**Лабораторна робота №3**

**Завдання 1**

* 1. Розмістіть на сторінці **div**-блок з ідентифікатором **“paragraphs”**. Розмістіть у

ньому 5 абзаців з довільним текстом. Сформуйте масив, який міститиме 5 елементів - кількість символів, які входять до відповідного абзацу тексту.

Для пошуку абзаців використовуйте метод **querySelectorAll** та відповідний

CSS-селектор. Для доступу до тексту, який входить в абзац використовуйте

властивість **innerText** для DOM-елемента. Виведіть у консоль отриманий масив.

**1.2** Дано масив цілих чисел:

let arr = [20, 17, 4, -4, 10, -9, 13, 4, 12, 22, 13, 19, 1, 3];

-  Для заданого масиву визначте та виведіть у консоль:

* найменше та найбільше значення;
* кількість парних та непарних чисел;
* кількість чисел, що містять дві цифри;
* кількість додатних та кількість від’ємних чисел.

- Відсортуйте масив **arr** за допомогою методу Sort у порядку зростання значень та виведіть у консоль результат.

-  Створіть масив з оберненим порядком елементів та виведіть його у консоль.

Лістининг коду

HTML

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Tusk 1</title>

</head>

    <body>

        <div id="paragraphs">

            <p>First</p>

            <p>Second</p>

            <p>Third</p>

            <p>Fourth</p>

            <p>Fifth</p>

        </div>

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

    </body>

</html>

JS

let divArr = document.querySelectorAll('p');

let divTextArr = [];

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

    divTextArr.push(divArr[i].innerText.length);

}

console.log(divTextArr);

let arr = [20, 17, 4, -4, 10,  -9, 13, 4, 12, 22, 13, 19, 1, 3];

//#1 min max

let min = arr[0];

let max = arr[0];

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

    if(max > arr[i]) //Max

        max = arr[i]

    if(min < arr[i]) //min

        min = arr[i]

}

console.log(`1. min ${min}, max ${max}.`)

let pair = 0;

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

    if(arr[i] % 2 == 0)

        pair++;

}

console.log(`2. pair ${pair}, not pair ${arr.length - pair}.`)

let count = 0;

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

    if(Math.abs(arr[i]).toString().length == 2) //use abs to delete "-" from all numbers

        count++

}

console.log(`3. Numbers that have more than one symbol ${count}`)

let countPos = 0;

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

    if(arr[i] >= 0)

        countPos++;

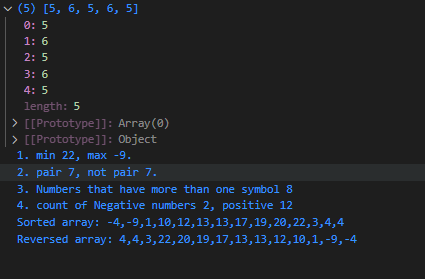
}

console.log(`4. count of Negative numbers ${arr.length - countPos}, positive ${countPos}`);

console.log(`Sorted array: ${arr.sort()}`)

console.log(`Reversed array: ${arr.reverse()}`)

Результат



**Завдання 2**

* Розмістіть на сторінці 3 div-блоки з класами “**table”**. У кожному з них розмістіть
* табличку. У першому блоці - табличку розміром 3х3, другому - 4х4, третьому - 5х5.
* В CSS-файлі розмістіть стиль для класу “**selected**”, який передбачає жовтий фон.
* За допомогою **querySelectorAll** знайдіть **div**-блоки з класом “**table**”.
* Для кожного знайденого блоку у циклі знайдіть всі вкладені у нього теги **td**.
* Для кожного другого тега **td** додайте клас “**selected**”.

Лістининг коду

HTML

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Tusk 2</title>

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

</head>

    <body>

        <div class="table">

            <table>

                <tr>

                    <td></td>

                    <td></td>

                    <td></td>

                </tr>

                <tr>

                    <td></td>

                    <td></td>

                    <td></td>

                </tr>

                <tr>

                    <td></td>

                    <td></td>

                    <td></td>

                </tr>

            </table>

        </div><br>

        <div class="table">

            <table>

                <tr>

                    <td></td>

                    <td></td>

                    <td></td>

                    <td></td>

                </tr>

                <tr>

                    <td></td>

                    <td></td>

                    <td></td>

                    <td></td>

                </tr>

                <tr>

                    <td></td>

                    <td></td>

                    <td></td>

                    <td></td>

                </tr>

                <tr>

                    <td></td>

                    <td></td>

                    <td></td>

                    <td></td>

                </tr>

            </table>

        </div><br>

        <div class="table">

            <table>

                <tr>

                    <td></td>

                    <td></td>

                    <td></td>

                    <td></td>

                    <td></td>

                </tr>

                <tr>

                    <td></td>

                    <td></td>

                    <td></td>

                    <td></td>

                    <td></td>

                </tr>

                <tr>

                    <td></td>

                    <td></td>

                    <td></td>

                    <td></td>

                    <td></td>

                </tr>

                <tr>

                    <td></td>

                    <td></td>

                    <td></td>

                    <td></td>

                    <td></td>

                </tr>

                <tr>

                    <td></td>

                    <td></td>

                    <td></td>

                    <td></td>

                    <td></td>

                </tr>

            </table>

        </div>

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

    </body>

</html>

JS

let div = document.querySelectorAll('td:nth-child(even)');

let j = 0;

