**RVS College of Arts & Science**

**(Autonomous), Sulur, Coimbatore-641402**

**School of Computer Studies - MCA**

**Master of Computer Applications**

**Name : BHARATH P**

**Register Number : 1P24MC006**

**Class : I - MCA**

**Subject : Web Development with JavaScript**

**RathnavelSubramaniam College of Arts & Science (Autonomous),Sulur, Coimbatore-641402**

**School of Computer Studies - MCA**

**Master of Computer Applications**

**Bonafide Certificate**

Certified that this is a bonafide record of ……………………………………………Practical work done by Mr./Ms………………………………………. Register No………………….pursuing ………..in First Semester during the academic year 2023-2024.

**Faculty HOD**

**Submitted for the practical examination held at Rathnavel Subramaniam College of Arts & Science on ………………………**

**Internal Examiner External Examiner**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Date** | **Problem** | **Page No** | **Sign** |
| **1** |  | Write a JavaScript code that will allow the ball to move between two fixed positions on the x-axis and y-axis on the page. |  |  |
| **2** |  | Write a JavaScript program to create number of balls within a defined area of the screen and make these balls move randomly within that area. |  |  |
| **3** |  | Write a JavaScript program that displays PacMan and makes him move from one side of the screen to the other side. |  |  |
| **4** |  | Write JavaScript code that will display the top five earners in Boston data (salaries) on the web page. |  |  |
| **5** |  | Write a React program to display any employees who make a minimum of 200k(salaries) per year in Boston data. |  |  |
| **6** |  | Write a Javascript program to print the first three most common words that given in a text. |  |  |
| **7** |  | Write a Javascript code to create a PacMan on demand (button click) and move these PacMan across the page. |  |  |
| **8** |  | Write a Javascript code to add a two "eyeball" element and have that element follow the mouse position on the screen using the mouse events. |  |  |
| **9** |  | Write a Javascript program to create map use the mapboxgl library to add a marker to the map. Then, to move the marker on the map from one bus stop to the next. |  |  |
| **10** |  | Write a Javascript program to edit the hanoi tower algorithm to add a custom number of disks to the tower using recursion function. |  |  |

**Program -1**

**Date:**

**Write a JavaScript code that will allow the ball to move between two fixed positions on the x-axis and y-axis on the page.**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>BALL</title>

</head>

<body>

    <style>

        html{

            transition: smooth;

        }

    </style>

    <div id="ball" style="

        background-color: red;

        top: 0px;

        left: 0px;

        border-radius: 50%;

        position: absolute;

        z-index: 5;

        height: 150px;

        width: 150px;

    ">

    </div>

    <script>

        var position=0

        var velocity=3

        var moveleft=2

        function movingBall(){

            position+=moveleft\*velocity

            var ball=document.getElementById("ball")

            if(position>=90 || position<=0){

                moveleft\*=-1

            }

            ball.style.top=position+'%'

            ball.style.left=position+'%'

        }

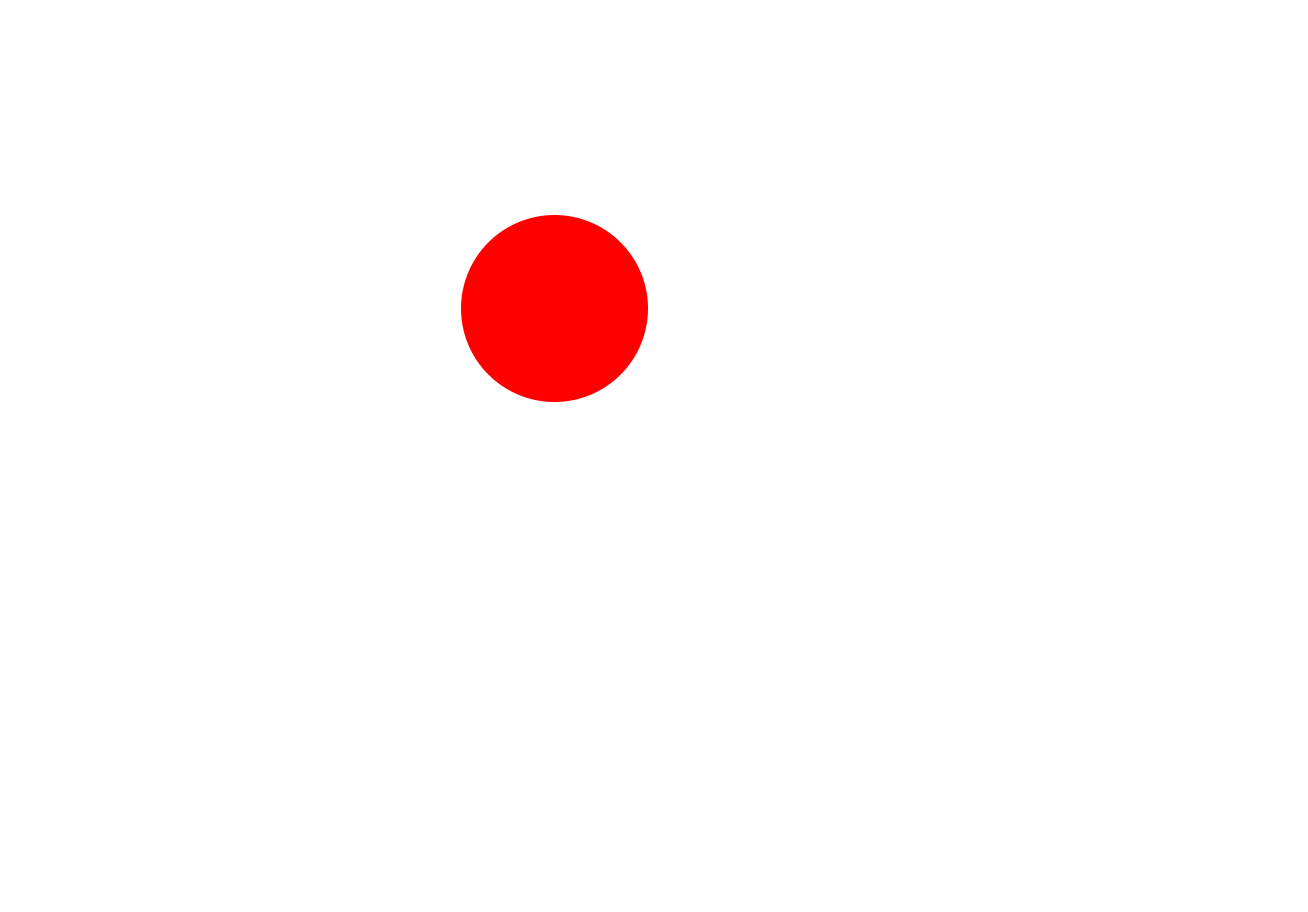
        setInterval(movingBall,100)

