TASK2 - QUESTIONS

1)HTML and script.js file and run a for loop on the data and print all the country names in the console.
2)Write a write up on the difference between copy by value and copy by reference.
3)How to copy by value a composite data type (array+objects).
4)JSON task
https://medium.com/@reach2arunprakash/guvi-zen-code-sprint-javascript-practice-problems-in-json-objects-and-list-49ac3356a8a5
5)Try the rest countries API. Extract and print the total population of all the countries in the console. use the HTML template. https://restcountries.eu/rest/v2/all

Each answer is on a new page.

TASK2 - ANSWERS

1) **HTML Code**

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sample API Data</title>
</head>
<body>
    <script src="./script.js"></script>
</body>
</html>
```

JS code

```js
// step 1. create a request variable
let request = new XMLHttpRequest();

// step 2. create a connection
request.open('GET', 'https://restcountries.eu/rest/v2/all', true);

// step 3. send the request
request.send();

// step 4. load the response
request.onload = function(){
    var data = JSON.parse(this.response);
    console.log(data);
    //task2 - Q1
    for(let i in data){
        console.log(data[i].name);
    }
}
```

2) Consider two data types, primitive and composite.

a. Primitive data types are passed by value i.e, for example, consider the below code

```
var a = 5;
var b = a;
console.log(a,b) \\ 5, 5
```

Here when ‘a’ is assigned to b, it is copied just by value meaning if you change b again, it doesn’t necessarily change the value of a. See the code below

```
var a = 5;
var b = a;
b = 10;
console.log(a,b) \\ 5, 10
```

We can clearly see the difference from above.

b. Composite data are generally passed by reference ie when we create an array as below

```
var array = [1];
var temp = array;
console.log(array, temp) \\ [1], [1]
```

Here while we create an array, it is assigned an address of the location in memory. So when we assign array to temp, temp is copied by reference to the address. Now if we manipulate temp, both temp and arr are changed because they are both pointing to the same reference.

```
var array = [1];
var temp = array;
temp.push(2);
console.log(array, temp) \\ [1,2], [1,2]
```

3) There are 3 ways to copy by value in composite data type

a. Using the spread operator

```
var array = [1];
var temp = [...array];
temp[0] = 2;
console.log(array, temp) \\ [1], [2]
```

b. Using Object.assign()

```
var array = [1];
var temp = Object.assign([], array);
temp[0] = 2;
console.log(array, temp) \\ [1], [2]
```

c. Using JSON.parse() and JSON.stringify();

```
var array = [1];
var temp = JSON.parse(JSON.stringify(array));
temp[0] = 2;
console.log(array, temp) \\ [1], [2]
```

4) **Problem 0: Part A**, script.js

```
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}
    ]
}

//height and weight added
cat.height = '30 cm';
cat.weight = 4;

//name updated
cat.name = 'Fluffyy';

//activities of FLuffyy's catfriends
for(let i=0; i<cat.catFriends.length; i++){
    console.log(cat.catFriends[i].activities);
}
//catfriends names
console.log(cat.catFriends[0].name);
console.log(cat.catFriends[1].name);

//Print the total weight of catFriends
console.log(cat.catFriends[0].weight + cat.catFriends[1].weight);

//Print the total activities of all cats (op:6)
console.log(cat.activities);
console.log(cat.catFriends[0].activities);
console.log(cat.catFriends[1].activities);

//Add 2 more activities to bar & foo cats
cat.catFriends[0].activities.push('cuddle with human');
cat.catFriends[0].activities.push('groom');
cat.catFriends[1].activities.push('do weird things');
cat.catFriends[1].activities.push('play with dog');

//Update the fur color of bar
```

```
cat.catFriends[0].furcolor = 'black';

//print object
console.log(cat);
```

Problem 0: Part B, script.js

```
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
    }
    ]
}

//set atFaultForAccident to false and print the date of accident
for(let i=0; i<myCar.accidents.length; i++){
    myCar.accidents[i].atFaultForAccident = false;
    console.log(myCar.accidents[i].date);
}
```

Problem 1, script.js

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

function printAllValues(obj) {
    for(let keys in obj){
        console.log(obj[keys]);
    }
}

printAllValues(obj);
```