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

    div[i].classList.add('selected');

}

CSS

table{

    border-collapse: collapse;

}

table, td{

    border: 1px solid black;

}

td{

    padding: 20px;

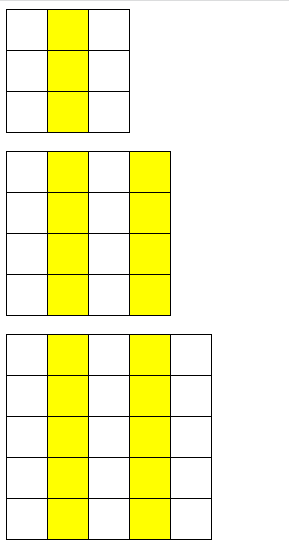
}

.selected {

    background-color: yellow;

}

Результат



**Завдання 3**

Створіть матрицю розміру N на M, яка містить цілі числа (наприклад, 7х5 та числами

на власний розсуд). Знайдіть та виведіть у консоль:

* кількість додатних елементів;
* кількість рядків, які не містять жодного нульового елемента;
* кількість стовпців, які містять хоча б один нульовий елемент;
* номер рядка, в якому знаходиться найдовша серія однакових елементів;
* добуток елементів в тих рядках, які не містять від’ємних елементів;
* суму елементів в тих стовпцях, які не містять від’ємних елементів;
* суму елементів в тих стовпцях, які  містять хоча б один від’ємний елемент;
* транспоновану матрицю;

Лістининг коду

HTML

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Tusk 3</title>

</head>

    <body>

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

    </body>

</html>

JS

let N = 4;

let M = 4;

let arr = [

    [-1, 1, 0, 5],

    [0, 0, 0, 8],

    [-7, 0, -9, 10],

    [1, 3, 3, 1]

];

let countPos = 0;

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

    for(let j = 0; j < M; j++){

        if(arr[i][j] >= 0)

            countPos++;

    }

}

console.log(arr)

console.log(`1. count of positive numbers: ${countPos}`)

let rowCount = 0;

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

    let check = false;

    for(let j = 0; j < M; j++){

        if(arr[i][j] == 0)

            check = true

    }

    if(!check)

        rowCount++;

}

console.log(`2. count of rows without null element: ${rowCount}`)

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

    let check = false;

    for(let j = 0; j < M; j++){

        if(arr[i][j] == 0)

            check = true

    }

    if(!check)

        rowCount++;

}

let countCol = 0;

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

    let check = false;

    for(let j = 0; j < M; j++){

        if(arr[j][i] == 0)

            check = true;

    }

    if(check)

        countCol++;

}

console.log(`3. count of rows with null element: ${countCol}`)

let arrStorage = []

let previouseNum = 0;

let counter = 0;

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

    let storage = [];

    for(let j = 0; j < M; j++){

        let num = arr[i][j];

        for(let l = 0; l < M; l++){

            if(num == arr[i][l] && l != j && previouseNum != num)

                storage.push(arr[i][l])

        }

        previouseNum = num

    }

    previouseNum = NaN;

    if(storage.length != 0)

        arrStorage.push(storage)

}

let iterationIndex = arrStorage[0].length;

let index = 0;

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

    if(arrStorage[i].length > iterationIndex){

        iterationIndex = arrStorage[i].length

        index = i;

    }

}

console.log(`4. number of row that have biggest iteration - ${iterationIndex}, element ${arrStorage[index][0]} & length: ${arrStorage[index].length}`)

console.log(`5. Mult of rows that don't have negative elements.`)

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

    let mult = 1;

    for(let j = 0; j < M; j++){

        if(arr[i][j] < 0){

            mult = NaN

            break;

        }

        mult \*= arr[i][j];

    }

    console.log(`  ${i} - ${mult}`);

}

console.log(`6. Sum of columns that don't have negative elements.`)

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

    let sum = 1;

    for(let j = 0; j < M; j++){

        if(arr[j][i] < 0){

            sum = NaN

            break;

        }

        sum += arr[j][i];

    }

    console.log(`  ${i} - ${sum}`);

}

console.log(`7. Sum of columns that have negative elements.`)

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

    let sum = 1;

    let check = false;

    for(let j = 0; j < M; j++){

        if(arr[j][i] < 0){

            check = true;

            break;

        }

    }

    if(check)

        for(let j = 0; j < M; j++)

            sum += arr[j][i];

    else

        sum = NaN

    console.log(`  ${i} - ${sum}`);

}

function transportion(N, M, arr){

    let newArr = [];

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

        newArr.push([]);

    for(let i = 0; i < N; i++)

        for(let j = 0; j < M; j++)

            newArr[j][i] = arr[i][j];

    return newArr

}

console.log("8. Matrix transportion.")

console.log(transportion(N, M, arr))

Результат