    </script>

</body>

</html>

**Output:**



**Program-2**

**Date:**

**Write a JavaScript program to create number of balls within a defined area of the screen and make these balls move randomly within that area.**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <div style="border: 3px solid black;

    height: 700px;

    width: 1099px;

    border-radius: 80;"></div>

    <script>

        function create(){

            var part=document.createElement('div')

            let color=randomColor()

            part.style.top='50px';

            part.style.left='50px';

            part.style.width='50px';

            part.style.height='50px';

            part.style.background=color;

            part.style.borderRadius='50%';

            part.style.borderColor = "black";

            part.style.position='absolute';

            document.body.appendChild(part)

            return part

        }

        function randomColor(){

            let palet=["red","blue","green","skyblue"]

            let color=palet[Math.floor(Math.random()\*4)]

            return color

        }

        let arr=[]

        for(let i=0;i<10;i++){

            arr.push(create())

        }

        let positionsX = [];

        let positionsY = [];

        let velocitiesX = [];

        let velocitiesY = [];

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

            positionsX[i]=Math.floor(Math.random()\* (1080-50)+50);

            positionsY[i]=Math.floor(Math.random()\* (700-50)+50);

            velocitiesX[i]=Math.floor(Math.random()\*(5-(-5))+(-5));

            velocitiesY[i]=Math.floor(Math.random()\*(5-(-5))+(-5));

        }

        function move()

        {

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

                positionsX[i]+=velocitiesX[i];

                positionsY[i]+=velocitiesY[i];

                arr[i].style.left=positionsX[i]+'px';

                arr[i].style.top=positionsY[i]+'px';

                if(positionsX[i]>1080 || positionsX[i]<0){

                    velocitiesX[i]\*=-1;

                }

                if(positionsY[i]>500 || positionsY[i]<0){

                    velocitiesY[i]\*=-1;

