Лабораторна робота №1

**Тема:** Введення в JavaScript розробку. Розгалуження, цикли, масиви. Фунції.

**Мета:** Навчитись працювати з базовими елементами JavaScript.

Хід роботи

**Варіант 9**

1. Дано три цілих числа, знайти середнє з них. Середнім назвемо число, яке більше найменшого з даних чисел, але менше максимального.
2. Визначити кількість натуральних тризначних чисел, сума цифр яких дорівнює заданому числу N.
3. Введений рядок вивести у зворотньому вигляді( 'йцукен' - 'некуцй')
4. Напишіть функцію фалідації телефона.  
   Пробіли, дужки, мінуси, можуть бути відсутні.  
     
   +380 - обов'язкові цифри, приймаються телефони лише України, 12 цифр.  
   +380682091234  
   +38068 209-12-34  
   +3 8 (068) 209 12 34  
   +38(068)209-12-34 і т.п.
5. Дано одновимірний масив. Переставити в зворотному порядку елементи масиву, розташовані між мінімальним і максимальними елементами.
6. Починаючи з k-го стовпця, зрушити їх вперед, а перші k поставити на місце останніх

**Код програми**

**lw1.js**

// Laboratory Work 1  
  
// Task 1: Find a middle number out of three entered.  
**function** taskUno() {  
  
 **let** \_values = [];  
  
 [...document.getElementsByClassName('task1-input')].forEach(item => {  
 \_values.push(item.value);  
 });  
  
 \_values.forEach(value => {  
 **if** (value > Math.min(...\_values) && value < Math.max(...\_values))  
 alert('Middle number is ... ' + value);  
 });  
  
}  
  
// Task 2: Find the count of character sums of tri-sized numbers that equals an input  
**function** taskDuo(target = parseInt(document.getElementById('task2-input').value), size = 900, startAt = 100) {  
  
 **let** \_range = [...Array(size).keys()].map(i => i + startAt);  
 **let** \_count = 0;  
  
 \_range.forEach(value => {  
 **let** strVal = value.toString(),  
 numbers = [];  
 **for** (**let** i = 0; i < strVal.length; i++) {  
 numbers.push(parseInt(strVal.charAt(i)));  
 }  
 **if** (numbers.reduce((a, b) => a + b, 0) === target) {  
 console.log(value);  
 \_count++;  
 }  
 });  
  
 alert('Result is ... ' + \_count);  
  
}  
  
// Task 3: Inverse the string  
**function** taskTres(inputStr = document.getElementById('task3-input').value) {  
  
 **let** \_reversed = [...inputStr].reverse().join('');  
 alert(\_reversed);  
  
}  
  
// Task 4: Phone number validation  
**function** taskQuadro(inputStr = document.getElementById('task4-input').value) {  
  
 **let** \_regex = /^\+38(\()?0[0-9]{2}(\))?[0-9]{3}[-. ]?[0-9]{2}[-. ]?[0-9]{2}$/;  
 **if** (inputStr.match(\_regex)) {  
 alert('Zer Gut!');  
 } **else** {  
 alert('Bad stuff m8!');  
 }  
  
}  
  
// Task 5: Reverse array elements between min and max element  
**function** findMin(iterable) {  
  
 **let** minVal = iterable[0],  
 minPos = 0;  
 **for** (**let** i = 0; i < iterable.length; i++) {  
 **if** (iterable[i] < minVal) {  
 minVal = iterable[i];  
 minPos = i;  
 }  
 }  
 **return** [minVal, minPos];  
  
}  
  
**function** findMax(iterable) {  
  
 **let** maxVal = iterable[0],  
 maxPos = 0;  
 **for** (**let** i = 0; i < iterable.length; i++) {  
 **if** (iterable[i] > maxVal) {  
 maxVal = iterable[i];  
 maxPos = i;  
 }  
 }  
 **return** [maxVal, maxPos];  
  
