**JS Advanced Exam**

**Problem 3. Unit Testing(94/100)**

**Your Task**

Using **Mocha** and **Chai** write **JS Unit Tests** to test a variable named **chooseYourCar**, which represents an object. You may use the following code as a template:

|  |
| --- |
| describe(**"*Tests* …"**, **function**() {  describe(**"*TODO* …"**, **function**() {  ***it***(**"*TODO …*"**, **function**() {  *//* ***TODO:*** …  });  });  *//* ***TODO:*** …  }); |

The object that should have the following functionality:

* **choosingType (type, color,** **year) -** A function that accepts **three** parameters: **string**, **string**, and **number**.
* If the **year** is **less** than 1900 and the **year** is **more** than 2022, **throw** an error: **"Invalid Year!"**
* If the value of the string **type** is different from "**Sedan**", **throw** an error: **"This type of car is not what you are looking for."**
* To be picked, the **car** must meet the **following requirement**:
  + If the **year** of the car is **greater** or **equal** to **2010**, **return** the string:

**"This ${color} ${type} meets the requirements, that you have."**

* Otherwise, if the above conditions are **not** met, **return** the following message:

**"This ${type} is too old for you, especially with that ${color} color."**

* There is **no** need for **validation** for the **input**, you will always be given two strings, and number.
* **brandName** **(brands, brandIndex) -** A function that accepts an **array** and **number**. The **brands** array will store the brand names (["**BMW**", "**Toyota**", "**Peugeot**"…]).
  + You must **remove** an **element** (brand) from the **array** that is located on the **index** specified as a parameter.
  + Finally, **return** the changed array of brands as a string, joined by a comma and a space.
  + There is a **need for validation** for the input, an **array** and index may not always be valid. In case of submitted **invalid** parameters, **throw** an error **"Invalid Information!"**:
    - If passed **brands** parameteris **not** an array.
    - If the **index** is not a number and is outside the limits of the array.
* **CarFuelConsumption (distanceInKilometers, consumptedFuelInLitres) -** A function that accepts two parameters: **number, number**.
* You test drive the car to find out what its consumption is.
* You need to **calculate** liters per 100 kilometers consumption by **dividing** the fuel consumption by 100 and then **multiply** by distance.
  + - **The result must be formatted to the second digit after the decimal point.**
* If the liters/100km is **less** or **equal** to 7L. **return** the following message:

**"The car is efficient enough, it burns ${litersPerHundredKm} liters/100 km."**

* Else, **return** the following message:

**"The car burns too much fuel - ${litersPerHundredKm} liters!"**

* You **need to validate** the input, if the **distanceInKilometers** and **consumptedFuelInLitres** are not a **numbers**, or are a **negative** numbers, **throw** an error: **"Invalid Information!"**.

**JS Code**

To ease you in the process, you are provided with an implementation that meets all of the specification requirements for the **chooseYourCar** object:

|  |
| --- |
| chooseYourCar.js |
| const chooseYourCar = {      choosingType(type, color, year) {          if (year < 1900 || year > 2022) {              throw new Error(`Invalid Year!`);          } else {              if (type == "Sedan") {                  if (year >= 2010) {                      return `This ${color} ${type} meets the requirements, that you have.`;                  } else {                      return `This ${type} is too old for you, especially with that ${color} color.`;                  }              }              throw new Error(`This type of car is not what you are looking for.`);          }      },    brandName(brands, brandIndex) {          let result = [];          if (!Array.isArray(brands) || !Number.isInteger(brandIndex) || brandIndex < 0 || brandIndex >= brands.length) {              throw new Error("Invalid Information!");          }          for (let i = 0; i < brands.length; i++) {              if (i !== brandIndex) {                  result.push(brands[i]);              }          }          return result.join(", ");      },      carFuelConsumption(distanceInKilometers, consumptedFuelInLiters) {         let litersPerHundredKm =((consumptedFuelInLiters / distanceInKilometers)\* 100).toFixed(2);          if (typeof distanceInKilometers !== "number" || distanceInKilometers <= 0 ||              typeof consumptedFuelInLiters !== "number" || consumptedFuelInLiters <= 0) {              throw new Error("Invalid Information!");          } else if (litersPerHundredKm <= 7) {              return `The car is efficient enough, it burns ${litersPerHundredKm} liters/100 km.`;          } else {              return `The car burns too much fuel - ${litersPerHundredKm} liters!`;          }      }  } |

**Submission**

Submit your tests inside a **describe()** statement, as shown above.