                }

            }

        }

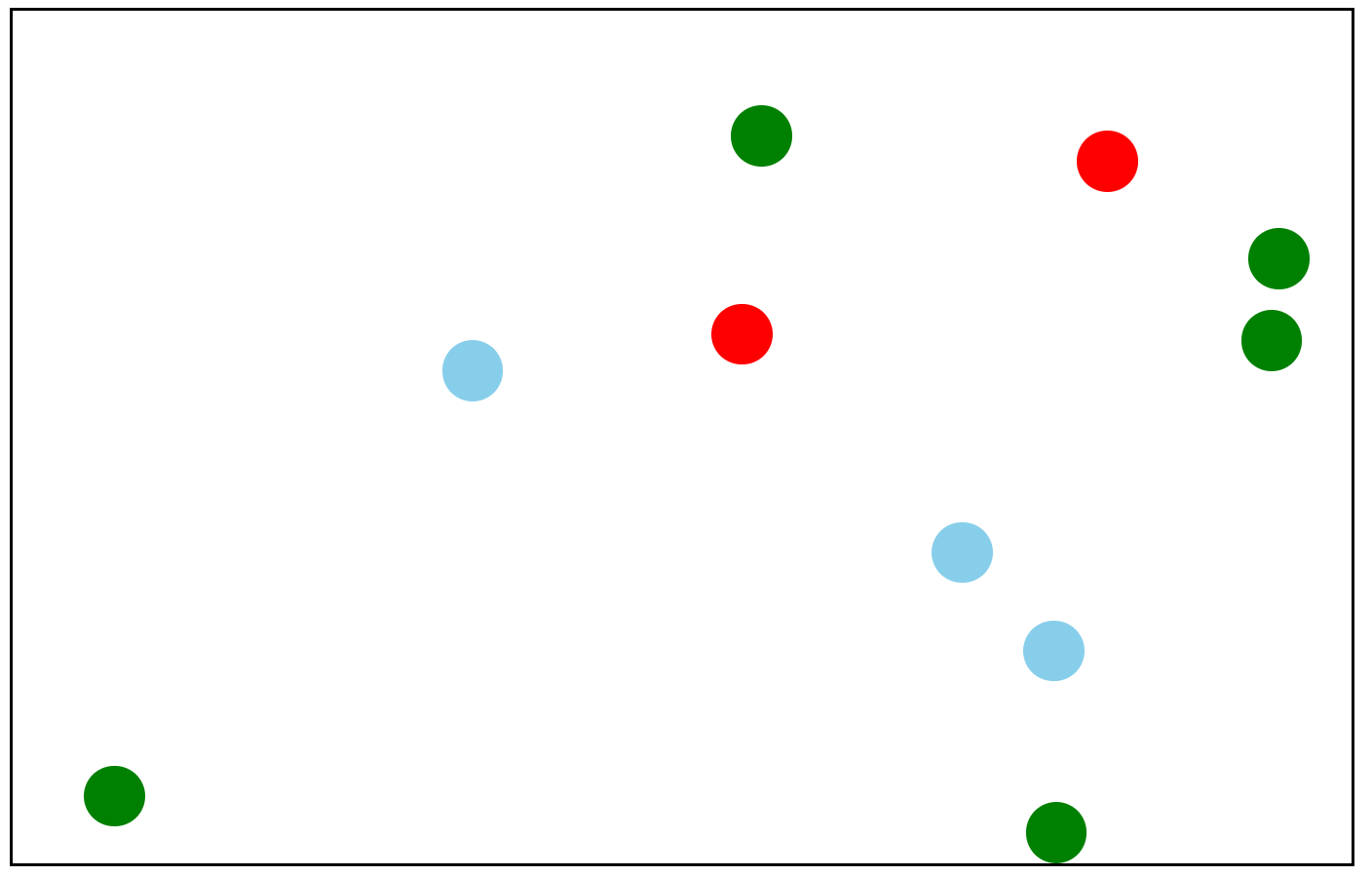
        setInterval(move, 16);

    </script>

</body>

</html>

**Output:**



**Program-3**

**Date:**

**Write a JavaScript program that displays PacMan and makes him move from one side of the screen to the other side.**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Coding Assignment 3</title>

</head>

<body>

    <div>

        <button onclick="create()">Start</button>

        <img id="img" src="PacMan1.png" style="display: none;">

    </div>

    <style>

        img {

            position: absolute;

            height: 100px;

            width: 100px;

            top: 100px;

            left: 50px;

        }

    </style>

    <script>

        function create(){

            isGameStarted=true;

            elem.style.display='block'

        }

        var imagesRight = ['PacMan1.png', 'PacMan2.png'];

        var imagesLeft = ['PacMan3.png', 'PacMan4.png'];

        let currentIndex = 0;

        var imageElement = document.getElementById('img');

        var currentImages = imagesRight;

        var isGameStarted=false;

        setInterval(imgMove, 200);

        function imgMove() {

            imageElement.src = currentImages[currentIndex];

            currentIndex = (currentIndex + 1) % currentImages.length;

        }

        var elem = document.getElementById('img');

        var pos = 0;

        var vel = 5;

        function move() {

            if (isGameStarted){

            pos = pos + vel;

            if (pos > window.innerWidth - elem.width || pos < 0) {

                vel = -vel;

                currentImages = vel > 0 ? imagesRight : imagesLeft;

            }

            elem.style.left = pos + 'px';

        }

    }

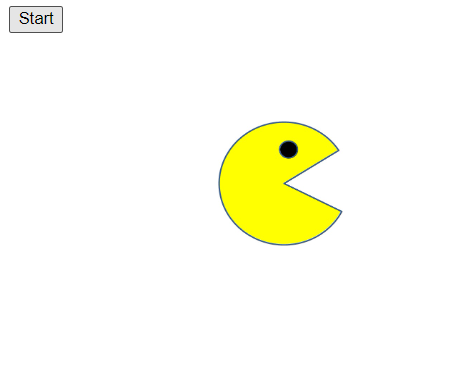
        setInterval(move, 10);

    </script>

</body>

</html>

**Output:**



**Program-4**

**Date:**

**Write JavaScript code that will display the top five earners in Boston data (salaries) on the web page.**

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<div id="app"></div>

<script type="module" src="data.js">

</script>

</body>

</html>

**data.js**

import { boston } from "./boston.js";

let data = boston.data;

let topearners = [];

for (let i = 0; i < data.length; i++) {

if (data[i][11] < 20000000) {

topearners.push({ name: data[i][8], salary: data[i][11] });

}

}

topearners.sort(function(a, b) {

return b.salary - a.salary;

