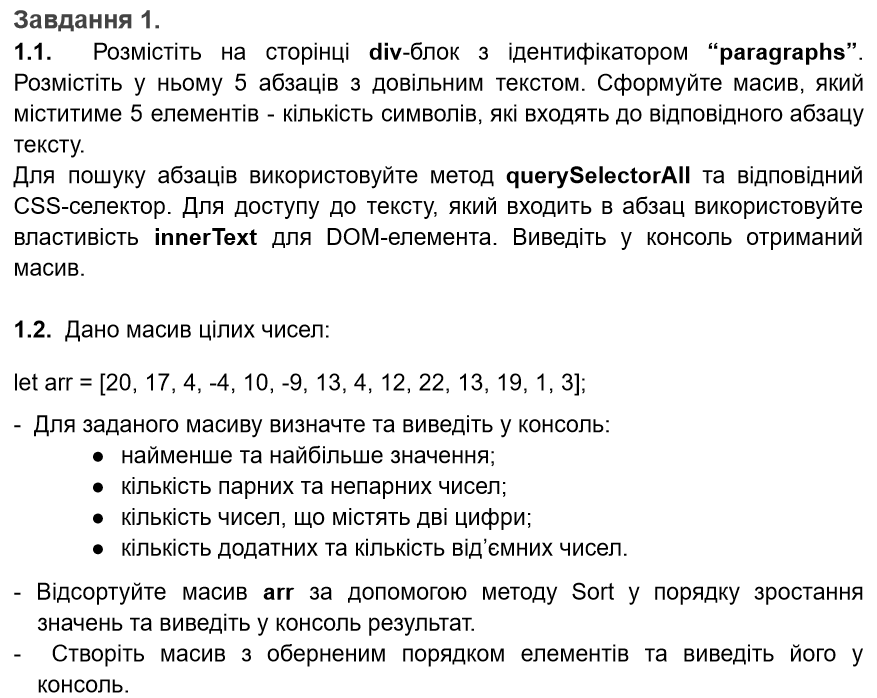
Лабораторна робота №3

Робота з одновимірними та багатовимірними масивами

Мета: навчитися створювати і обробляти одновимірні та багатовимірні масиви та знайомство з операціями над ними.

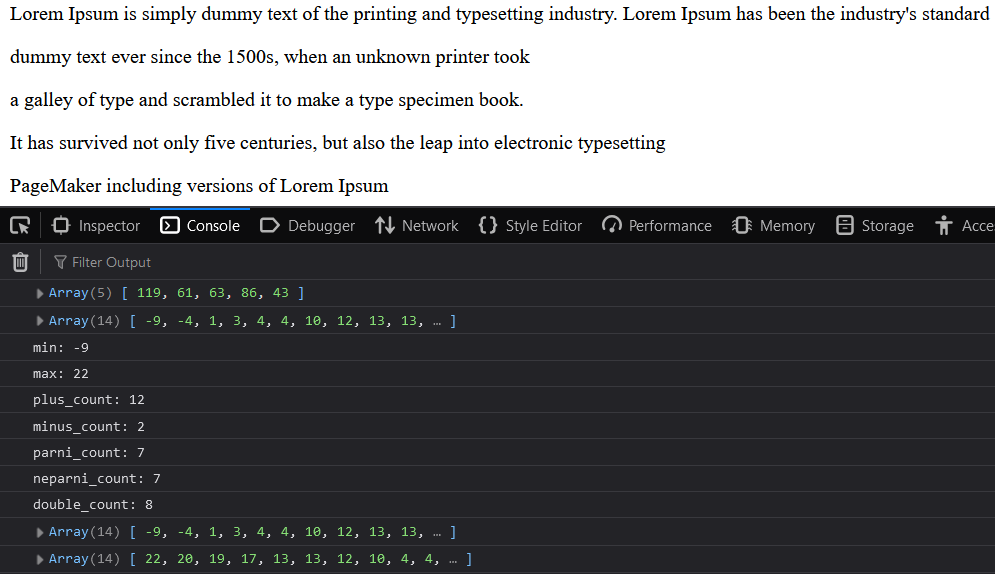
Хід виконання

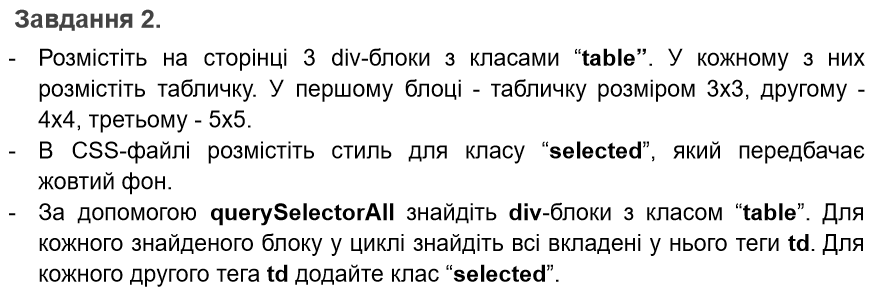


Лістинг index.html:

