

JS Advanced Exam

Problem 3. Unit Testing

Your Task

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

- **buyLotteryTicket (ticketPrice, ticketCount, buy)** - A function that accepts **three** parameters: **number, number, and boolean**.
 - There is a **need for validation** for the input, in case of submitted **invalid** parameters, **throw** an error **"Invalid input!"**
 - If the value of the boolean **buy** is **false**, **throw** an error:
"Unable to buy lottery ticket!"
 - To be valid, the **ticket purchase** must meet the **following requirement**:
 - If the **ticketPrice** is **greater** than **0**, and **ticketCount** is **greater or equal** to **1**, and the type of **ticketPrice** and **ticketCount** is **number**, **return** the string:
"You bought \${ticketCount} tickets for \${totalPrice}\$." ,
where **totalPrice** is **ticketPrice** multiplied by **ticketCount**.
- **checkTicket (ticketNumbers, luckyNumbers)** - A function that accepts two parameters: **array** and **array**.
 - There is a **need for validation** for the input, in case of submitted **invalid** parameters, **throw** an error **"Invalid input!"**
 - To be valid, the **ticket** must meet the **following requirement**:
 - Both **ticketNumbers** and **luckyNumbers** must be **arrays** with exact length of **6** numbers inside.

- After validation the function compares the numbers from the ticket with the winning numbers.
 - If there is **from 3 to 5** winning numbers in the ticket, **return** the following message: **"Congratulations you win, check your reward!"**
 - If **all 6** are winning numbers, **return** the following message: **"You win the JACKPOT!!!"**
- **secondChance (ticketID, secondChanceWinningIDs)** - A function that accepts two parameters: **number** and **array**.
 - There is a **need for validation** for the input, in case of submitted **invalid** parameters, **throw** an error **"Invalid input!"**
 - To be valid, the **ticket** must meet the **following requirement**:
 - **ticketID** must be from type **number**.
 - **secondChanceWinningIDs** must be **array**.
 - After validation the function checks whether the **ticketID** is included in the **secondChanceWinningIDs** array.
 - If there is a match, **return** the following message: **"You win our second chance prize!"**
 - Else, **return** the following message: **"Sorry, your ticket didn't win!"**

JS Code

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

Lottery.js

```
const lottery = {
  buyLotteryTicket(ticketPrice, ticketCount, buy) {
    if (buy === false) {
      throw new Error("Unable to buy lottery ticket!");
    } else {
      if (
        ticketPrice <= 0 ||
        ticketCount < 1 ||
        typeof ticketPrice !== "number" ||
        typeof ticketCount !== "number" ||
        typeof buy !== "boolean"
      ) {
        throw new Error("Invalid input!");
      }
    }
  }
}
```

```

    ) {
        throw new Error("Invalid input!");
    } else {
        let totalPrice = ticketPrice * ticketCount;
        return `You bought ${ticketCount} tickets for ${totalPrice}$.`;
    }
}
},
checkTicket(ticketNumbers, luckyNumbers) {
    if (
        !Array.isArray(ticketNumbers) ||
        !Array.isArray(luckyNumbers) ||
        ticketNumbers.length !== 6 ||
        luckyNumbers.length !== 6
    ) {
        throw new Error("Invalid input!");
    }

    const uniqueTicketNumbers = ticketNumbers.filter(
        (number, index, array) => array.indexOf(number) === index
    );
    let winningNumbers = 0;

    for (const number of uniqueTicketNumbers) {
        if (luckyNumbers.includes(number)) {
            winningNumbers++;
        }
    }

    if (winningNumbers >= 3 && winningNumbers < 6) {
        return "Congratulations you win, check your reward!";
    } else if (winningNumbers === 6) {
        return "You win the JACKPOT!!!";
    }
}
,

```

```
secondChance(ticketID, secondChanceWinningIDs) {  
  if (typeof ticketID !== "number" || !Array.isArray(secondChanceWinningIDs)) {  
    throw new Error("Invalid input!");  
  }  
  if (secondChanceWinningIDs.includes(ticketID)) {  
    return "You win our second chance prize!";  
  } else {  
    return "Sorry, your ticket didn't win!";  
  }  
},  
};  
  
module.exports = lottery;
```

Submission

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