TASK 2

Question 1

Create a html and script.js file and run a for loop on the data and print all the country names in the console.

```
//step1. create a request
var request = new XMLHttpRequest;
//step2. open the connection
request.open('GET', 'https://restcountries.eu/rest/v2/all', true);
//step3. sending the request
request.send();
//step4. process and load the request
request.onload = function () {
    var data = JSON.parse(this.response);
    console.log(data);
    for (i = 0; i < data.length - 1; i++) {
        console.log(data[i].name)
    }
```

Question 2

Write a write up on Difference between copy by value and copy by reference.

- Copy by value occurs when the copying process is involved with the primitive data types whereas, copy by reference is done when we are dealing with the non - primitive or composite data types.

- When the copying is done, that is the value is copied into a new variable, the data is stored in the new variable, in case of primitive data types.

- Whereas, during the copying, when the value is copied into a new variable, the memory location of the data is stored in the new variable.

- Hence any changes made to the new variable in the primitive case does not affect the old values and both the variables point to the same memory location.

- But unlike primitive values, in non - primitive data types, any changes made to the new variable in the primitive case also affects the old values.

- Example

```
// Copy by value
var number1 = 10;
var number2 = number1;
number2 = 20;
console.log(number2, number1); // output is 20 10
```

```
// Copy by reference
var arr1 = [10,20];
var arr2 = arr1;
arr2.push(30);
console.log(arr2, arr1); // output is [10,20,30], [10,20,30]
```

Question 3

How to copy by value a composite data type (array+objects).

There are 3 ways to copy by value for composite data types.

1. Using the spread (...) operator
2. Using the Object method
3. Using the JSON.stringify() and JSON.parse() methods

1. Using Spread

Spread syntax (...) allows an iterable such as an array expression or string to be expanded in places where zero or more arguments (for function calls) or elements (for array literals) are expected, or an object expression to be expanded in places where zero or more key-value pairs (for object literals) are expected.

```
var arr1 = [10,20];

arr2 = [...arr1];
```

```
arr2.push(30);

console.log(arr2, arr1); // output is [10,20,30], [10,20]
```

2. Using Object.assign()

The Object.assign() method copies all enumerable own properties from one or more *source objects* to a *target object*. It returns the target object. Note this will be a shallow copy.

```
var arr1 = [10,20];
arr2 = Object.assign([],arr1);
arr2.push(30);
console.log(arr2, arr1);
// output is [10,20,30], [10,20]
```

Note the empty *[]* as the first argument, this will ensure you don't mutate the original object

3. Using JSON.parse() and JSON.stringify()

The JSON object, available in all modern browsers, has two useful methods to deal with JSON-formatted content: parse and stringify. JSON.parse() takes a JSON string and transforms it into a JavaScript object. JSON.stringify() takes a JavaScript object and transforms it into a JSON string.Using JSON.parse() and JSON.stringify() for copy performs deep copy .

```
var arr1 = [10,20];
arr2 = JSON.parse(JSON.stringify(arr1));
arr2.push(30);
console.log(arr2, arr1); // output is [10,20,30], [10,20]
```

The deep copy is a true copy for nested objects. Shallow copy copies only reference in case of nested objects.

Question 4

PART A - Question

```
var cat = { name : "Fluffy",
            activities : ["play", "eat cat food"],
            height : "10", weight : "5",
            catFriends: [
            {
                name : "bar",
                activities : ["be grumpy", "eat bread omlet"],
                weight: "8",
                color: "white" },
            {
                name : "foo",
                activities : ["sleep", "pre-sleep naps"],
                weight : "3" }]};
```

Codes for tasks in Part A

```
//1.Fluffy name is spelled wrongly. Update it to Fluffyy
cat.name = "Fluffyy";

//3. List all the activities of Fluffyy's catFriends.
for(var z=0; z<cat.catFriends.length-1; z++){
    console.log(cat.catFriends[z].activities);
}

//4. Print the catFriends names.
for(var k=0; k<cat.catFriends.length-1; k++){
    console.log(cat.catFriends[k].name);
}

//5. Print the total weight of catFriends
var totalWeight = 0;
for(var l=0; l<cat.catFriends.length-1; l++){
    totalWeight = totalWeight + parseInt(cat.catFriends[l].weight);
}
```

```
//6. Print the total activities of all cats
var totalActivities = 0;
for(var a=0; a<3; a++){
    totalActivities = totalActivities + cat.activities.length;
}
console.log(totalActivities);

//7. Add 2 more activities to bar & foo cats
console.log(totalActivities);
cat.catFriends[0].gender = "female";
cat.catFriends[0].age = "10";
cat.catFriends[1].gender = "male";
cat.catFriends[1].age = "7";

//8. Update the fur color of bar
cat.catFriends[0].color = "black";
```

