Playing with JSON object’s Values:

Fluffy sorry, Fluffyy is my fav cat and it has 2 catFriends  
Write a code to get the below details of Fluffyy so that  
I can take him to vet.

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);

**Basic Tasks to play with JSON**

1. Add height and weight to Fluffy

cat.height= 5

cat.weight=6

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

cat.name=”Fluffyy”

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

    console.log(cat.catFriends[0].activities);

       console.log(cat.catFriends[1].activities);

4.Print the catFriends names.

console.log(cat.catFriends[0].name);

       console.log(cat.catFriends[1].name);

5.Print the total weight of catFriends

       weight = cat.catFriends[0].weight + cat.catFriends[1].weight ;

       console.log(weight);

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

       act = cat.activities + cat.catFriends[0].activities + cat.catFriends[1].activities ;

       console.log(act);

7.Add 2 more activities to bar & foo cats

       act=cat.catFriends[0].activities+" be bold"+" be humble"

       act2=cat.catFriends[1].activities+" exercise"+"ride bike"

       console.log(act);

       console.log(act2);

8.Update the fur color of bar

       console.log(cat.catFriends[0].furcolor="black");

# Problem 0 : Part B (15 mins):

## 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.
2. console.log(myCar.accidents[0].atFaultForAccident=false);
3. console.log(myCar.accidents[1].atFaultForAccident=false);
4. console.log(myCar.accidents[2].atFaultForAccident=false);

2.Print the dated of my accidents

       console.log(myCar.accidents[0].date);

       console.log(myCar.accidents[1].date);

       console.log(myCar.accidents[2].date);

# ****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  
}

**Answer**

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

function printAllValues(obj) {

  var sha = Object.values(obj);

  return sha;

}

var act = printAllValues(obj);

console.log(act);

# 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  
}

**Answer**

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

function printAllValues(obj) {

  var sha = Object.keys(obj);

  return sha;

}

var act = printAllValues(obj);

console.log(act);

# 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  
}

**Answer**

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

function printAllValues(obj) {

  var alllee = Object.entries(obj);

  return alllee ;

}

var act = printAllValues(obj);

console.log(act);

# 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;  
}

**Answer**

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

function transformFirstAndLast(arr) {

  var myObject = {}

  arr.forEach(function(){

  myObject[arr[0]] = arr[arr.length-1]}

  )

  return myObject

}

var result = transformFirstAndLast(arr)

console.log(result);

# 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;  
}

**Answer**

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

function transformFirstAndLast(arr) {

  var obj = {};

  var collection = arr;

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

    var dataArray =collection[i];

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

      obj[dataArray[0]] = dataArray[1]

    }

  }

  return obj;

}

var result = transformFirstAndLast(arr)

console.log(result);

# 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;  
}

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

function transformEmployeeData(arr) {

   var tranformEmployeeList = [];

for(var i in array)

{

    var tempobj = {}

 for(var j in array[i])

 {

    tempobj[array[i][j][0]] = array[i][j][1];

 }

 tranformEmployeeList.push(tempobj)

}

console.log(tranformEmployeeList)

}

transformEmployeeData(array);

# 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}

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

var expected = {foo: 6, bar: 5};

var actual =   {foo: 5, bar: 6};

function assertObjectsEqual(actual, expected, testName){

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

      console.log("Passed "+testName ,"Expected "+JSON.stringify(actual), " and got "+JSON.stringify(expected) )

   }else{

      console.log("Failed "+testName ,"Expected "+JSON.stringify(actual), " but got "+JSON.stringify(expected)  )

   }

  }

  assertObjectsEqual(actual , expected,"Comparison");

# 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

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,ans) {

   for(var i in securityQuestions){

     if (question == securityQuestions[i].question ){

        if(ans == securityQuestions[i].expectedAnswer){

           return true;

        }

   }else{

      return 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); //false

# 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));

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)

 {

   var age20 = arr.filter(student =>{

     return student.age < 20;

   });

   console.log(age20);

 }

 returnMinors(students);