});

topearners = topearners.slice(0, 5);

let Earners = '<h1><u>Top Earners:</u></h1>';

for (let i = 0; i < 5; i++) {

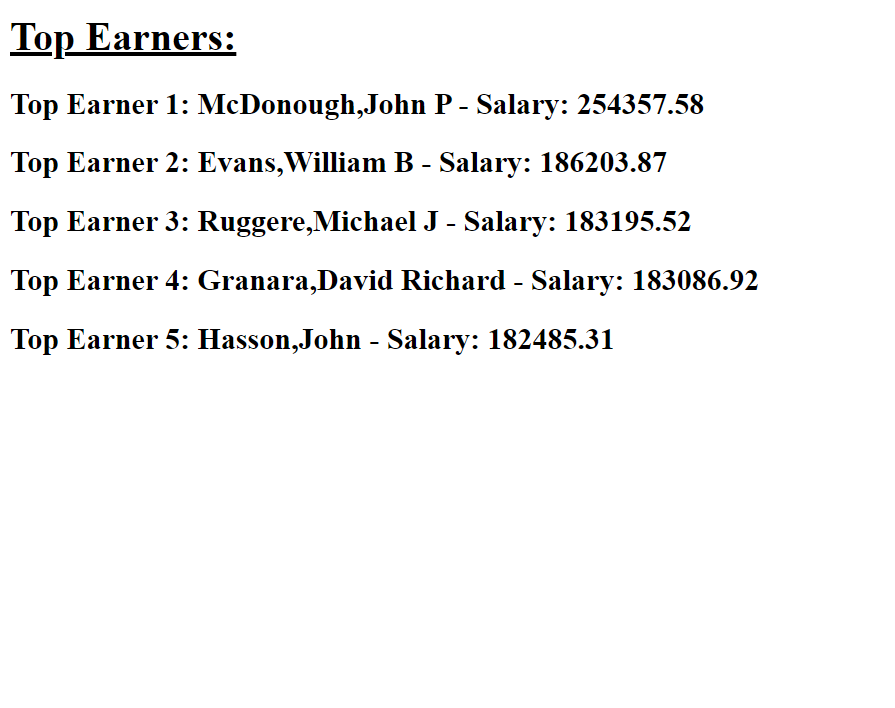
Earners += `<h2>Top Earner ${i+1}: ${topearners[i].name} - Salary: ${topearners[i].salary}</h2>`;

}

document.getElementById("app").innerHTML = Earners;

console.log(topearners)

**output:**



**program -5**

**Date:**

**Write a React program to display any employees who make a minimum of 200k(salaries) per year in Boston data.**

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<div id="app"></div>

<script type="module" src="data.js">

</script>

</body>

</html>

**data.js**

import { boston } from "./boston.js";

let data = boston.data;

let topearners = [];

for (let i = 0; i < data.length; i++) {

if (data[i][11] < 200000) {

topearners.push({ name: data[i][8], salary: data[i][11] });

}

}

topearners.sort(function(a, b) {

return b.salary - a.salary;

});

topearners = topearners.slice(0, 5);

let Earners = '<h1><u>Top Earners:</u></h1>';

for (let i = 0; i < 1; i++) {

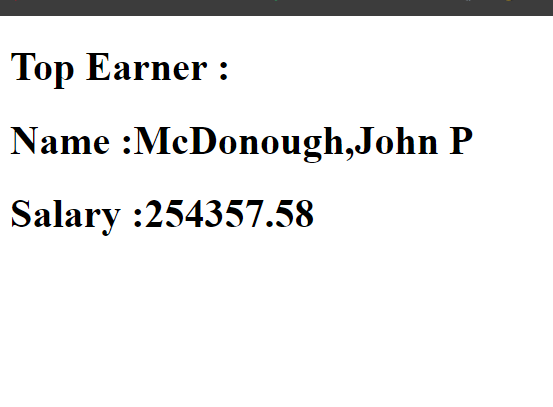
Earners += `<h2>Top Earner ${i+1}: ${topearners[i].name} - Salary: ${topearners[i].salary}</h2>`;

}

document.getElementById("app").innerHTML = Earners;

console.log(topearners)

**output:**



**Program-6**

**Date:**

**Write a Javascript program to print the first three most common words that given in a text**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Top Words</title>

</head>

<body>

    <h1 id="result"></h1>

    <script>

        let text = "The quick brown fox jumps over the lazy dog. The sun is shining bright and clear. The air is filled with the sweet scent of blooming flowers. The dog is running quickly and jumping high. The fox is trying to catch it, but it is too fast for him.";

        text = text.toLowerCase();

        text = text.replace(/[^\w\s]/gi, "");

        const words = text.split(" ");

        let wordCount = {};

        for (let word of words) {

            if (word) {

                wordCount[word] = (wordCount[word] || 0) + 1;

            }

        }

        let wordCountArray = Object.entries(wordCount);

        wordCountArray.sort((a, b) => b[1] - a[1]);

        let topThreeWords = wordCountArray.slice(0, 3).map(entry => `${entry[0]} (${entry[1]})`);

        let resultText = "The top three most common words are: " + topThreeWords.join(", ").toLowerCase();

        console.log(resultText);