}  
  
  
**function** taskPenta() {  
  
 **let** \_array = [],  
 \_length = Math.random() \* 10 + 1,  
 \_output = document.getElementById('task5-output');  
 \_output.innerHTML = '';  
  
 **for** (**let** i = 0; i < \_length; i++) {  
 \_array.push(Math.random() \* 100);  
 }  
 \_output.innerHTML += 'Before: ';  
 \_array.forEach(value => {  
 \_output.innerHTML += Math.round(value) + ' ';  
 });  
  
 **let** minPos = findMin(\_array)[1],  
 maxPos = findMax(\_array)[1],  
 left = (minPos < maxPos ? minPos : maxPos) + 1,  
 right = minPos > maxPos ? minPos : maxPos,  
 \_reversed = \_array.slice(left, right).reverse();  
  
 **for** (**let** i = left, z = 0; i < right; i++, z++) {  
 \_array[i] = \_reversed[z];  
 }  
 \_output.innerHTML += '<br>After: ';  
 \_array.forEach(value => {  
 \_output.innerHTML += Math.round(value) + ' ';  
 });  
}  
  
// Task 6 Switch first and last column(2) and shift the columns(1)  
**function** taskSex(beginCol = parseInt(document.getElementById('task6-input').value), shift = parseInt(document.getElementById('task6-input2').value), \_rows = parseInt(document.getElementById('task6-rows').value), \_columns = 10) {  
  
 **let** \_array = [],  
 \_output = document.getElementById('task6-output');  
 \_output.innerHTML = '';  
  
 **for** (**let** i = 0; i < \_rows; i++) {  
 **let** \_subArray = [];  
 **for** (**let** z = 0; z < \_columns; z++) {  
 \_subArray.push(Math.round(Math.random() \* 100));  
 }  
 \_array.push(\_subArray);  
 }  
 \_output.innerHTML += 'Before: <br>';  
 \_array.forEach(value => {  
 value.forEach(value2 => \_output.innerHTML += value2 + ' ');  
 \_output.innerHTML += '<br>';  
 });  
   
 // Here we SHIFT  
 **let** \_shifted = [];  
 **for** (**let** i = 0; i < \_rows; i++) {  
 **let** \_subShifted = Array(beginCol);  
 **for** (**let** z = 0; z < beginCol; z++) {  
 \_subShifted[z] = \_array[i][z];  
 }  
  
 **let** \_cut = Array(\_columns - beginCol),  
 \_arrayCut = [];  
 **for** (**let** z = beginCol; z < \_columns; z++) {  
 \_arrayCut.push(\_array[i][z]);  
 }  
 **for** (**let** z = 0; z < \_cut.length; z++) {  
 **let** getId = (z - shift < 0) ? (\_cut.length - Math.abs((z - shift))) : (z - shift);  
 \_cut[z] = \_arrayCut[getId];  
 }  
 \_subShifted = \_subShifted.concat(\_cut);  
 \_shifted.push(\_subShifted);  
 }  
 \_output.innerHTML += '<br>Shift:<br>';  
 \_shifted.forEach(value => {  
 value.forEach(value2 => \_output.innerHTML += value2 + ' ');  
 \_output.innerHTML += '<br>';  
 });  
  
 //Here we SWAP  
 **for** (**let** i = 0; i < \_rows; i++) {  
 **let** temp = \_shifted[i][beginCol];  
 \_shifted[i][beginCol] = \_shifted[i][\_columns - 1];  
 \_shifted[i][\_columns - 1] = temp;  
 }  
 \_output.innerHTML += '<br>Swap:<br>';  
 \_shifted.forEach(value => {  
 value.forEach(value2 => \_output.innerHTML += value2 + ' ');  
 \_output.innerHTML += '<br>'; });  
}

GitHub Repo. :

