**Shop-The-Look Lens**

**Assignment**

1. Identifying the White Box in a Black & White Matrix.

Ans : class WhiteBox {

constructor(top, left, height, width) {

this.top = top;

this.left = left;

this.height = height;

this.width = width;

}

}

function findWhiteBox(matrix) {

let top = null;

let left = null;

let height = 0;

let width = 0;

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

for (let j = 0; j < matrix[i].length; j++) {

if (matrix[i][j] === 'w') {

top = i;

left = j;

break;

} }

if (top !== null) {

break;

}

}

if (top === null) {

return null;

}

for (let i = top; i < matrix.length; i++) {

if (matrix[i][left] === 'w') {

height++;

} else {

break;

}

}

for (let j = left; j < matrix[top].length; j++) {

if (matrix[top][j] === 'w') {

width++;

} else {

break;

}

}

return new WhiteBox(top, left, height, width);

}

const inputMatrix = [

['b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b'],

['b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b'],

['b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b'],

['b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b'],

['b', 'b', 'b', 'b', 'w', 'w', 'w', 'w', 'w', 'b', 'b', 'b'],

['b', 'b', 'b', 'b', 'w', 'w', 'w', 'w', 'w', 'b', 'b', 'b'],

['b', 'b', 'b', 'b', 'w', 'w', 'w', 'w', 'w', 'b', 'b', 'b'],

['b', 'b', 'b', 'b', 'w', 'w', 'w', 'w', 'w', 'b', 'b', 'b'],

['b', 'b', 'b', 'b', 'w', 'w', 'w', 'w', 'w', 'b', 'b', 'b'],

['b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b'],

['b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b'],

['b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b', 'b']

];

const whiteBox = findWhiteBox(inputMatrix);

if (whiteBox !== null) {

console.log("Top:", whiteBox.top);

console.log("Left:", whiteBox.left);

console.log("Height:", whiteBox.height);

console.log("Width:", whiteBox.width);

} else {

console.log("No white box found.");

}

Problem 2: Array Division and Exception Handling

Ans:

let arr\_a = [9, 33, 0, 7, 2, 82, 77]

let str\_a = []

for (let i in arr\_a){

if(arr\_a[parseInt(i)+1]!==0){

if(i==arr\_a.length-1){

str\_a.push(parseFloat((arr\_a[i]/arr\_a[0]).toFixed(2)))

}

else {

str\_a.push(parseFloat((arr\_a[i]/arr\_a[parseInt(i)+1]).toFixed(2)))

}

}

else{

str\_a.push("None")

}

}

console.log(str\_a)

Problem 3: Sum of Numbers Divisible by 3 in a String

Ans:

let a ="The best 6 of 8 will get 9 points"

let str\_a = []

let sum = 0

for (let i of a){

if(parseInt(i)%3===0){

str\_a = str\_a+[parseInt(i)]

sum = sum+parseInt(i)

}

}

let length\_a = str\_a.length;

console.log('Sum :',sum)

console.log('Last :', str\_a[length\_a-1])

Problem 4: Last Man Standing in a Circle

Ans:

function lastManWithGun(numPeople) {

let circle = [];

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

circle.push(i);

}

let index = 0;

while (circle.length > 1) {

let killIndex = (index + 1) % circle.length;

circle.splice(killIndex, 1);

index = (killIndex) % circle.length;

}

return circle[0];

}

const numPeople = 100;

const lastMan = lastManWithGun(numPeople);

console.log("The last man with the gun is:", lastMan);

Problem 5: Database Schema for Hotels and Menus

Ans:

CREATE TABLE hotels (

hotel\_id INT PRIMARY KEY,

hotel\_name VARCHAR(300)

);

CREATE TABLE menus (

menu\_id INT PRIMARY KEY,

hotel\_id INT,

menu\_name VARCHAR(300),

FOREIGN KEY (hotel\_id) REFERENCES hotels(hotel\_id)

);

CREATE TABLE food\_items (

food\_item\_id INT PRIMARY KEY,

food\_item\_name VARCHAR(300)

);

CREATE TABLE menu\_food\_items (

menu\_id INT,

food\_item\_id INT,

FOREIGN KEY (menu\_id) REFERENCES menus(menu\_id),

FOREIGN KEY (food\_item\_id) REFERENCES food\_items(food\_item\_id),

PRIMARY KEY (menu\_id, food\_item\_id)

);