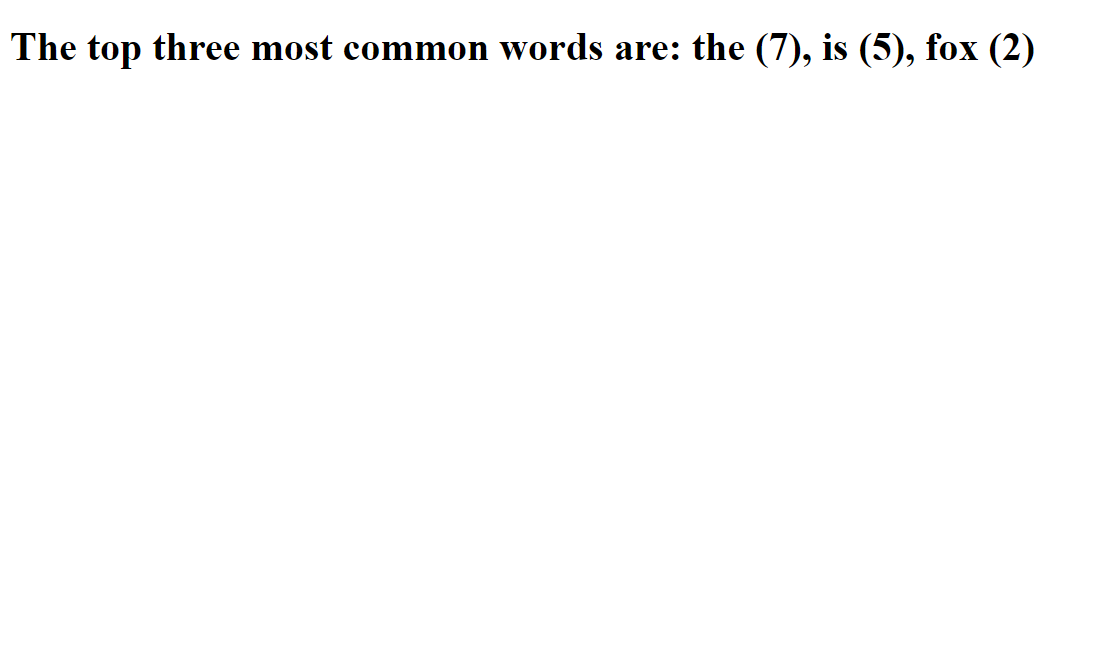
        document.getElementById("result").innerHTML = resultText;

    </script>

</body>

</html>

**Output:**



**Program-7**

**Date:**

**Write a Javascript code to create a PacMan on demand (button click) and move these PacMan across the page.**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <button onclick="ballCount()">add</button>

    <button onclick="moveBall()">move</button>

    <script>

            var imgs=[]

            var x\_pos=[]

            var y\_pos=[]

            var x\_vel=[]

            var y\_vel=[]

            var gravity=1

        function ballCount(){

                var img=document.createElement("img")

                img.src="PacMan1.png"

                img.style.width="50px"

                img.style.height="50px"

                img.style.position="absolute"

                img.style.zIndex="5"

                ypos=Math.floor(Math.random()\*600)

                xpos=Math.floor(Math.random()\*1000)

                img.style.left=xpos+'px'

                img.style.top=ypos+'px'

                xvel=5

                yvel=5

                document.body.appendChild(img)

                imgs.push(img)

                x\_pos.push(xpos)

                y\_pos.push(ypos)

                x\_vel.push(xvel)

                y\_vel.push(yvel)

     }

function moveBall(){

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

                x\_pos[i]+=x\_vel[i]

                y\_pos[i]+=y\_vel[i]

                if(x\_pos[i] > 1300 || x\_pos[i] <= 0) {

                    x\_vel[i] =-x\_vel[i]\*gravity

                }

                if(y\_pos[i] > 610 || y\_pos[i] <= 0) {

                    y\_vel[i] =-y\_vel[i]\*gravity

                }

                imgs[i].style.left=x\_pos[i]+'px'

                imgs[i].style.top=y\_pos[i]+'px'

            }

            setTimeout(moveBall,20)