<https://github.com/MiraculousNightingale/human-machine-interaction/tree/lw1>

**Зображення**

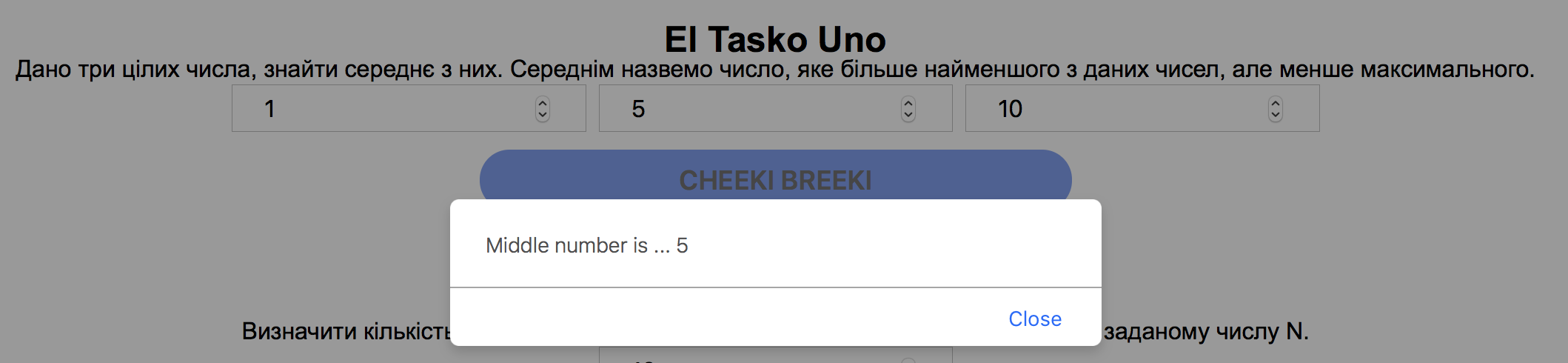


