## JS Advanced Regular Exam – 18 Feb 2023

**Problem 3. Unit Testing**

**Your Task**

Using **Mocha** and **Chai** write **JS Unit Tests** to test a variable named **findNewApartment**, 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:

* **isGoodLocation (city, nearPublicTransportation) -** A function that accepts two parameters: **string** and **boolean**.
* If the value of the string **city** is different than a "**Sofia**", "**Plovdiv**" or "**Varna**"

return :

**"This location is not suitable for you."**

* If the value of the boolean **nearPublicTransportation** is **false**

return :

**"There is no public transport in area."**

* Otherwise, if the above conditions are not met, **return** the following message:
* **"You can go on home tour!"**
* You need to validate the input, if the **city** and **nearPublicTransportation** are not a **string** and **boolean**, **throw** an error: "**Invalid input!**".
* **isLargeEnough (apartments, minimalSquareMeters) -** A function that accepts an **array** and **number**.
  + The **apartments** array will store the area of the apartment in square meters ([40, 50, 60…])
  + You must **add** the area of apartment in **resultArr** if is **equal** or **bigger** than **minimalSquareMeters**.
* Finally, **return** the changed array of apartments.
  + There is a need for validation for the input, an **array** and **number** may not always be valid. In case of submitted **invalid** parameters, **throw** an error "**Invalid input!**":
    - If passed **apartments** parameteris not an array.
    - If **apartments** is empty **array.**
    - If the **minimalSquareMeters** is not a number.
* **isItAffordable (price, budget) -** A function that accepts two parameters: **number** and **number**.
* You need to **calculate** if you can afford buying the apartment by **subtracting** the **price** of the apartment from your **budget**.
* If the **result** is lower than **0,** return**:**

**"You don't have enough money for this house!"**

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

**"You can afford this home!"**

* You need to validate the input, if the **price** and **budget** are not a **number** and **price** and **budget** are **less** or **equal** to0**, throw** an error: "**Invalid input!**".

**JS Code**

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

|  |
| --- |
| findApartment.js |
| const findNewApartment = {    isGoodLocation(city, nearPublicTransportation) {      if (typeof city !== "string" || typeof nearPublicTransportation !== "boolean"){          throw new Error("Invalid input!");      }      if (city !== "Sofia" && city !== "Plovdiv" && city !== "Varna") {          return "This location is not suitable for you.";      }else {          if (nearPublicTransportation == true) {              return "You can go on home tour!";          }          else {              return "There is no public transport in area.";          }      }    },    isLargeEnough(apartments, minimalSquareMeters) {      let resultArr = [];      if (!Array.isArray(apartments) || typeof minimalSquareMeters !== "number" || apartments.length == 0) {        throw new Error("Invalid input!");      }      apartments.map((apartment) => {        if (apartment >= minimalSquareMeters) {          resultArr.push(apartment);        }      });      return resultArr.join(', ');    },    isItAffordable(price, budget) {      if (typeof price !== "number" || typeof budget !== "number"       || price <= 0 || budget <= 0) {        throw new Error("Invalid input!");      }      let result = budget - price;      if (result < 0) {        return "You don't have enough money for this house!";      } else {        return "You can afford this home!";      }    },  }; |

**Submission**

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