**arrayStats(array)**

arrayStats([9,15,25.5, -5, 5, 7, 10, 5, 11, 30, 4,1,-20]); // Returns: { mean: 7.5, median: 7, mode: 5, range: 50, minimum: -20, maximum: 30, count: 13, sum: 97.5 }  
  
arrayStats([7, 9, 11, 15, 19, 20, 35, 0]); // Returns: { mean: 14.5, median: 13, mode: 0, range: 35, minimum: 0, maximum: 35, count: 8, sum: 116 }  
  
arrayStats([11, 54, 79, 5, -25, 54, 19, 11, 56, 100]); // Returns: { mean: 36.4, median: 36.5, mode: [11,54], range: 125, minimum: -25, maximum: 100, count: 10, sum: 364 }  
  
arrayStats([]) // throws an error   
arrayStats("banana"); // throws an error  
arrayStats(["guitar", 1, 3, "apple"]); // throws an error   
arrayStats(); // throws an error

## makeObjects(array1, array2, array3, ........)

makeObjects([4, 1], [1, 2]); // returns {'4':1, '1': 2}  
makeObjects(["foo", "bar"], [5, "John"]); // returns {foo:'bar', '5': "John"}  
makeObjects (["foo", "bar"], ["name", "Patrick Hill"], ["foo", "not bar"]) //returns {foo: "not bar", name: "Patrick Hill"}  
makeObjects([true, undefined], [null, null]); // returns {true: undefined, null : null}  
makeObjects([undefined, true], ["date", "9/11/2022"]); // returns {undefined: true, date : "9/11/2022"}  
makeObjects([4, 1, 2], [1,2]); // throws error  
makeObjects([]) // throws an error

makeObjects("banana"); // throws an error  
makeObjects(1,2,3); // throws an error  
makeObjects(["guitar", 1, 3, "apple"]); // throws an error

makeObjects(); // throws an error

makeObjects([1],[1,2]); // throws an error

## commonElements(array1, array2, array3, ........)

If any of those conditions fail, you will throw an error.  
const arr1 = [5, 7];   
const arr2 = [20, 5];   
const arr3 = [true, 5, 'Patrick'];   
const arr4 = ["CS-546", 'Patrick'];   
const arr5 = [67.7, 'Patrick', true];   
const arr6 = [true, 5, 'Patrick'];   
const arr7 = [undefined, 5, 'Patrick'];   
const arr8 = [null, undefined, true];  
const arr9 = ["2D case", ["foo", "bar"], "bye bye"]  
const arr10= [["foo", "bar"], true, "String", 10]  
  
commonElements(arr1, arr2); // Returns [5]  
commonElements(arr3,arr4,arr5); // returns ['Patrick']  
commonElements(arr5,arr6); // returns ['Patrick', true]  
commonElements(arr9,arr6); // returns []  
commonElements(arr7,arr8); // returns [undefined]  
commonElements(arr3, arr4, arr5, arr7); // returns ['Patrick']  
commonElements(arr9, arr10); // returns [["foo", "bar"]]  
commonElements(); // throws error  
commonElements("test"); // throws error  
commonElements([1,2,"nope"]); // throws error

# In file2: stringUtils.js

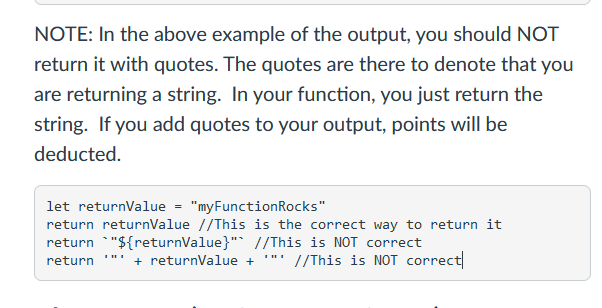
## palindromes(string)

palindromes('Hi mom, At noon, I'm going to take my kayak to the lake'); // Returns: ["mom", "noon", "kayak"]  
palindromes('Wow! Did you see that racecar go?'); // Returns: ["Wow", "Did", "racecar"]  
palindromes('Hello World'); // Returns: []  
palindromes(); // throws error  
palindromes(" "); // throws error  
palindromes(1); //throws error  
palindromes(["hello there"]) //throws error

