**03.Home Gardener**

**Your Task**

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

* **plantCareInstructions(plantType) -** A function that accepts **one** parameter: **string**.
* If the **plantType** is **"succulent"** **return** the string:

**"Succulents require minimal watering, indirect sunlight, and well-draining soil."**

* If the **plantType** is **"vegetable"** **return** the string:

**"Vegetables need full sun, regular watering, and nutrient-rich soil."**

* If the **plantType** is **"flowering"** **return** the string:

**"Flowering plants require moderate watering, occasional fertilization, and pruning."**

* If the **plantType** is **"tree"** **return** the string:

**"Trees need deep watering, proper spacing, and regular mulching."**

* If the value of the string **plantType** is different from " **succulent, vegetable, flowering, tree** ", **throw** an error:

**"Invalid plant type!"**

* **availablePlants(plantSizes, maxHeight) -** A function that accepts an **array** and **number**. The **plantSizes** is array of numeric values representing the sizes of plants (in centimeters), you need to check every element in the array and if its **less** or **equal** to **maxHeight**.
  + Count the number of plants that meet these criteria and **return** a string in the following format:

**"There are ${suitablePlants.length} plants suitable for your garden height criteria!"**

* + 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 Information!"**
    - If passed **plantSizes** or **maxHeight** parameterare not an **array** and **number**.
    - If **plantSizes** is empty array or **maxHeight** is less than **1**.
* **gardenExpenses (tools, seeds, discount) -** A function that accepts three parameters: **array, array** and **boolean**.
* A method that calculates the **total cost** of **gardening** **tools** and **seeds**, optionally applying a **discount**.

**Note**: The result must be formatted to the second digit after the decimal point.

* **tools**: an array containing tool names. Available options:
  + - **shovel,** which costs **$25**
    - **rake,** which costs **$15**
    - **watering can,** which costs **$10**
* **seeds**: an array containing seed types. Available options:
  + - * **vegetable seeds**, which costs **$5**
      * **flower seeds**, which costs **$8**
      * **herb seeds**, which costs **$3**
* If the **discount** is **true,** **10%** discountshouldbeapplied**.** Then **return** the following message:

**"You spent $${totalCost} on tools and seeds with a 10% discount!"**

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

**"You spent $${totalCost} on tools and seeds!"**

* You need to validate the input, if the **tools, seeds** and **discount** are not a **array, array** **and** **boolean** 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 **homeGardener** object:

|  |
| --- |
| homeGardener.js |
| const homeGardener = {      plantCareInstructions(plantType) {        if (plantType === "succulent") {          return "Succulents require minimal watering, indirect sunlight, and well-draining soil.";        } else if (plantType === "vegetable") {          return "Vegetables need full sun, regular watering, and nutrient-rich soil.";        } else if (plantType === "flowering") {          return "Flowering plants require moderate watering, occasional fertilization, and pruning.";        } else if (plantType === "tree") {          return "Trees need deep watering, proper spacing, and regular mulching.";        } else {          throw new Error("Invalid plant type!");        }      },        availablePlants(plantSizes, maxHeight) {        if (!Array.isArray(plantSizes) || typeof maxHeight !== "number" || plantSizes.length < 1 || maxHeight < 1) {          throw new Error("Invalid Information!");        }        let suitablePlants = plantSizes.filter(size => size <= maxHeight && size > 0);        return `There are ${suitablePlants.length} plants suitable for your garden height criteria!`;      },        gardenExpenses(tools, seeds, discount) {        if (          !Array.isArray(tools) ||          !Array.isArray(seeds) ||          typeof discount !== "boolean"        ) {          throw new Error("Invalid Information!");        }          let totalCost = 0;          tools.forEach(tool => {          if (tool === "shovel") {            totalCost += 25;          } else if (tool === "rake") {            totalCost += 15;          } else if (tool === "watering can") {            totalCost += 10;          }        });          seeds.forEach(seed => {          if (seed === "vegetable seeds") {            totalCost += 5;          } else if (seed === "flower seeds") {            totalCost += 8;          } else if (seed === "herb seeds") {            totalCost += 3;          }        });          if (discount) {          totalCost \*= 0.9;          return `You spent $${totalCost.toFixed(2)} on tools and seeds with a 10% discount!`;        } else {          return `You spent $${totalCost.toFixed(2)} on tools and seeds!`;        }      }    }; |

**Submission**

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