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  
}  
]  
}

Problem 0:

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

Sol: //for(var i=0;i<myCar.accidents.length;i++){

myCar.accidents[i].atFaultForAccident="false";

}

console.log(myCar);

1. Print the dated of my accidents

Sol:// for(var i=0;i<myCar.accidents.length;i++){

console.log(myCar.accidents[i].date);

}

Problem 1:

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

Sol//function printAllKeys(obj) {

const propertyNames = Object.values(obj);

return propertyNames;

}

var obj={name : "RajiniKanth", age : 25, hasPets : true};

console.log(printAllKeys(obj));

Problem 2:

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

Sol//function printAllKeys(obj) {

const propertyNames = Object.keys(obj);

return propertyNames;

}

var obj={name : "RajiniKanth", age : 25, hasPets : true};

console.log(printAllKeys(obj));

Problem 3:

1. Write a function called “convertObjectToList” which converts an object literal into an array of arrays.

Sol//var obj = {name: "ISRO", age: 35, role: "Scientist"};

function convertListToObject(obj) {

const res = Object.entries(obj);

return res;

}

console.log(convertListToObject(obj));

Problem 4:

1. Write a function ‘transformFirstAndLast’ that takes in an array, and returns an object with:

Sol// var arr = ["GUVI", "I", "am", "a geek"];

function transformFirstAndLast(arr) {

const newObject = Object.assign({}, arr);

return newObject;

}

console.log(transformFirstAndLast(arr) );

Problem 5:

1. 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.

Sol// var array = [["make", "Ford"], ["model", "Mustang"], ["year", 1964]];

function transformFirstAndLast(arr) {

const newObject = {};

array.forEach(function(element) {

newObject[element[0]] = element[1];

});

return newObject;

}

console.log(transformFirstAndLast(array) );

Problem 6:

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().

Sol//

var expected = {foo: 5, bar: 6};

var actual = {foo: 5, bar: 9};

var testName="detect ";

function assertObjectsEqual(actual, expected,testName){

if(JSON.stringify(expected)==JSON.stringify(actual)){

console.log ("passed");

}else{

console.log( "FAILED [my test] "+JSON.stringify(expected)+ "but got"+ JSON.stringify(actual));

}

}assertObjectsEqual(actual, expected,testName);