**JS Advanced Retake Exam**

**Link in Judge:** <https://judge.softuni.org/Contests/Practice/Index/3008#2>

**Problem 3. Unit Testing**

**Your Task**

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

* **showMovies(movieArr)**-A function that accepts an array:
* The array includes the available movies in the cinema ([‘King Kong’, ‘The Tomorrow War’, ‘Joker’,etc.])
* If the length of the input array is zero, the function returns the string: "**There are currently no movies to show.**"
* Otherwise, the function returns: an array of available movies, separated by a comma and space
* There is no need for validation for the input, you will always be given an array
* **ticketPrice(projectionType)**- A function that accept string:
  + The function checks whether the submitted projectionType is present in the schedule with the types of projections
  + If present in the schedule, return the price according to the type
* Otherwise, the function throws an error in the following format "**Invalid projection type.**"
* There is no need for validation for the input
* **swapSeatsInHall(firstPlace, secondPlace)**- A function that accepts two numbers
* The function swaps the seat number in the cinema hall, when two numbers are submitted.
* The exchange is not successful and the function returns "**Unsuccessful** **change of seats in the hall.**" :
  + If one of the two numbers do not exist
  + The numbers are not integers
  + If one of the numbers is greater than the capacity of the hall – **20**
  + Seats are less or equal of **0**
* Otherwise return: "**Successful change of seats in the hall.**"
* There is a need for validation for the input

**JS Code**

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

|  |
| --- |
| cinema.js |
| const cinema = {  showMovies: function(movieArr) {  if (movieArr.length == 0) {  return 'There are currently no movies to show.';          } else{           let result = movieArr.join(', ');              return result;          }      },    ticketPrice: function(projectionType) {    const schedule = {              "Premiere": 12.00,              "Normal": 7.50,              "Discount": 5.50          }          if (schedule.hasOwnProperty(projectionType)) {              let price = schedule[projectionType];              return price;          } else {              throw new Error('Invalid projection type.')          }      },    swapSeatsInHall: function(firstPlace, secondPlace) {          if (!Number.isInteger(firstPlace) || firstPlace <= 0 || firstPlace > 20 ||  !Number.isInteger(secondPlace) || secondPlace <= 0 || secondPlace > 20 || firstPlace === secondPlace) {              return "Unsuccessful change of seats in the hall.";          } else {              return "Successful change of seats in the hall.";          }      }  }; |

**Submission**

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