## replaceChar(string)

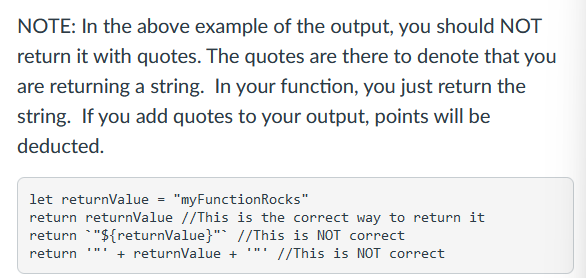
replaceChar("Daddy"); // Returns: "D\*d$y"

replaceChar("Mommy"); // Returns: "M\*m$y"

  
replaceChar("Hello, How are you? I hope you are well"); // Returns: "H\*l$o\* $o\* $r\* $o\*?$I\*h$p\* $o\* $r\* $e\*l"  
replaceChar(""); // Throws Error  
replaceChar(123); // Throws Error

## charSwap(string1, string2)

charSwap("Patrick", "Hill"); //Returns "Hillick Patr"  
charSwap("hello", "world"); //Returns "worlo helld"

  
charSwap("Patrick", ""); //Throws error  
charSwap(); // Throws Error  
charSwap("John") // Throws error  
charSwap ("h", "Hello") // Throws Error  
charSwap ("h","e") // Throws Error

# In file2: objUtils.js

## deepEquality(obj1, obj2)

const first = {a: 2, b: 3};

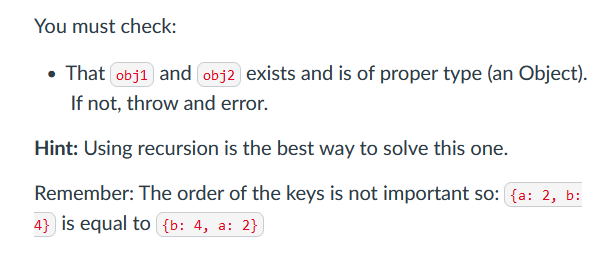
const second = {a: 2, b: 4};

const third = {a: 2, b: 3};  
const forth = {a: {sA: "Hello", sB: "There", sC: "Class"}, b: 7, c: true, d: "Test"}  
const fifth = {c: true, b: 7, d: "Test", a: {sB: "There", sC: "Class", sA: "Hello"}}

console.log(deepEquality(first, second)); // false

console.log(deepEquality(forth, fifth)); // true  
console.log(deepEquality(forth, third)); // false  
console.log(deepEquality({}, {})); // true

console.log(deepEquality([1,2,3], [1,2,3])); // throws error   
console.log(deepEquality("foo", "bar")); // throws error



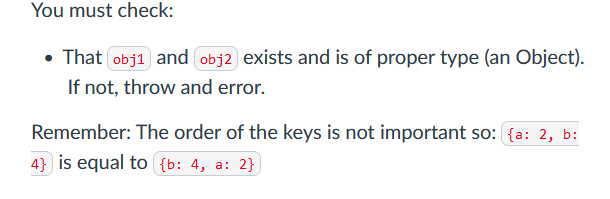
====================================================================

## commonKeysValues(obj1, obj2)

const first = {name: {first: "Patrick", last: "Hill}, age: 46};

const second = {school: "Stevens", name: {first: "Patrick", last: "Hill}};  
const third = {a: 2, b: {c: true, d: false}};  
const forth = {b: {c: true, d: false}, foo: "bar"};

console.log(commonKeysValues(first, second)); // returns  {name: {first: "Patrick", last: "Hill"}, first: "Patrick", last: "Hill"}   
console.log(commonKeysValues(third, forth)); // returns {b: {c: true, d: false}, c: true, d: false }  
console.log(commonKeysValues({}, {})); // {}  
console.log(commonKeysValues({a: 1}, {b: 2})); // {}  
console.log(commonKeysValues([1,2,3], [1,2,3])); // throws error   
console.log(commonKeysValues("foo", "bar")); // throws error



## calculateObject(object, func)

calculateObject({ a: 3, b: 7, c: 5 }, n => n \* 2);

/\* Returns:

{

a: 2.45,

b: 3.74,

c: 3.16

}

\*/

In file4: app.js

// Mean Tests

try {

// Should Pass

const meanOne = mean([2, 3, 4]);

console.log('mean passed successfully');

} catch (e) {

console.error('mean failed test case');

}

try {

// Should Fail

const meanTwo = mean(1234);

console.error('mean did not error');

} catch (e) {

console.log('mean failed successfully');

}