PART B

Given Question

```
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
 }
 ]
}
```

Codes for tasks

```
//1. Loop over the accidents array. Change atFaultForAccident from
true to false.
for(var i in myCar){
    myCar.accidents[i].atFaultForAccident = false;
}
console.log(myCar);
//2. Print the dates of my accidents
for(var i=0; i<3; i++){
    console.log(myCar.accidents[i].date);
}
```

PROBLEM 1

```
//Write a function called "printAllValues" which returns an newArray
of all the input object's values.
var object = {name: "RajiniKanth", age: 33, hasPets : false};
function printAllValues(obj) {
 console.log(Object.values(object));
}
printAllValues();
```

PROBLEM 2

```
//Write a function called "printAllValues" which returns an newArray
of all the input object's keys.
var object = {name: "RajiniKanth", age: 33, hasPets : false};
function printAllKeys(obj) {
 console.log(Object.keys(object));
}
printAllKeys();
```

PROBLEM 3

```
//Parsing an JSON object and convert it to a list:
var object = {name: "ISRO", age: 35, role: "Scientist"};
function convertObjectTolist(){
    console.log(Object.entries(object));
}
convertObjectTolist();
```

PROBLEM 4
Question - Parsing a list and transform the first and last element

```
//Parsing a list and transform the first and last elements of it
var array = ["GUVI", "I", "am", "GEEK"];
function transformFirstAndLast(arr) {
    var object = {};
    var key = array[0];
    object[key] = array[array.length-1];
    console.log(object);
}
transformFirstAndLast();
```

PROBLEM 5
Question - Parsing a list of lists and convert into a JSON object:

```
var arr = [["make", "Ford"], ["model", "mustang"], ["year", 1964]];
function fromListToObject() {
    let newObject = {};
    for(let i in arr){
        var key = arr[i][0];
        newObject[key] = arr[i][1];
    }
    console.log(newObject);
}
fromListToObject();
//Output is { make: 'Ford', model: 'mustang', year: 1964 }
```

PROBLEM 6
Parsing a list of lists and convert into a JSON object

```
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 in arr){
```

```
        transformEmployeeList[i] = {};
        for(let j in arr[i]){
            var key = arr[i][j][0];
            transformEmployeeList[i][key] = arr[i][j][1];
        }
    }
    return transformEmployeeList;
}
console.log(transformEmployeeData(arr));

// Output ids [ { firstname: 'Vasanth', lastName: 'Raja' age: 24, role:
'JSWizard' }, { firstName: 'Sri', lastName: 'Devi', age: 28, role: 'Coder'
} ]
```

PROBLEM 7

```
var expected = {foo: 5, bar: 6};
var actual = {foo: 5, bar: 9}
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
        testName = console.log("Passed");
    return testName;

}
assertObjectsEqual(actual, expected, 'detects that two object are equal');
//Output is "Passed"
```

PROBLEM 8
Parsing JSON objects and compare

```
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,answer) {
       var theAnswer = false;
       for(var i in securityQuestions){
           if(securityQuestions[i].question===question){
               if(securityQuestions[i].expectedAnswer===answer)
               theAnswer = true;
           }
       }
       return theAnswer;
   }


   //Test case1:
   var ques1 = 'What was your first pet's name?';
   var ans1  =  'FlufferNutter';
   console.log(chksecurityQuestions(securityQuestions,ques1,ans1));

   //Test case2:
   var ques2 = 'What was your first pet's name?';
   var ans2  =  'DufferNutter';
   console.log(chksecurityQuestions(securityQuestions,ques2,ans2));
```

PROBLEM 9
Question - 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}
 ];
```

Code - solution

```
function returnMinors(arr)
{
    for(var i=0; i<students.length-1; i++){
        if(students[i].age<20){
            console.log(students[i].name);
        }
    }
}
returnMinors(students);
```

Question 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

```
var request = new XMLHttpRequest;
request.open('GET', 'https://restcountries.eu/rest/v2/all', true);
request.send();
request.onload = function () {
    var data = JSON.parse(this.response);
    console.log(data);
    var totalPopulation = 0;
    for (k = 0; k < data.length; k++) {
        totalPopulation = totalPopulation +
parseInt(data[k].population);
    }
    console.log(totalPopulation);
}
```

```
//total population is 7349137231
```