**2. Assignment #2 - Testing Your Solver**

Using the code editor to your left, your task is to implement Black-Box Testing for your Boggle solver. Black-box testing is a software testing method in which the internal structure/design/implementation of the item (e.g. function) being tested is not known to the tester.

Your test program must be contained in a single file boggle\_solver.test.jsboggle\_solver.test.js.

We have provided the files for your solution (i.e. boggle\_solver.jsboggle\_solver.js) and your test program (i.e. boggle\_solver.test.jsboggle\_solver.test.js) in the code windows to the left.

NOTE: If you want to use jest from the command line, click on the Tools->Terminal to bring up the terminal. You must change directories to /home/codio/workspace/Boggle\_Testing.

Copy your Boggle solver code into the code window for boggle\_solver.jsboggle\_solver.js.

Make sure your solution runs by selecting TEST SOLVER below, or issue the CLI commands in the terminal (node boggle\_solver.js).

TEST SOLVER

Now lets move on to testing your solution. Copy your Boggle tester code into the code window for boggle\_solver.test.jsboggle\_solver.test.js.

Make sure your solution runs by selecting TEST TESTSUITE below, or issue the CLI commands in the terminal (node boggle\_solver.js).

TEST TESTSUITE

We have provided the file with several test suites using the describe() with several test() functions for you. This project should only utilize the expect() objects from jest to compare the actual results from your function to the expected results. See the example edge case that test for cases if the Dictionary was empty.

Think about your test plan and the many cases to ensure your solution works under  
1 normal input,  
2 problem constraints, and  
3 input edge cases.

**2. Code Given**

*/\*\**

*\* Given a Boggle board and a dictionary, returns a list of available words in*

*\* the dictionary present inside of the Boggle board.*

*\* @param {string[][]}* grid *- The Boggle game board.*

*\* @param {string[]}* dictionary *- The list of available words.*

*\* @returns {string[]} solutions - Possible solutions to the Boggle board.*

*\*/*

exports.findAllSolutions = function(grid, dictionary) {

let solutions = [];

return solutions;

}

var grid = [['t', 'w', 'y', 'r'],

['e', 'n', 'p', 'h'],

['g', 'z', 'qu', 'r'],

['o', 'n', 't', 'a']];

var dictionary = ['art', 'ego', 'gent', 'get', 'net', 'new', 'newt', 'prat',

'pry', 'qua', 'quart', 'quartz', 'rat', 'tar', 'tarp',

'ten', 'went', 'wet', 'arty', 'egg', 'not', 'quar'];

console.log(exports.findAllSolutions(grid, dictionary));