*1. Assignment #1*

Using the code editor to your left, your task is to implement a solver for the Boggle game. Given a valid grid of letters and an arbitrary dictionary of words, return a list of contained words.

Your program must be contained in a single file boggle\_solver.js that exports the function

exports.findAllSolutions = **function**(grid, dictionary) { …… }

We have provide the file and function for you. Feel free to add additional helper functions and data structures as needed to the boggle\_solver.js file.

RULES

* Words must use adjacent tiles, including diagonals.
* Each word may not use a cube more than once.
* Words must be at least 3 letters long.
* Warning: The “Qu” tile counts as 2 letters. There are no raw “Q” tiles. The “St” tile counts as 2 letters. There are no raw “S” tiles.

Examples:  
Input: grid = [[“A”, “B”], [“C”, “D”]],  
dictionary = [“A”, “B”, “AC”, “ACA”, “ACB”, “DE”]  
Output: [“ACB”]

Input: grid = [[“A”, “B”, “C”, “D”], [“E”, “F”, “G”, “H”], [“I”, “J”, “K”, “L”], [“A”, “B”, “C”, “D”]]  
Dictionary = [“ABEF”, “AFJIEB”, “DGKD”, “DGKA”]  
Output: [“ABEF”, “AFJIEB”, “DGKD”]

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'],

['St', '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));