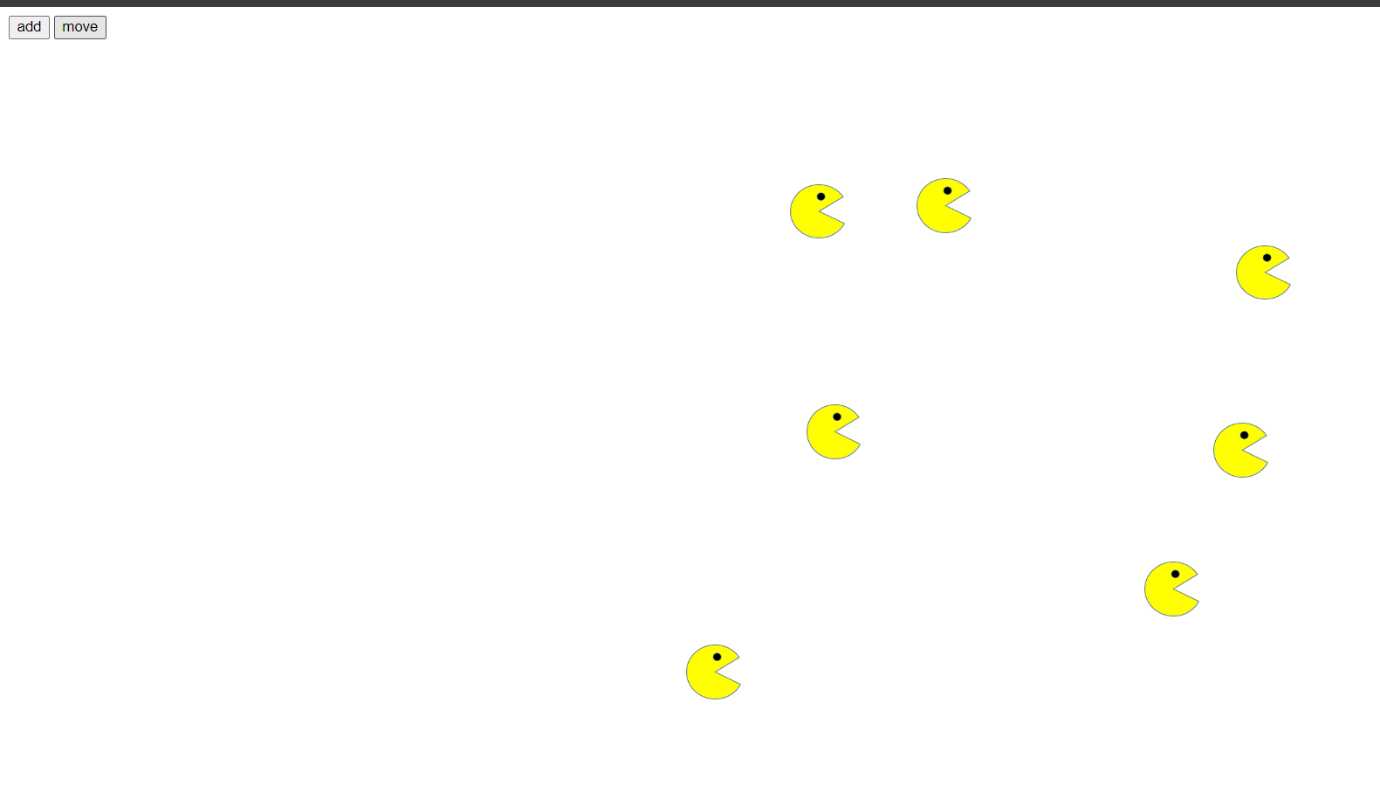
        }

</script>

</body>

</html>

**Output:**



**Program-8**

**Date:**

**Write a Javascript code to add a two "eyeball" element and have that element follow the mouse position on the screen using the mouse events.**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Coding Assignment 7</title>

</head>

<body onmousemove="move(event)">

    <div class="eyes">

        <div class="eye">

          <div class="eyeball"></div>

        </div>

        <div class="eye">

          <div class="eyeball"></div>

        </div>

      </div>

<style>

    \*{

        margin: 0;

        padding: 0;

    }

    body {

    width: 100%;

    height: 650px;

  background-color: #000;

}

.eyes {

  width: 100%;

  height: 200px;

  display: flex;

  justify-content: center;

  align-items: center;

  flex-direction: row;

  column-gap: 20px;

}

.eye {

  width: 250px;

  height: 150px;

  margin-top: 400px;

  border-radius: 100%;

  background-color: #fff;

  display: flex;

  justify-content: center;

  align-items: center;

}

.eyeball {

  width: 100px;

  height: 100px;

  border-radius: 101px;

  background-color: #000;

}

</style>

<script>

const eyeballs = document.querySelectorAll(".eyeball");

function move(item){

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

    {

        var rectX=eyeballs[i].offsetLeft

        var rectY=eyeballs[i].offsetTop

        var x = (item.pageX - rectX) / 10 + "px";

        var y = (item.pageY - rectY) / 10 + "px";

        eyeballs[i].style.transform = "translate3d(" + x + "," + y + ", 0px)";

    }

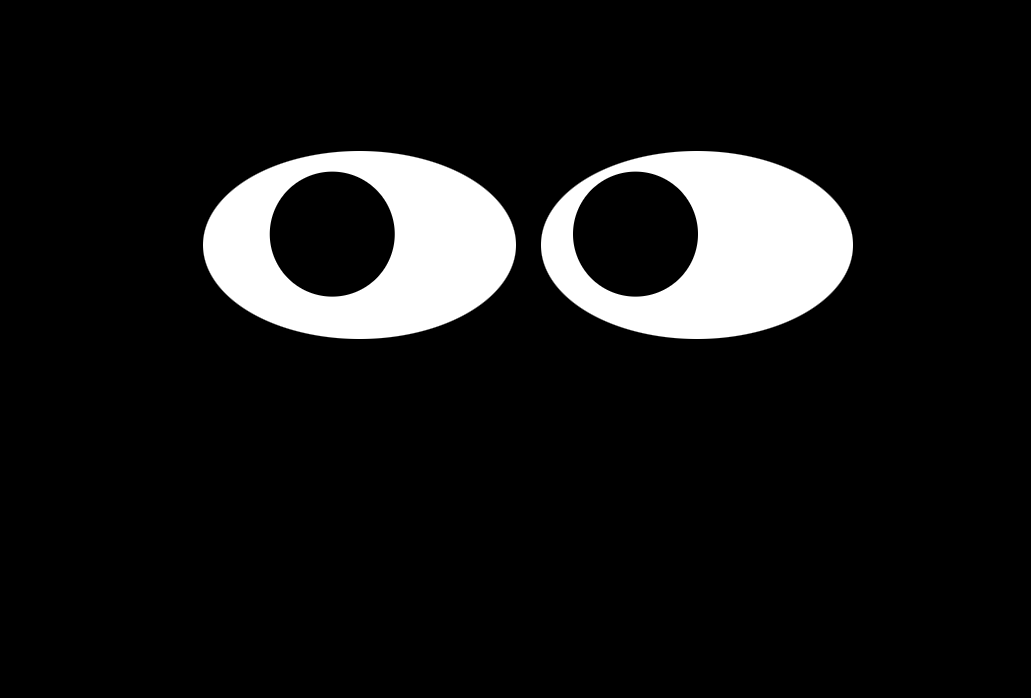
    }

</script>

</body>

</html>

**Output:**



**Program-9**

**Date:**

**Write a Javascript program to create map use the mapboxgl library to add a marker to the map. Then, to move the marker on the map from one bus stop to the next.**

