CPEN 400A JavaScript Quiz

Date: Sep 27, 2016

Duration: 60 min

Instructions

A QUnit test script has been provided along this quiz – just load index.html on a browser to run it. You are encouraged to run the test script (refresh index.html) periodically to ensure that your functions satisfy the specifications.

You may use any native JavaScript method to complete the programming tasks below. However, you should not use another JavaScript library (i.e Underscore.js). Unless explicitly stated or required by a test case, you do not need to account for specialized corner cases.

Submission

Create a private Piazza note to **Instructors**, select the **quiz** folder, and upload the **quiz.js** file only. No other files should be included. Make sure that your name and student number are included in your quiz.js file.

Programming Tasks

Please read the specifications carefully and implement the following functions. Focus on **correctness** rather than efficiency.

Each function is worth **1 mark**. You receive full marks for a function if all tests pass. You will lose **0.25 marks** for each failing test case, and will receive 0 for a function if four or more tests fail.

1. **caseInsensitiveStringSearch(strArr, str)**: Returns the integer position of a search string in an array of strings if found. If the string is not found, the function returns -1. The string matches are case insensitive.

2. **lowerCaseEmails(userEmails)**: Accepts an array of objects, consisting of two attributes: an integer user id and a string email address. Returns the array of objects with the email column lowercased. If the email column is missing, return a friendly error message instead.

   Input = [{userId: 1, email: “user1@gmail.com”}, {userId: 2, email: “User2@gmail.com”}]

Result = [{userId: 1, email: “user1@gmail.com”}, {userId: 2, email: “user2@gmail.com”}]

3. **tenSortedIntegers()**: Generates a random array of 10 integers (between 1 and 50 inclusive), and sorts this array in descending order.

4. **combineArrays(arrA, arrB)**: Combines two arrays by alternatively taking elements. The two arrays can have different lengths.

   Input: arrA = [‘a’, ‘b’, ‘c’], arrB = [1, 2]

   Result = [‘a’, 1, ‘b’, 2, ‘c’]

5. **concatAllStringSubsets (str)**: Accepts a single string as input, and returns the concatenation of all its distinct substrings in lexicographical order.

   Input: “gaft”

   Result: “aafaftfftggagafgaftt”

   String Subsets in Lexicographical Order: [“a”, “af”, “aft”, “f”, “ft”, “g”, “ga”, “gaf”, “gaft”, “t”]