Problem 0 : Part B

## Iterating with JSON object’s Values

Above is some information about my car. As you can see, I am not the best driver.  
I have caused a few accidents.  
Please update this driving record so that I can feel better about my driving skills.

var myCar = {  
make: ‘Bugatti’,  
model: ‘Bugatti La Voiture Noire’,  
year: 2019,  
accidents: [  
{  
date: ‘3/15/2019’,  
damage\_points: ‘5000’,  
atFaultForAccident: true  
},  
{  
date: ‘7/4/2022’,  
damage\_points: ‘2200’,  
atFaultForAccident: true  
},  
{  
date: ‘6/22/2021’,  
damage\_points: ‘7900’,  
atFaultForAccident: true  
}  
]  
}

1. Loop over the accidents array. Change atFaultForAccident from true to false.

**Answer:** var myCar = {  
make: ‘Bugatti’,  
model: ‘Bugatti La Voiture Noire’,  
year: 2019,  
accidents: [  
{  
date: ‘3/15/2019’,  
damage\_points: ‘5000’,  
atFaultForAccident: true,

**atFaultForAccident: false** // **updated**  
},  
{  
date: ‘7/4/2022’,  
damage\_points: ‘2200’,  
atFaultForAccident: true,

**atFaultForAccident: false //updated**  
},  
{  
date: ‘6/22/2021’,  
damage\_points: ‘7900’,  
atFaultForAccident: true,

**atFaultForAccident: false //updated**  
}  
]  
}**console.log(myCar.accidents[0].atFaultForAccident);**

**console.log(myCar.accidents[1].atFaultForAccident);**

**console.log(myCar.accidents[2].atFaultForAccident);**

2. Print the dated of my accidents

Answer:

**Console.log(myCar.accidents[0].date);**

**Console.log(myCar.accidents[1].date);**

**Console.log(myCar.accidents[2].date);**

**Problem 1**

**Parsing an JSON object’s Values:**

Write a function called “printAllValues” which returns an newArray of all the input object’s values.

Input (Object):

var object = {name: “RajiniKanth”, age: 33, hasPets : false};  
Output:

[“RajiniKanth”, 33, false]

**Answer:**

var object = {name : “RajiniKanth”, age :33,hasPets:false};

function printAllValues(object) {

**console.log(object.values(object));** **//**[“RajiniKanth”, 33, false]

}

# Problem 2

## Parsing an JSON object’s Keys:

Write a function called “printAllKeys” which returns an newArray of all the input object’s keys.

Example Input:  
{name : ‘RajiniKanth’, age : 25, hasPets : true}  
Example Output:  
[‘name’, ‘age’, ‘hasPets’]

**Answer:**

Var object ={name : ‘RajiniKanth’, age : 25, hasPets : true}

**function printAllkeys(object) {**

**}**

**console.log(object.keys(object)); //** [‘name’, ‘age’, ‘hasPets’]

# Problem 3

## Parsing an JSON object and convert it to a list:

Write a function called “convertObjectToList” which converts an object literal into an array of arrays.  
Input (Object):  
var object = {name: “ISRO”, age: 35, role: “Scientist”};  
Output:  
[[“name”, “ISRO”], [“age”, 35], [“role”, “Scientist”]]

**Answer:**

var object = {name: “ISRO”, age: 35, role: “Scientist”};

function convertListToObject (object){

}

**Console.log(object.entries(object));** // [[“name”, “ISRO”], [“age”, 35], [“role”, “Scientist”]]

# Problem 4

## Parsing a list and transform the first and last elements of it:

Write a function ‘transformFirstAndLast’ that takes in an array, and returns an object with:  
1) the first element of the array as the object’s key, and  
2) the last element of the array as that key’s value.  
Input (Array):  
var array = [“GUVI”, “I”, “am”, “Geek”];  
Output:  
var object = {   
GUVI : “Geek”  
}

Answer:

var array = [“GUVI”, “I”, “am”, “Geek”];

