need to write your solution and run the tests.
5. You can check whether your solution is correct or not by going to the Tests tab on the right panel of CodeSandbox.
6. Work on the questions given below and be ready with your solutions in CodeSandbox. You have to submit your CodeSandbox link below on this page.

Important Instructions:

1. You can make use of methods such as map(), filter(), reduce(), find(), length, push, toLowerCase(), toUpperCase(), includes() wherever needed.
2. DO NOT make use of for-loops.
3. Do not copy from someone else as that would be cheating. We have already provided you enough practice questions before, so that you get comfortable doing this assignment.
4. Once completed, you should post your assignment solution in your team or pod and get it reviewed by your friends before submitting it on the LMS for final review.

Please note that this assignment should take 3-5 hours of your time at max if you have practiced the previously given practice set questions.

Questions:

These questions are provided just for reference here. These are the same questions as in the CodeSandbox provided above.

1. Write an ES6 function that takes in an array of objects representing sounds and returns an array of sounds name with all the letters capitalized.

```
// Your ES6+ code here

const sounds = [
  {
    name: 'rain',
    sound: 'tap tap tap',
  },
  {
    name: 'fire',
    sound: 'blaze',
  },
  {
    name: 'water',
    sound: 'slosh slosh',
  },
]

console.log(capitalizeLetters(sounds)) // Output: ["RAIN", "FIRE", "WATER"]
```

COPY

2. Write an ES6 function that takes an array of objects with name, price and quantity, and returns a new array with only the names containing `bread`

```
// Your ES6+ code here

const products = [
  { name: 'Bread', price: 480, quantity: 3 },
  { name: 'Clips', price: 200, quantity: 5 },
  { name: 'green Bread Knife', price: 3077, quantity: 1 },
  { name: 'Slipper', price: 150, quantity: 2 },
]

console.log(filterByKeyword(products))
// Output: ["Bread", "green Bread Knife"]
```

COPY

3. Write an ES6 function that takes an array of objects with name, price and quantity, and returns the new array having only names of products who are having a quantity of more than 10 and a name of more than 5 characters in length.

```
// Your ES6+ code here

const products = [
  { name: 'Bread', price: 150, quantity: 20 },
  { name: 'Hoodie', price: 200, quantity: 50 },
  { name: 'Pyjama', price: 307, quantity: 10 },
  { name: 'Slipper', price: 480, quantity: 30 },
]

console.log(filterProducts(products)) // Output: ["Hoodie", "Slipper"]
```

COPY

4. Write an ES6 function that takes an array of objects with name, salary and role, and returns a new array with names whose salary is an even number.

```
// your ES6+ code here

const employees = [
  { name: 'Raju', salary: 154, role: 'dev' },
  { name: 'Aakash', salary: 200, role: 'dev' },
  { name: 'Ramesh', salary: 3077, role: 'pm' },
  { name: 'Jiten', salary: 487, role: 'qa' },
]

console.log(filterBySalary(employees)) // Output: ["Raju", "Aakash"]
```

COPY

5. Write an ES6 function that takes in an array of numbers and returns the sum of all the even numbers. Use the nullish coalescing operator to handle undefined or null values.

```
// Your ES6+ code here

console.log(sumOfEvenNumbers([1, 2, 3, 4, 5, 6])) // 12
console.log(sumOfEvenNumbers([1, 3, 5])) // 0
```

COPY

6. Write an ES6 function that takes an array of objects with name, salary, role and workExperience, and returns a new array with an incremented salary of 1000. Also add a property `isSenior` as true for employees who have been working for more than equal to 3 years.

```
// Your ES6+ code here

const employees = [
  { name: 'Raju', salary: 1500, role: 'dev', workExperience: 3 },
  { name: 'Aakash', salary: 2000, role: 'dev', workExperience: 3 },
  { name: 'Dinesh', salary: 2000, role: 'dev', workExperience: 2.5 },
  { name: 'Mohan', salary: 2000, role: 'dev', workExperience: 2 },
  { name: 'Ramesh', salary: 3070, role: 'pm', workExperience: 3 },
  { name: 'Jiten', salary: 4800, role: 'qa', workExperience: 3 },
]

console.log(incrementSalaryAndPromoteSenior(employees))
// Output: [
//   { name: "Raju", salary: 2500, role: "dev", workExperience: 3, isSenior: true },
//   { name: "Aakash", salary: 3000, role: "dev", workExperience: 3, isSenior: true },
//   { name: "Dinesh", salary: 2000, role: "dev", workExperience: 2.5 },
//   { name: "Mohan", salary: 2000, role: "dev", workExperience: 2 },
//   { name: "Ramesh", salary: 4070, role: "pm", workExperience: 3, isSenior: true },
//   { name: "Jiten", salary: 5800, role: "qa", workExperience: 3, isSenior: true }
// ]
```