<!DOCTYPE html>  
<html lang="ua">  
<head>  
 <meta charset="UTF-8">  
 <title>Task 01</title>  
</head>  
<body>  
<div id="paragraphs">  
 <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard </p>  
 <p>dummy text ever since the 1500s, when an unknown printer took </p>  
 <p>a galley of type and scrambled it to make a type specimen book. </p>  
 <p> It has survived not only five centuries, but also the leap into electronic typesetting</p>

<p>PageMaker including versions of Lorem Ipsum</p>  
</div>  
<script>  
 let *div* = *document*.getElementById('paragraphs');  
 let *array* = [];  
 let *paragraphs* = *div*.querySelectorAll('p');  
  
 *paragraphs*.forEach(p => {  
 *array*.push(p.innerText.length);  
 });  
  
 *console*.log(*array*);  
  
 let *arr* = [20, 17, 4, -4, 10, -9, 13, 4, 12, 22, 13, 19, 1, 3];  
 let *min* = *arr*[0];  
 let *max* = *arr*[0];  
 let *plus\_count* = 0;  
 let *minus\_count* = 0;  
 let *parni\_count* = 0;  
 let *neparni\_count* = 0;  
 let *double\_count* = 0;  
  
 for (let num of *arr*) {  
 if (num < *min*) {  
 *min* = num;  
 }  
  
 if (num > *max*) {  
 *max* = num;  
 }  
  
 if (num > 0) {  
 *plus\_count*++;  
 }  
  
 if (num < 0) {  
 *minus\_count*++;  
 }  
  
 if (num % 2 === 0) {  
 *parni\_count*++;  
 } else {  
 *neparni\_count*++;  
 }  
  
 if ((num >= 10 && num <= 99) || (num <= -10 && num >= -99)) {  
 *double\_count*++;  
 }  
 }  
  
 *console*.log(*arr*);  
 *console*.log(`min: ${*min*}`);  
 *console*.log(`max: ${*max*}`);  
 *console*.log(`plus\_count: ${*plus\_count*}`);  
 *console*.log(`minus\_count: ${*minus\_count*}`);  
 *console*.log(`parni\_count: ${*parni\_count*}`);  
 *console*.log(`neparni\_count: ${*neparni\_count*}`);  
 *console*.log(`double\_count: ${*double\_count*}`);  
  
 *arr*.sort((a, b) => a - b);  
 *console*.log(*arr*);  
  
 let *reversedArr* = [...*arr*].reverse();  
 *console*.log(*reversedArr*);  
</script>  
</body>  
</html>

Результат виконання програми

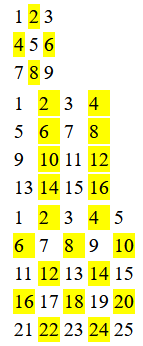


Лістинг style.css:

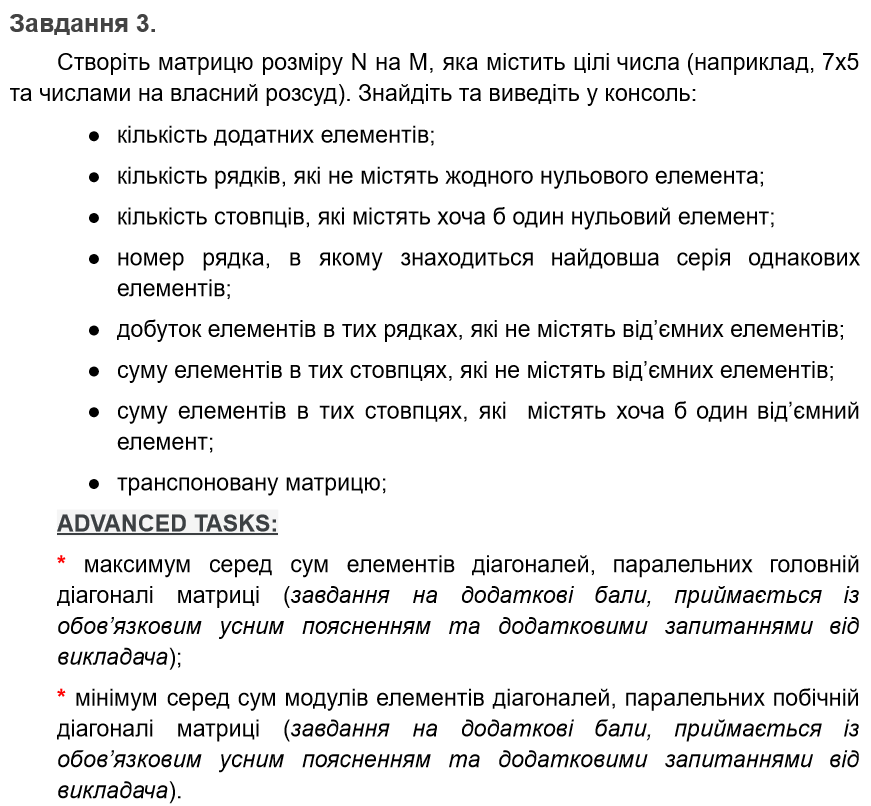
.selected {  
 background-color: yellow;  
}