Problem 2, script.js

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

function printAllKeys(obj) {
    let keysArray = [];
    for(let keys in obj){
        keysArray.push(keys);
    }
    return keysArray;
}

console.log(printAllKeys(obj));
```

Problem 3, script.js

```
var obj = {name: 'ISRO', age: 35, role: 'Scientist'};
function convertListToObject(obj) {
    // your code here
    let list = [];
    for(let keys in obj){
        list.push([keys, obj[keys]]);
    }
    return list;
}

console.log(convertListToObject(obj));
```

Problem 4, script.js

```
var array = ['GUVI', 'I', 'am', 'Geek'];
function transformFirstAndLast(arr) {
    let newObject = {};
    let arrLength = arr.length;
    newObject[arr[0]] = arr[arrLength-1];
    return newObject;
}

console.log(transformFirstAndLast(array));
```

Problem 5, script.js

```
var arr = [['make', 'Ford'], ['model', 'Mustang'], ['year', 1964]];
function fromListToObject(arr) {
    var newObject = {};
    for(let i=0; i<arr.length; i++){
        newObject[arr[i][0]] =  arr[i][1];
    }
    return newObject;
}

console.log(fromListToObject(arr));
```

Problem 6, script.js

```
var arr= [[['firstName', 'Vasanth'], ['lastName', 'Raja'], ['age', 24],
['role', 'JSWizard']], [['firstName', 'Sri'], ['lastName', 'Devi'],
['age', 28], ['role', 'Coder']]];
function transformEmployeeData(arr) {
    var transformEmployeeList = [];
    for (let i = 0; i < arr.length; i++){
        transformEmployeeList[i] = {};
        for (let j = 0; j < arr[i].length; j++){
            transformEmployeeList[i][arr[i][j][0]] = arr[i][j][1];
        }
    }
    return transformEmployeeList;
}
console.log(transformEmployeeData(arr));
```

Problem 7, script.js

```
var expected = {foo: 5, bar: 6};
var actual = {foo: 6, bar: 6}
function assertObjectsEqual(actual, expected, testName){
    let actualString = JSON.stringify(actual);
    let expectedString = JSON.stringify(expected);
    if(actualString !== expectedString){
        console.log('FAILED [' + testName + '] Expected ' + expectedString
+ ', but got ' + actualString);
    }
    else{
        console.log('Passed');
    }
}
assertObjectsEqual(actual, expected, 'detects that two objects are
equal');
```

Problem 8, script.js

```
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) {
    let answerGiven = false;
    for(let i=0; i<securityQuestions.length; i++){
        if(securityQuestions[i].question === question){
            if(securityQuestions[i].expectedAnswer === ans){
                answerGiven = true;
            }
        }
    }
    return answerGiven;
}

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

```
var ans  =  'FlufferNutter';
console.log(chksecurityQuestions(securityQuestions, ques, ans));

var ques1 = 'What was your first pet’s name?';
var ans1  =  '1985';
console.log(chksecurityQuestions(securityQuestions, ques1, ans1));
```

Problem 9, script.js

```
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)
{
    let minors = [];
    for(let i=0; i<arr.length; i++){
        if(arr[i].age<20){
            minors.push(arr[i].name);
        }
    }
    return minors;
}
console.log(returnMinors(students));
```

5) **HTML Code**

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sample API Data</title>
</head>
<body>
    <script src="./script.js"></script>
</body>
</html>
```

JS Code

```javascript
// step 1. create a request variable
let request = new XMLHttpRequest();

// step 2. create a connection
request.open('GET', 'https://restcountries.eu/rest/v2/all', true);

// step 3. send the request
request.send();

// step 4. load the response
request.onload = function(){
    var data = JSON.parse(this.response);
    console.log(data);
    //task2 - Q5
    let totalWorldPopulation = 0;
    for(let i in data){
        totalWorldPopulation += data[i].population;
    }
    console.log(totalWorldPopulation);

}
```

Mannam Akhil
mannamakhil@gmail.com