**** All following exams please using Javascript only ****

1.

```
/**
There is an array, each item has such format:
{firstName: 'xxx', lastName: 'xxx', customerID: 'xxx', note: 'xxx',
profession: 'xxx'}
lastName, note can be empty, customerID can only be a set of digital
numbers.
profession can only have 'student', 'freelancer', 'productOwner',
'engineer' or 'systemAnalytics'.
**/

/**
Q1. Please follow the principle ('firstName' + 'lastName' + 'customerID')
to sort this array and print it out.
**/

function sortUserName(user) {
    return users.sort((a, b) => {
        const aName = `${a.firstName}${a.lastName}${a.customerID}`;
        const bName = `${b.firstName}${b.lastName}${b.customerID}`;
        return aName.localeCompare(bName);
    });
}



/**
Q2. Please sort by 'profession' to follow the principle.
('systemAnalytics' > 'engineer' > 'productOwner' > 'freelancer' >
'student'')
**/

function sortByType(user) {
    const professionOrder = {
        'systemAnalytics': 5,
        'engineer': 4,
        'productOwner': 3,
        'freelancer': 2,
        'student': 1
    };

    return users.sort((a, b) => {
        return professionOrder[b.profession] -
        professionOrder[a.profession];
    });
}
```

2.

```
/** HTML
<div class="container">
  <div class="header">5/8 外出確認表</div>
     <div class="content">
       <ol class="shop-list">
     <li class="item">麵包</li>
     <li class="item">短袖衣服</li>
     <li class="item">飲用水</li>
     <li class="item">帳篷</li>
       </ol>
       <ul class="shop-list">
     <li class="item">暈車藥</li>
     <li class="item">感冒藥</li>
     <li class="item">丹木斯</li>
     <li class="item">咳嗽糖漿</li>
       </ul>
     </div>
  <div class="footer">以上僅共參考</div>
   </div>
**/

/** CSS
.container {
  font-size: 14px;
}
.container .header
  {font-size: 18px;
}
.container .shop-list
   {list-style: none;
   margin-left: -15px;
}
.container .shop-list li.item
   {color: green;
}
.container .shop-list .item {
  /* Explain why does this color not works, and how to fix make it work on
1st list */
   color: blue;
}
```

.container .shop-list .item 選擇器的優先級低於 .container .shop-list li.item，因此藍色被覆蓋。解決方案是提高li.item選擇器的優先級。

```
 /* Write styling make every other line give background color to next one */
```

將奇數行給予背景顏色

```
.container .shop-list li.item:nth-child(odd) {
  background-color: #f2f2f2;
}
**/
```

3.

```
/**

let items = [1, 1, 1, 5, 2, 3, 4, 3, 3, 3, 3, 3, 3, 7, 8, 5, 4, 9, 0, 1,
3, 2, 6, 7, 5, 4, 4, 7, 8, 8, 0, 1, 2, 3, 1];
Please write down a function to console log unique value from this array.

**/

function getUniqueNumber (items) {
     return [...new Set(items)];
}

let items = [1, 1, 1, 5, 2, 3, 4, 3, 3, 3, 3, 3, 3, 7, 8, 5, 4, 9, 0, 1, 3,
2, 6, 7, 5, 4, 4, 7, 8, 8, 0, 1, 2, 3, 1];
console.log(getUniqueNumber(items));
```

4.

/ Can you explain about Interface and Enum, and where will you be using, please make some examples. **/**

Interface（介面）是 TypeScript 中一個非常重要的概念，用於定義物件的結構，它描述了物件應該擁有的屬性和方法

```
interface Person {
  name: string;
  greet(): void;
}

// 使用 Person Interface 創建一個物件
const person: Person = {
  name: 'Sam',
  greet() {
    console.log(`Hello, my name is ${this.name}.`);
  }
};

person.greet(); // Hello, my name is Sam.
```

Enum（列舉）用於定義一組命名的常量，常用在有一組固定的、相關的值時

```
enum DayOfWeek {
  Monday,
  Tuesday,
  Wednesday,
  Thursday,
  Friday,
  Saturday,
  Sunday
}
```

4.

```
console.log(DayOfWeek.Monday); // 0
```

5.

/ Can you explain the problem with the following code, and how to fix it. **/**

```
class Count extends React.Component
  {constructor(props) {
    super(props);
    this.state = { count: 0 };
    this.handleAddCount = this.handleAddCount.bind(this);
  }

  handleAddCount(){
    this.setState(prevState => ({
      count: prevState.count + 1
    }));
    this.setState(prevState => ({
      count: prevState.count + 1
    }));
    this.setState(prevState => ({
      count: prevState.count + 1
    }));
  }

  render()
    {return
    (
      <div>
```

```
          <h2>{this.state.count}</h2>
          <button onClick={this.handleAddCount}>Add</button>
        </div>
      );
    }
  }

  ReactDOM.render(
    <Count />,
    document.getElementById('root')
  );
```

6.

/ Please write the sample code to debounce handleOnChange **/**

```
var SearchBox = React.createClass({
      render: function() {
            return <input type="search" name="p"
            onChange={this.handleOnChange} />;
      },
      handleOnChange: debounce(function(event) {
            // make ajax call
      }, 300)
});

function debounce(func, delay) {
  let timeoutId;
  return function(...args) {
    clearTimeout(timeoutId);
    timeoutId = setTimeout(() => {
      func.apply(this, args);
    }, delay);
  };
}
```