<https://medium.com/@reach2arunprakash/guvi-zen-code-sprint-javascript-practice-problems-in-json-objects-and-list-49ac3356a8a5>

Problem 0 : Part A (15 mins):

var cat = {  
 name: ‘Fluffy’,  
 activities: [‘play’, ‘eat cat food’],  
 catFriends: [  
 {  
 name: ‘bar’,  
 activities: [‘be grumpy’, ‘eat bread omblet’],  
 weight: 8,  
 furcolor: ‘white’  
 },   
 {  
 name: ‘foo’,  
 activities: [‘sleep’, ‘pre-sleep naps’],  
 weight: 3  
 }  
 ]  
}console.log(cat);

1. Add height and weight to Fluffy

: cat.height = 10;cat.weight = ‘4kg’

1. Fluffy name is spelled wrongly. Update it to Fluffyy

:Cat.name=cat.name = 'Fluffyy';

1. List all the activities of Fluffyy’s catFriends.

: cat.catFriends[0].activities.slice(1,3)

1. Print the catFriends names.
2. for(i=0;i<cat.catFriends.length;i++){
3. console.log(cat.catFriends[i].name);
4. }

5. Print total weights of friend

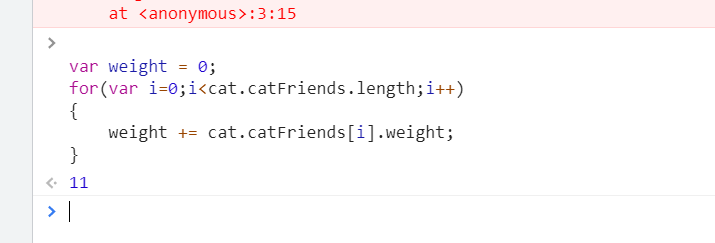
var weight = 0;

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

{

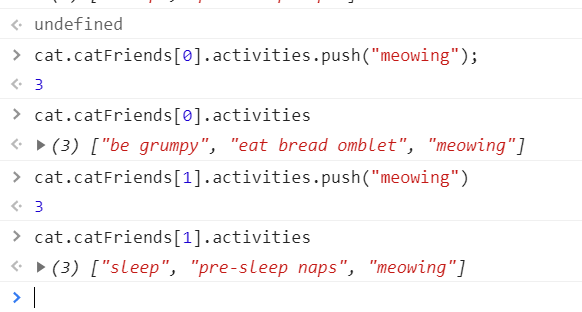
    weight += cat.catFriends[i].weight;

}

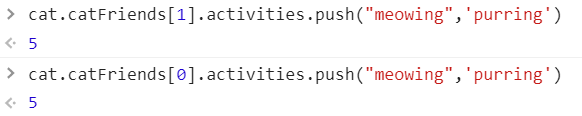


6. Print the total activities of all cats (op:6)





1. Add 2 more activites



We can push 2 activities.

