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🡪

for(i=0;i<(myCar.accidents.length);i++)

{

myCar.accidents[i].atFaultForAccident=false

}

1. Print the dated of my accidents

Answer🡪

for(i=0;i<(myCar.accidents.length);i++)

{

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

## 

## 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"};

# console.log(Object.values(object))

# Problem 2(5 mins) :

## 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 obj = {name : "RajiniKanth", age : 33, hasPets : false};

printAllValues(obj)

function printAllValues(obj) {

console.log(Object.keys(obj))

}

# Problem 3( 7–9 mins):

## 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 obj = {name: "ISRO", age: 35, role: "Scientist"};

convertListToObject(obj)

function convertListToObject(obj) {

var a=Object.keys(obj)

console.log([[a[0],obj.name],[a[1],obj.age],[a[2],obj.role]])

}

# Problem 4( 5 mins):

## 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 arr = ["GUVI", "I", "am", "geek"];

var result=transformFirstAndLast(arr)

console.log(result)

function transformFirstAndLast(arr) {

var newObject={}

newObject[arr[0]]=arr[arr.length-1]

return newObject;

}

# Problem 5 ( 7 -9 mins):

## 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  
}

Answer🡪

var arr = [["make", "Ford"], ["model", "Mustang"], ["year", 1964]];

var result=transformFirstAndLast(arr)

console.log(result)

function transformFirstAndLast(arr) {

var newObject={}

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

{

newObject[arr[i][0]]=arr[i][1]

}

return newObject;

}

# Problem 6 (10 mins):

## 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 arr = [[["firstName", "Vasanth"], ["lastName", "Raja"], ["age", 24], ["role", "JSWizard"]], [["firstName", "Sri"], ["lastName", "Devi"], ["age", 28], ["role", "Coder"]]];

var result=transformFirstAndLast(arr)

console.log(result)

function transformFirstAndLast(arr) {

var newObject={}

var newObject1={}

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

{

for(var j=0;j<arr.length;j++)

{

if(i===0)

{

newObject[arr[i][j][0]]=arr[i][j][1]

}

else

{

newObject1[arr[i][j][0]]=arr[i][j][1]

}

}

}

return [newObject,newObject1];

}

# Problem 7 (10 — 20 mins):

## 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}

v

answer🡪

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

var actual = {foo: 7, bar: 6}

assertObjectsEqual(actual, expected, 'detects that two objects are equal');

function assertObjectsEqual(actual, expected, testName) {

var actualS = JSON.stringify(actual);

var expectedS = JSON.stringify(expected);

if(actualS != expectedS) {

console.log('FAILED [' + testName + '] Expected "' + expectedS + '",but got ' + actualS + '"');

} else {

console.log("passed");

}

}

# Problem 8(10 mins):

## 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) {  
  
 // your code here return true or false;   
}//Test case1:var ques = “What was your first pet’s name?”;  
var ans = “FlufferNutter”;var status = chksecurityQuestions(securityQuestions, ques, ans);console.log(status); // true//Test case2:var ques = “What was your first pet’s name?”;  
var ans = “DufferNutter”;var status = chksecurityQuestions(securityQuestions, ques, ans);console.log(status); /

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"

}

]

//Test case1:

var ques = "What was your first pet’s name?";

var ans = "FlufferNutter";

var status = chksecurityQuestions(securityQuestions,ques, ans);

console.log(status); // true

//Tes

//Test case2:

var ques = "What was your first pet’s name?";

var ans = "DufferNutter";

var status1 = chksecurityQuestions(securityQuestions,ques, ans);

console.log(status1);

function chksecurityQuestions(securityQuestions,question,answer) {

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

{

if(question===securityQuestions[i].question && answer===securityQuestions[i].expectedAnswer)

{

return "true";

}

else

return "false"

}

}