COPY

7. Write an ES6 function that takes an object of two arrays of objects with name salary, role and workExperience, and returns a new array of employees having a role as QA in all teams

```
// Your ES6+ code here

const teams = {
  team1: [
    { name: 'Alice', salary: 50000, role: 'engineer', workExperience: 2 },
    { name: 'Bob', salary: 75000, role: 'QA', workExperience: 5 },
    { name: 'Charlie', salary: 60000, role: 'developer', workExperience: 3 },
  ],
  team2: [
    { name: 'David', salary: 55000, role: 'engineer', workExperience: 3 },
    { name: 'Eve', salary: 80000, role: 'QA', workExperience: 4 },
    { name: 'Frank', salary: 65000, role: 'developer', workExperience: 2 },
  ],
}

const qaEmployees = getQAEmployees(teams)

console.log(qaEmployees)
// Output:
// [
//   { name: "Bob", salary: 75000, role: "QA", workExperience: 5 },
//   { name: "Eve", salary: 80000, role: "QA", workExperience: 4 },
// ]
```

COPY

8. Write an ES6 function that takes an array of objects representing books with properties title, author and pageCount. Return the first book object that has more than 500 pages.

```
const books = [
  {
    title: 'The Lord of the Rings',
    author: 'J.R.R. Tolkien',
    pageCount: 1178,
  },
  { title: 'To Kill a Mockingbird', author: 'Harper Lee', pageCount: 281 },
  {
    title: "The Hitchhiker's Guide to the Galaxy",
    author: 'Douglas Adams',
    pageCount: 193,
  },
  {
    title: 'The Name of the Wind',
    author: 'Patrick Rothfuss',
    pageCount: 662,
  },
]

// Your code here

const bookWithMoreThan500Pages = findBookWithMoreThan500Pages(books)
console.log(bookWithMoreThan500Pages)
// Output: { title: "The Lord of the Rings", author: "J.R.R. Tolkien", pageCount: 1178 }
```

COPY

9. Write an ES6 function that takes an array of objects with name, role, and array of hours which represents the time the person spends on the server each day. Find the person who is the most active in the community and spends most of the time in it.

```
// your ES6+ code here

const neogCommunity = [
  { name: 'Raju', role: 'student', hours: [1, 2, 3, 1, 2, 3, 0] },
  { name: 'Aakash', role: 'mentor', hours: [1, 2, 3, 1, 2, 3, 0] },
  { name: 'Ramesh', role: 'student', hours: [1, 2, 3, 1, 2, 3, 3] },
  { name: 'Jiten', role: 'TA', hours: [2, 2, 3, 5, 2, 3, 0] },
  { name: 'Harsh', role: 'student', hours: [1, 7, 3, 2, 2, 3, 0] },
  { name: 'Akshay', role: 'student', hours: [1, 6, 3, 1, 2, 3, 0] },
  { name: 'Rohan', role: 'mentor', hours: [1, 2, 3, 12, 2, 3, 0] },
  { name: 'Mohan', role: 'student', hours: [1, 8, 3, 0, 2, 3, 0] },
]

console.log(mostActivePerson(neogCommunity))
// Output: { name: "Rohan", role: "mentor", hours: [1, 2, 3, 12, 2, 3, 0] }
```

COPY

10. Write an ES6 function that takes an array of objects with name, role and array of hours which represents the time a person spends on the server each day. Give the tag of `regular` to students who are active in the community and spends more than 20 hours a week.

```
// your ES6+ code here

const neogCommunity = [
  { name: 'Raju', role: 'student', hours: [1, 2, 3, 1, 2, 3, 0] },
  { name: 'Aakash', role: 'mentor', hours: [1, 2, 3, 4, 5, 6, 7] },
  { name: 'Ramesh', role: 'student', hours: [4, 5, 6, 4, 5, 6, 0] },
  { name: 'Jiten', role: 'TA', hours: [2, 2, 3, 5, 2, 3, 0] },
  { name: 'Harsh', role: 'student', hours: [7, 8, 9, 7, 8, 9, 0] },
  { name: 'Akshay', role: 'student', hours: [1, 3, 5, 7, 9, 0, 2] },
  { name: 'Rohan', role: 'mentor', hours: [1, 2, 3, 12, 2, 3, 0] },
  { name: 'Mohan', role: 'student', hours: [4, 6, 8, 0, 1, 9, 2] },
]

console.log(tagRegularStudents(neogCommunity))
// Output: [
//   { name: "Raju", role: "student", hours: [1, 2, 3, 1, 2, 3, 0] },
//   { name: "Aakash", role: "mentor", hours: [1, 2, 3, 4, 5, 6, 7] },
//   { name: "Ramesh", role: "student", hours: [4, 5, 6, 4, 5, 6, 0], tag: "regular" },
//   { name: "Jiten", role: "TA", hours: [2, 2, 3, 5, 2, 3, 0] },
//   { name: "Harsh", role: "student", hours: [7, 8, 9, 7, 8, 9, 0] tag: "regular" },
//   { name: "Akshay", role: "student", hours: [1, 3, 5, 7, 9, 0, 2] tag: "regular" },
//   { name: "Rohan", role: "mentor", hours: [1, 2, 3, 12, 2, 3, 0] },
//   { name: "Mohan", role: "student", hours: [4, 6, 8, 0, 1, 9, 2] tag: "regular" }
// ]
```

COPY

All the best. We hope you complete and submit your assignment in time. Please note that we are here to help you, if needed.

Click on the Share button on your CodeSandbox, then click on Copy link button and submit that link here in the submission form below. Make sure the access is public.