1. Update the fur color of bar



# Problem 0 : Part B (15 mins):

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

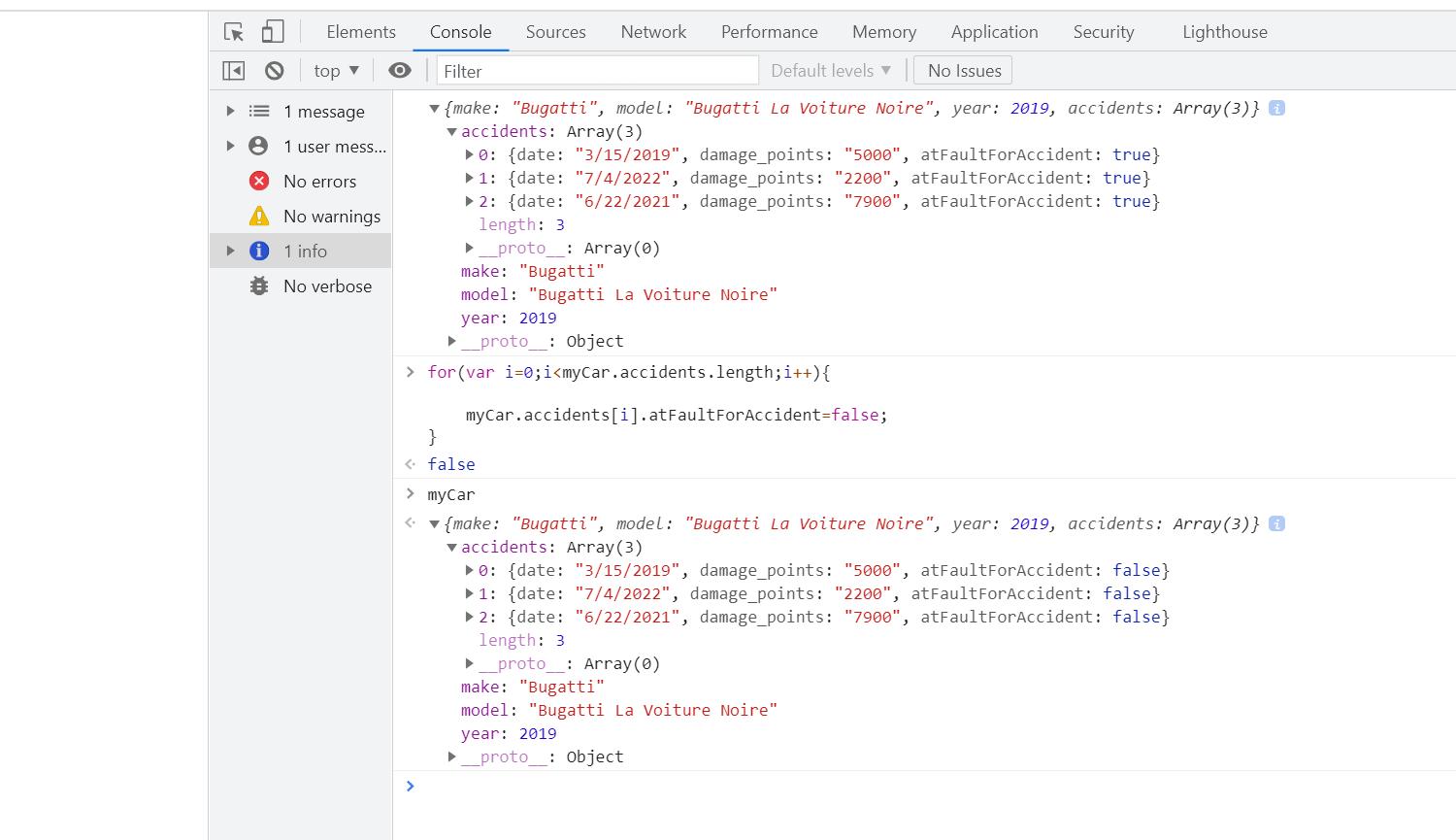
    }

    ]

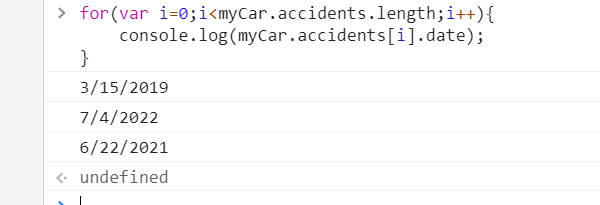
}

console.log(myCar);

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



1. Print the dated of my accidents.



# ****Real challenges starts here****

# :bowtie:

# ****Problem 1 (5 mins):****

## **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]

**Sample Function proto:**

var obj = {name : “RajiniKanth”, age : 33, hasPets : false};function printAllValues(obj) {  
 // your code here  
}

var obj = {name : 'RajiniKanth', age : 33, hasPets : false};

function printAllvalues(obj)

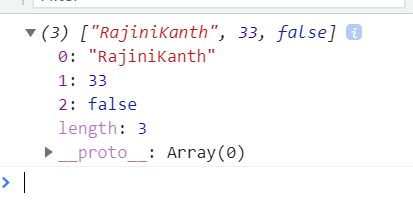
{

    const arr = Object.values(obj);

    return arr

}

console.log(printAllvalues(obj));



# 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’]

**Sample Function proto:**

function printAllKeys(obj) {  
 // your code here  
}

var obj = {name : 'RajiniKanth', age : 33, hasPets : false};

function printAllkeys(obj){

    const arr = Object.keys(obj);

    return arr}

console.log(printAllkeys(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”]]

**Sample Function proto:**

var obj = {name: “ISRO”, age: 35, role: “Scientist”};  
function convertListToObject(obj) {  
 // your code here  
}

var obj = {name: 'ISRO', age: 35, role: 'Scientist'};

/\* Output:

[[“name”, “ISRO”], [“age”, 35], [“role”, “Scientist”]]

function convertObjectToList(obj){

    list1 = Object.entries(obj)

    return list1;

}

convertObjectToList(obj);

console.log(list1);

To read :