Лістинг index.html:

<!DOCTYPE html>  
<html lang="ua">  
<head>  
 <meta charset="UTF-8">  
 <title>Task 02</title>  
 <link rel="stylesheet" href="style.css">  
</head>  
<body>  
<div class="table">  
 <table>  
 <tr>  
 <td>1</td>  
 <td>2</td>  
 <td>3</td>  
 </tr>  
 <tr>  
 <td>4</td>  
 <td>5</td>  
 <td>6</td>  
 </tr>  
 <tr>  
 <td>7</td>  
 <td>8</td>  
 <td>9</td>  
 </tr>  
 </table>  
</div>  
  
<div class="table">  
 <table>  
 <tr>  
 <td>1</td>  
 <td>2</td>  
 <td>3</td>  
 <td>4</td>  
 </tr>  
 <tr>  
 <td>5</td>  
 <td>6</td>  
 <td>7</td>  
 <td>8</td>  
 </tr>  
 <tr>  
 <td>9</td>  
 <td>10</td>  
 <td>11</td>  
 <td>12</td>  
 </tr>  
 <tr>  
 <td>13</td>  
 <td>14</td>  
 <td>15</td>  
 <td>16</td>  
 </tr>  
 </table>  
</div>  
  
<div class="table">  
 <table>  
 <tr>  
 <td>1</td>  
 <td>2</td>  
 <td>3</td>  
 <td>4</td>  
 <td>5</td>  
 </tr>  
 <tr>  
 <td>6</td>  
 <td>7</td>  
 <td>8</td>  
 <td>9</td>  
 <td>10</td>  
 </tr>  
 <tr>  
 <td>11</td>  
 <td>12</td>  
 <td>13</td>  
 <td>14</td>  
 <td>15</td>  
 </tr>  
 <tr>  
 <td>16</td>  
 <td>17</td>  
 <td>18</td>  
 <td>19</td>  
 <td>20</td>  
 </tr>  
 <tr>  
 <td>21</td>  
 <td>22</td>  
 <td>23</td>  
 <td>24</td>  
 <td>25</td>  
 </tr>  
 </table>  
</div>  
  
<script>  
 let *tables* = *document*.querySelectorAll('.table');  
  
 *tables*.forEach(table => {  
 let tds = table.querySelectorAll('td');  
 tds.forEach((td, index) => {  
 if ((index + 1) % 2 === 0) {  
 *console*.log(td.innerHTML);  
 td.classList.add('selected');  
 }  
 });  
 });  
</script>  
</body>  
</html>

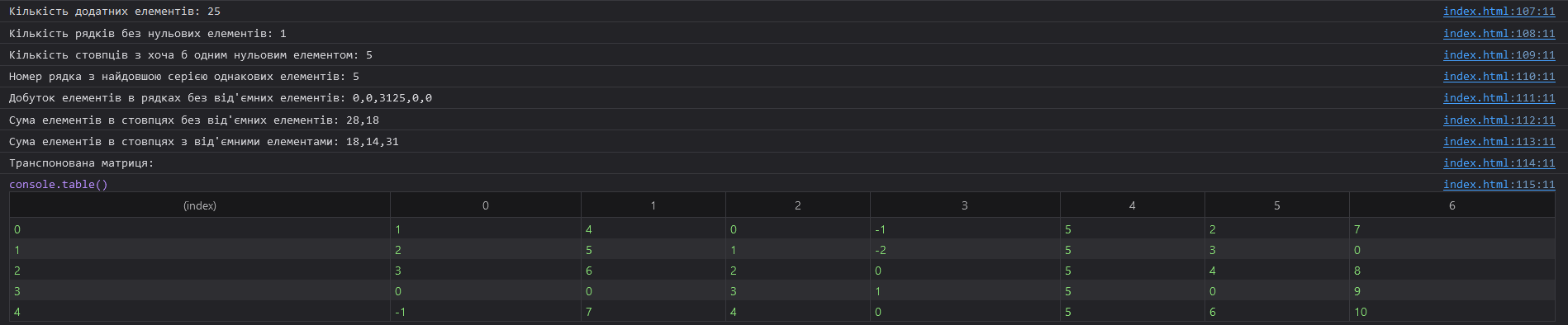


Результат виконання програми



Лістинг програми:

<!DOCTYPE html>  
<html lang="ua">  
<head>  
 <meta charset="UTF-8">  
 <title>Task 03</title>  
</head>  
<body>  
<script>  
 let *matrix* = [  
 [1, 2, 3, 0, -1],  
 [4, 5, 6, 0, 7],  
 [0, 1, 2, 3, 4],  
 [-1, -2, 0, 1, 0],  
 [5, 5, 5, 5, 5],  
 [2, 3, 4, 0, 6],  
 [7, 0, 8, 9, 10]  
 ];  
  
 let *positiveCount* = 0;  
 let *noZeroRowsCount* = 0;  
 let *zeroColsCount* = 0;  
 let *longestSeriesRow* = -1;  
 let *longestSeriesLength* = 0;  
 let *rowProducts* = [];  
 let *colSumPositive* = [];  
 let *colSumNegative* = [];  
 let *transposeMatrix* = [];  
  
 *matrix*.forEach(row => {  
 row.forEach(el => {  
 if (el > 0) *positiveCount*++;  
 });  
 });  
  
 *matrix*.forEach(row => {  
 if (!row.includes(0)) {  
 *noZeroRowsCount*++;  
 }  
 });  
  
 let *colsCount* = *matrix*[0].length;  
 for (let col = 0; col < *colsCount*; col++) {  
 let hasZero = false;  
 for (let row = 0; row < *matrix*.length; row++) {  
 if (*matrix*[row][col] === 0) {  
 hasZero = true;  
 break;  
 }  
 }  
 if (hasZero) *zeroColsCount*++;  
 }  
  
 *matrix*.forEach((row, rowIndex) => {  
 let currentSeriesLength = 1;  
 let maxSeriesLength = 1;  
 for (let i = 1; i < row.length; i++) {  
 if (row[i] === row[i - 1]) {  
 currentSeriesLength++;  
 } else {  
 maxSeriesLength = *Math*.max(maxSeriesLength, currentSeriesLength);  
 currentSeriesLength = 1;  
 }  
 }  
 maxSeriesLength = *Math*.max(maxSeriesLength, currentSeriesLength);  
 if (maxSeriesLength > *longestSeriesLength*) {  
 *longestSeriesLength* = maxSeriesLength;  
 *longestSeriesRow* = rowIndex + 1;  
 }  
 });  
  
 *matrix*.forEach(row => {  
 if (row.every(el => el >= 0)) {  
 let product = row.reduce((acc, el) => acc \* el, 1);  
 *rowProducts*.push(product);  
 }  
 });  
  
 for (let col = 0; col < *colsCount*; col++) {  
 let sumPositive = 0;  
 let sumNegative = 0;  
 let hasNegative = false;  
  
 for (let row = 0; row < *matrix*.length; row++) {  
 if (*matrix*[row][col] >= 0) {  
 sumPositive += *matrix*[row][col];  
 } else {  
 hasNegative = true;  
 sumNegative += *matrix*[row][col];  
 }  
 }  
  
 if (!hasNegative) {  
 *colSumPositive*.push(sumPositive);  
 } else {  
 *colSumNegative*.push(sumNegative + sumPositive);  
 }  
 }  
  
 for (let col = 0; col < *colsCount*; col++) {  
 let newRow = [];  
 for (let row = 0; row < *matrix*.length; row++) {  
 newRow.push(*matrix*[row][col]);  
 }  
 *transposeMatrix*.push(newRow);  
 }  
  
 *console*.log(`Кількість додатних елементів: ${*positiveCount*}`);  
 *console*.log(`Кількість рядків без нульових елементів: ${*noZeroRowsCount*}`);  
 *console*.log(`Кількість стовпців з хоча б одним нульовим елементом: ${*zeroColsCount*}`);  
 *console*.log(`Номер рядка з найдовшою серією однакових елементів: ${*longestSeriesRow*}`);  
 *console*.log(`Добуток елементів в рядках без від'ємних елементів: ${*rowProducts*}`);  
 *console*.log(`Сума елементів в стовпцях без від'ємних елементів: ${*colSumPositive*}`);  
 *console*.log(`Сума елементів в стовпцях з від'ємними елементами: ${*colSumNegative*}`);  
 *console*.log('Транспонована матриця:');  
 *console*.table(*transposeMatrix*);  
</script>  
</body>  
</html>

Результат виконання програми

Висновок: навчився створювати і обробляти одновимірні та багатовимірні масиви та ознайомився з операціями над ними.