**function transformFirstAndLast(array) {  
 var myobject {}**

**array,forEach(function(){**

**myObject[array[0]]=array[array.length-1]**

**})  
return newObject;  
}**

**Console.log(tranformFirstAndLast(array);**

**Problem 5**

## Parsing a list of lists and convert into a JSON object:

Write a function “fromListToObject” which takes in an array of arrays, and returns an object with each pair of elements in the array as a key-value pair.  
Input (Array):  
var array = [[“make”, “Ford”], [“model”, “Mustang”], [“year”, 1964]];  
Output:  
var object = {  
make : “Ford”  
model : “Mustang”, year : 1964 }

var array = [[‘make’, ‘Ford’], [‘model’, ‘Mustang’], [‘year’, 1964]];  
**function fromListToObject(array) {  
emptyObject = {};**

**for(var i=0;i<array.length;i++){**

**var newArray= array[i]**

**emptyobject[newArray[0]]=newArray[1];**

**}  
return emptyObject;  
}**

**Var object = fromListToObject(array);**

**Console.log(object); //**

var object = {  
make : “Ford”  
model : “Mustang”

year : 1964 }

# Problem 6

## Parsing a list of lists and convert into a JSON object:

Write a function called “transformGeekData” that transforms some set of data from one format to another.

Input (Array):  
var array = [[[“firstName”, “Vasanth”], [“lastName”, “Raja”], [“age”, 24], [“role”, “JSWizard”]], [[“firstName”, “Sri”], [“lastName”, “Devi”], [“age”, 28], [“role”, “Coder”]]];  
Output:  
[  
{firstName: “Vasanth”, lastName: “Raja”, age: 24, role: “JSWizard”},  
{firstName: “Sri”, lastName: “Devi”, age: 28, role: “Coder”}  
]

Answer: var array = [[[“firstName”, “Vasanth”], [“lastName”, “Raja”], [“age”, 24], [“role”, “JSWizard”]], [[“firstName”, “Sri”], [“lastName”, “Devi”], [“age”, 28], [“role”, “Coder”]]];

Function transformGeekData(array){

EmployeList [];

for(var i=0;i<array.length;i++){

var tempData=array[i];

for(var j =0; j<array[i].length;j++){

tempData[array[i][j][0]]=array[i][j][1]

}

EmployeList.push(tempData);

}

return EmployeList;

}

**Console.log(transformGeekData(array));**

# Problem 7

## Parsing two JSON objects and Compare:

Write an “assertObjectsEqual” function from scratch.  
Assume that the objects in question contain only scalar values (i.e., simple values like strings or numbers).  
It is OK to use JSON.stringify().  
Note: The examples below represent different use cases for the same test. In practice, you should never have multiple tests with the same name.  
Success Case:  
Input:  
var expected = {foo: 5, bar: 6};  
var actual = {foo: 5, bar: 6}  
assertObjectsEqual(actual, expected, ‘detects that two objects are equal’);  
Output:  
Passed  
Failure Case:  
Input:var expected = {foo: 6, bar: 5};  
var actual = {foo: 5, bar: 6}  
assertObjectsEqual(actual, expected, ‘detects that two objects are equal’);  
Output:  
FAILED [my test] Expected {“foo”:6,”bar”:5}, but got {“foo”:5,”bar”:6}

**Answer:**

**function assertObjectsEqual(actual, expected,testName {**

**var actualString = JSON.stringify(actual);**

**var expectedString = JSON.stringify(expected);**

**if(actualString != expectedString) {**

**console.log('FAILED [' + testName + '] Expected "' + expected + '", but got "' + actual + '"');**

**} else {**

**console.log("passed");**

**}**

**}**

**Problem 8**

**Parsing JSON objects and Compare:**

I have a mock data of security Questions and Answers. You function should take the object and a pair of strings and should return if the quest is present and if its valid answer

**var securityQuestions = [  
{  
question: “What was your first pet’s name?”,  
expectedAnswer: “FlufferNutter”  
},  
{  
question: “What was the model year of your first car?”,  
expectedAnswer: “1985”  
},  
{  
question: “What city were you born in?”,  
expectedAnswer: “NYC”  
}  
]**

**function chksecurityQuestions(securityQuestions,question) {  
 var Answer =””;**

**for(var i=0;i< securityQuestions.length;i++){**

**Answer = (securityQuestion[i][“question”]);**

**If(Answer== securityQuestion[i].ecpectedAnswer){**

**Alert(“Incorrect security Question response!”);**

**Break;**

**}if(chksecurityQuestions){**

**return true;**

**}else{**

**return false;**

**}**

**}**

**}**

**}**

**Console.log(chksecurityQuestions(securityQuestions));**