<https://www.samanthaming.com/tidbits/76-converting-object-to-array/>

object.keys(objname)

object.values(objname)

object.entries(objname): objname to array

object.fromentries(arrayname): list of array to object

# 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”  
}

**Sample Function proto:**

var arr = [“GUVI”, “I”, “am”, “a geek”];function transformFirstAndLast(arr) {  
   
 return newObject;  
}

var arr = ['GUVI', 'I', 'am', 'Geek'];

function fromListToObject(arr){

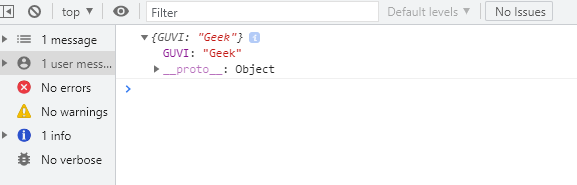
    var newObject = {}

        newObject[arr[0]] = arr[3]

    return newObject

}

console.log(fromListToObject(arr));



# 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  
}

**Sample Function proto:**

var arr = [[“make”, “Ford”], [“model”, “Mustang”], [“year”, 1964]];function fromListToObject(arr) {  
 var newObject = {};  
   
 return newObject;  
}

solution:

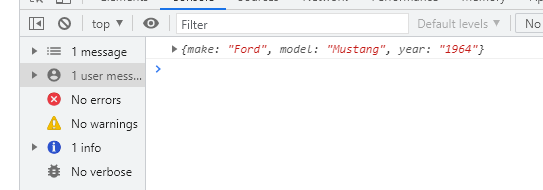
var arr = [['make', 'Ford'], ['model', 'Mustang'], ['year', '1964']];

const entries = new Map(arr);

var obj = {};

obj = Object.fromEntries(entries);

console.log(obj);



Read : <https://stackoverflow.com/questions/4215737/convert-array-to-object>

# 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”}  
]

**Sample Function proto:**

var arr= [[[“firstName”, “Vasanth”], [“lastName”, “Raja”], [“age”, 24], [“role”, “JSWizard”]], [[“firstName”, “Sri”], [“lastName”, “Devi”], [“age”, 28], [“role”, “Coder”]]];function transformEmployeeData(arr) {  
 var tranformEmployeeList = [];  
   
 //Your code  
   
 return tranformEmployeeList;  
}

solution : (knowingly not creating the function as the code is just repetitive. )

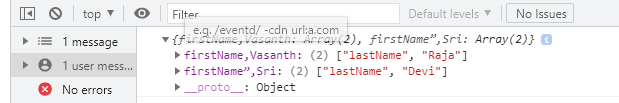
var arr= [[['firstName', 'Vasanth'], ['lastName', 'Raja'], ['age', 24], ['role', 'JSWizard']], [['firstName”', 'Sri'], ['lastName', 'Devi'], ['age', 28], ['role', 'Coder']]];

const entries = new Map(arr);

var obj = {};

obj = Object.fromEntries(entries);

console.log(obj);



# Problem 7 (10 — 20 mins):

## **Parsing two JSON objects and Compare:**

Read this : <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify>

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}

var expected = {foo: 5, bar: 6};  
var actual = {foo: 5, bar: 6}function assertObjectsEqual(actual, expected, testName){  
 // your code here  
}

solution:

obj1 = {foo: 6, bar: 5};

obj2 = {foo: 5, bar: 6};

function assertObjectsEqual(obj1,obj2){

if(JSON.stringify(obj1)===JSON.stringify(obj2))

return true;

else

return false;

}

console.log(assertObjectsEqual(obj1,obj2));

Note:

***JSON.stringify(object1)===JSON.stringify(object2)***

***if key val pair order is changed it will give false otherwise it is fast when order is same.***

***// false***

***\_.isEqual(object1, object2)***

***//true***

<https://stackoverflow.com/questions/26049303/how-to-compare-two-json-have-the-same-properties-without-order>

# 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); // flase

# Solution: Working on It….

# Problem 9(20 mins):

## **Parsing JSON objects and Compare:**

Write a function to return the list of characters below 20 age

var students = [  
 {  
 name: “Siddharth Abhimanyu”, age: 21}, { name: “Malar”, age: 25},  
 {name: “Maari”,age: 18},{name: “Bhallala Deva”,age: 17},  
 {name: “Baahubali”,age: 16},{name: “AAK chandran”,age: 23}, {name:“Gabbar Singh”,age: 33},{name: “Mogambo”,age: 53},  
 {name: “Munnabhai”,age: 40},{name: “Sher Khan”,age: 20},  
 {name: “Chulbul Pandey”,age: 19},{name: “Anthony”,age: 28},  
 {name: “Devdas”,age: 56}   
 ];function returnMinors(arr)  
{}console.log(returnMinors(students));

Solution: Working on it.