**Index.html**

<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8" />

    <title>Map Animation</title>

    <meta name="viewport" content="initial-scale=1,maximum-scale=1,user-scalable=no" />

    <script src="https://api.mapbox.com/mapbox-gl-js/v1.11.0/mapbox-gl.js"></script>

    <link href="https://api.mapbox.com/mapbox-gl-js/v1.11.0/mapbox-gl.css" rel="stylesheet" />

</head>

<style>

    body {

  margin: 0;

  padding: 0;

}

#map {

  position: absolute;

  top: 0;

  bottom: 0;

  width: 100%;

}

.map-overlay {

  position: absolute;

  left: 0;

  padding: 10px;

}

</style>

<body>

    <div id="map"></div>

    <div class="map-overlay top">

        <button style="font-size: 2em" onclick="move()">

            Show stops between Gandhipuram  and Sulur

        </button>

    </div>

    <script>

const busStops = [

  [76.96836858666605,11.017184003743793],

  [77.01797872410174,11.028641620104397],

  [77.02994960123718,11.00388866436712],

  [ 77.0488302502839,11.004048909680243],

  [ 77.12861237062205,11.031052884109815]

];

// TODO: add your own access token

mapboxgl.accessToken = 'pk.eyJ1Ijoic2VudGhpbGt1cHB1c3dhbXkiLCJhIjoiY2twbTlkMzBlMDQxYjJ2bXcwcG5xdnhzNCJ9.mmPcPQN\_3raRRh8Q0S1kEw';

// This is the map instance

let map = new mapboxgl.Map({

  container: 'map',

  style: 'mapbox://styles/mapbox/streets-v11',

  center: [76.96836858666605,11.017184003743793],

  zoom: 12,

});

var marker= new mapboxgl.Marker()

            .setLngLat([76.96836858666605,11.017184003743793])

            .addTo(map);

let counter = 0;

function move() {

  setTimeout(()=> {

    if (counter>=busStops.length) return;

    marker.setLngLat(busStops[counter]);

    counter++;

    move();

    },1000);

}

// Do not edit code past this point

if (typeof module !== 'undefined') {

  module.exports = { move };

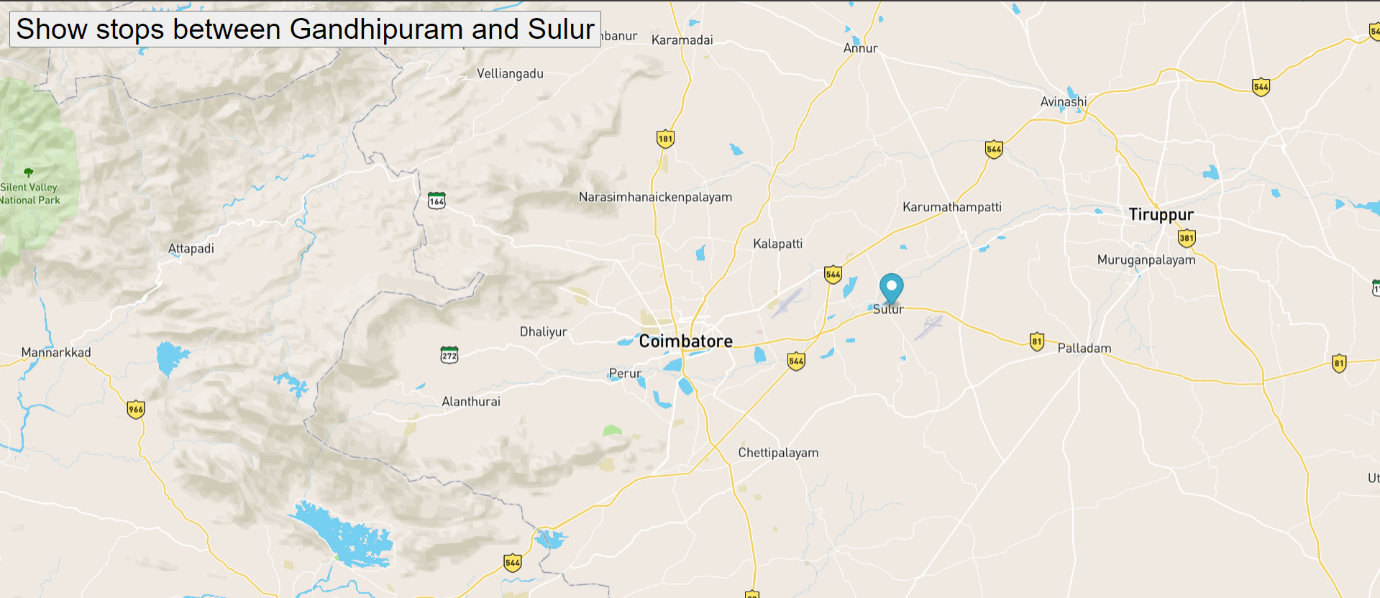
}

    </script>

</body>

</html>

Output:



**Program-10**

**Date:**

**Write a Javascript program to edit the hanoi tower algorithm to add a custom number of disks to the tower using recursion function.**

**Index.hrml**

<html>

<head>

<link rel="stylesheet" href="./style.css" />

<script src="./hanoi.js"></script>

</head>

<body>

<button onclick="makeMove()" id="makemove" width="40">Make Move</button>

<input type="number" id="InputNum">

<div class="game" id="game"></div>

<script src="./tower.js"></script>

</body>

</html>

**hanoi.js**

const writeAll = (pegMap) => {

let s = JSON.stringify(pegMap);

// write output

console.log(s);

pegHist.push(JSON.parse(s));

};

const pegHist = [];

const pegMap = {

A: [],

B: [],

C: [],

};

const pegArray = ['A', 'B', 'C'];

const updateMapAndPlot = (pegMap, from, to) => {

try {

let theDisk = pegMap[from].pop(); // this actually moves the disk

pegMap[to].push(theDisk);

writeAll(pegMap);

} catch (e) {

console.log('no disk in Map');

}

};

// Pegs A, B, C

// disks are labeled 0 to N

let moves = 0;

const moveDisks = function (n, from, to, spare) {

if (n === 1) {

updateMapAndPlot(pegMap, from, to);

moves++;

} else {

moveDisks(n - 1, from, spare, to);

updateMapAndPlot(pegMap, from, to);

moves++;

moveDisks(n - 1, spare, to, from);

}

};

// given from and to, figure out the other peg

const missing = (from, to) => {

const all = {

A: 1,

B: 2,

C: 3,

};

let total = all[from] + all[to];

let miss = 5 - total; // 0-based so not 6 but one less

let keys = Object.keys(all); // array of keys

return keys[miss];

};

const initialize = (nDisks, pegMap) => {

for (let i = nDisks; i >= 1; i--) {

pegMap['A'].push(i); // this stacks disks with the smallest on top

}

writeAll(pegMap);

};

**tower.js**

let counter = 0;

let pegs = [];

let diskArray = []; // this stores the disks

const game = document.getElementById('game');

function plotAll(pegMap) {

pegArray.map((peg, pegindex) => {

// loop over all pegs

let disks = pegMap[peg]; // the array of disks on pegMap.A for example

if (disks.length > 0) {

disks.map((disk, index) => {

let pickDisk = diskArray.filter((item) => {

// pick out correct disk from diskArray

return item.id == disk;

});

positionDisk(pickDisk[0].newdiv, disk, index, pegindex); // disk is the disk number where larger is wider disk

});

}

});

}

function positionDisk(domdiv, diskNumber, indexOnPeg, pegindex) {

// set its position

let pegCenter = 300 \* pegindex + 100;

let diskWidth = diskNumber \* 40 + 20;

let base = 400;

let diskHeight = 20;

domdiv.style.left = pegCenter - diskWidth / 2 + 'px';

domdiv.style.width = diskWidth + 'px';

domdiv.style.top = base - diskHeight \* indexOnPeg + 'px';

console.log('Disk:' + diskNumber + ' at top: ' + domdiv.style.top);

domdiv.style.height = diskHeight + 'px';

domdiv.innerHTML = ' ' + diskNumber;

}

const sleep = (milliseconds) => {

return new Promise((resolve) => setTimeout(resolve, milliseconds));

};

function diskFactory(id, indexOnPeg, pegNumber, type) {

// this determines start positions

let newdiv = document.createElement('div');

positionDisk(newdiv, id, indexOnPeg, pegNumber);

newdiv.setAttribute('class', type); // these are the pegs

newdiv.setAttribute('id', id);

game.appendChild(newdiv);

return {

id,

newdiv,

};

}

function pegFactory(id, indexOnPeg, pegNumber, type) {

// this determines start positions

let newdiv = document.createElement('div');

newdiv.setAttribute('class', 'peg'); // these are the pegs

newdiv.setAttribute('id', id);

let pegCenter = 300 \* pegNumber + 60;

let base = 400;

let diskHeight = 200;

newdiv.style.left = pegCenter + 'px';

newdiv.style.width = '20px';

newdiv.style.top = '200px';

newdiv.style.height = '200px';

game.appendChild(newdiv);

}

function initializeDisks(pegMap) {

let disks = pegMap[pegArray[0]]; // all disks start on peg 0

diskArray = disks.map((diskid, indexOnPeg) => {

// this is a kind of shadow DOM

return diskFactory(diskid, indexOnPeg, 0, 'disk');

});

pegArray.map((item, pegId) => {

pegFactory(pegId, 0, pegId, 'peg');

});

}

function makeMove() {

var InputNum = Number(document.getElementById("InputNum").value)

console.log(InputNum)

if (counter == 0) {

var nDisks = InputNum;

initialize(nDisks, pegMap);

moveDisks(nDisks, 'A', 'C', 'B');

initializeDisks(pegHist[0]);

}

if (counter < pegHist.length) {

plotAll(pegHist[counter]);

} else {

alert('Tower is Finished');

}

counter++;

}

**Output:**