Рисунок 1. Завдання 1

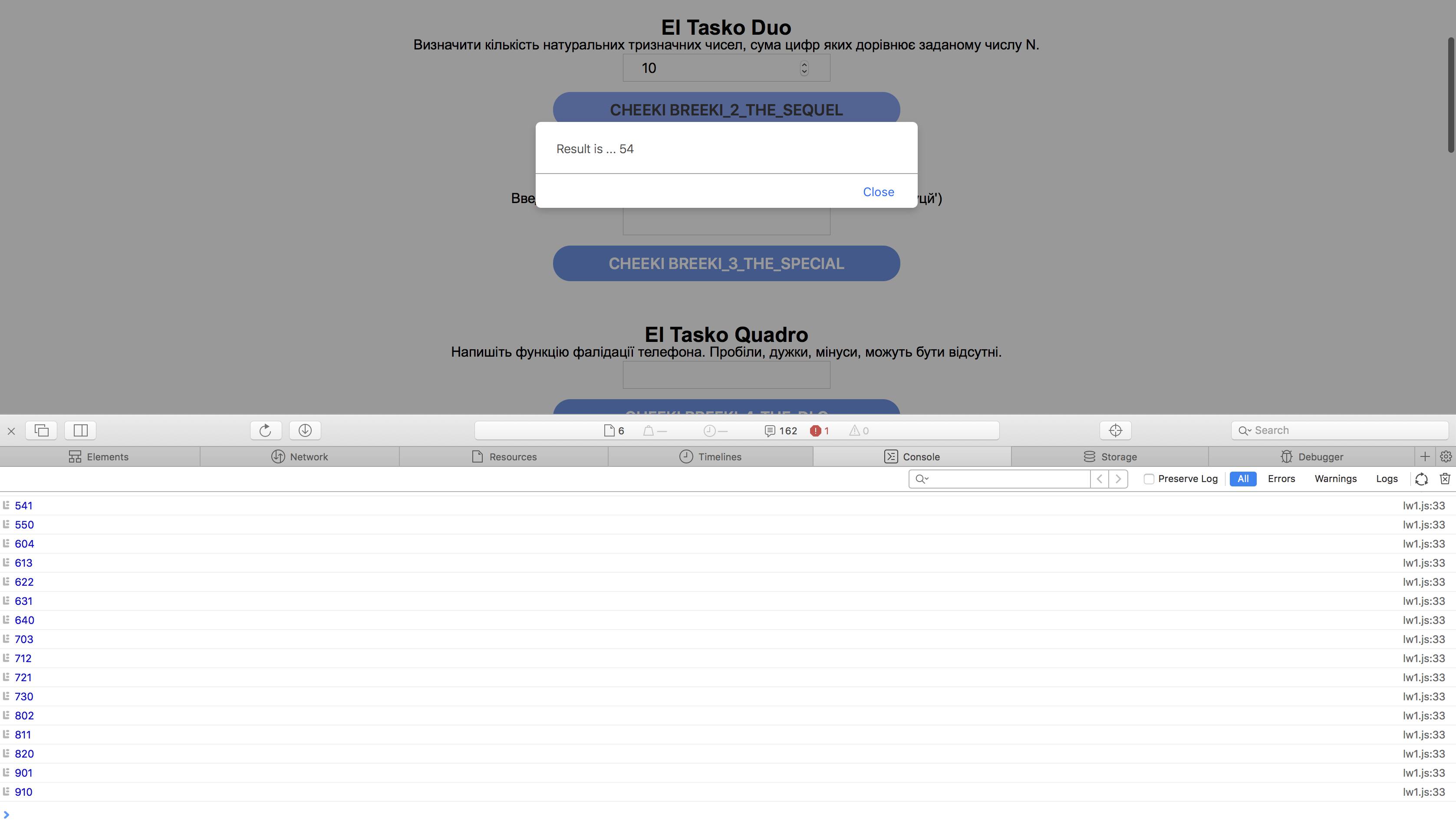


Рисунок 2. Завдання 2

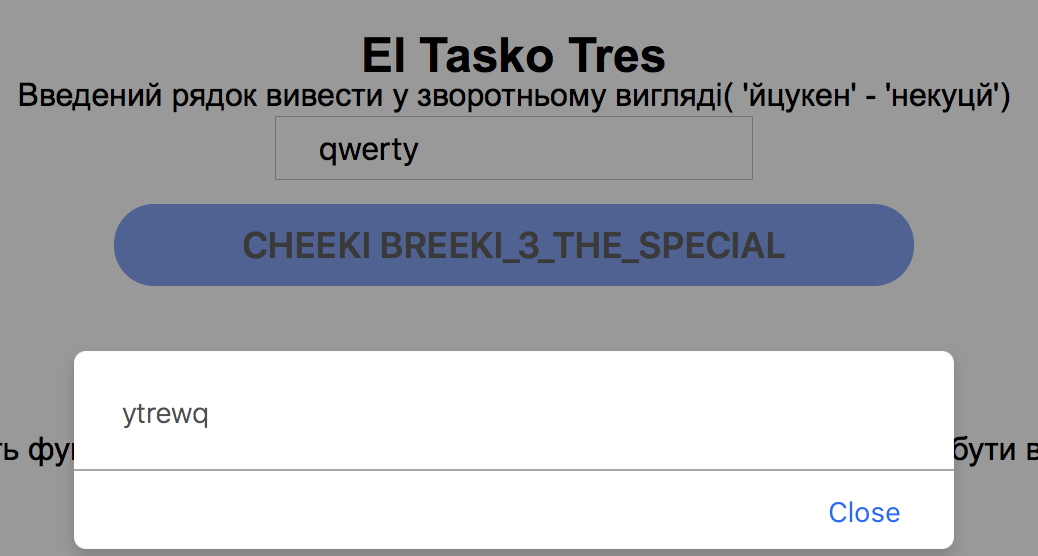


Рисунок 3. Завдання 3

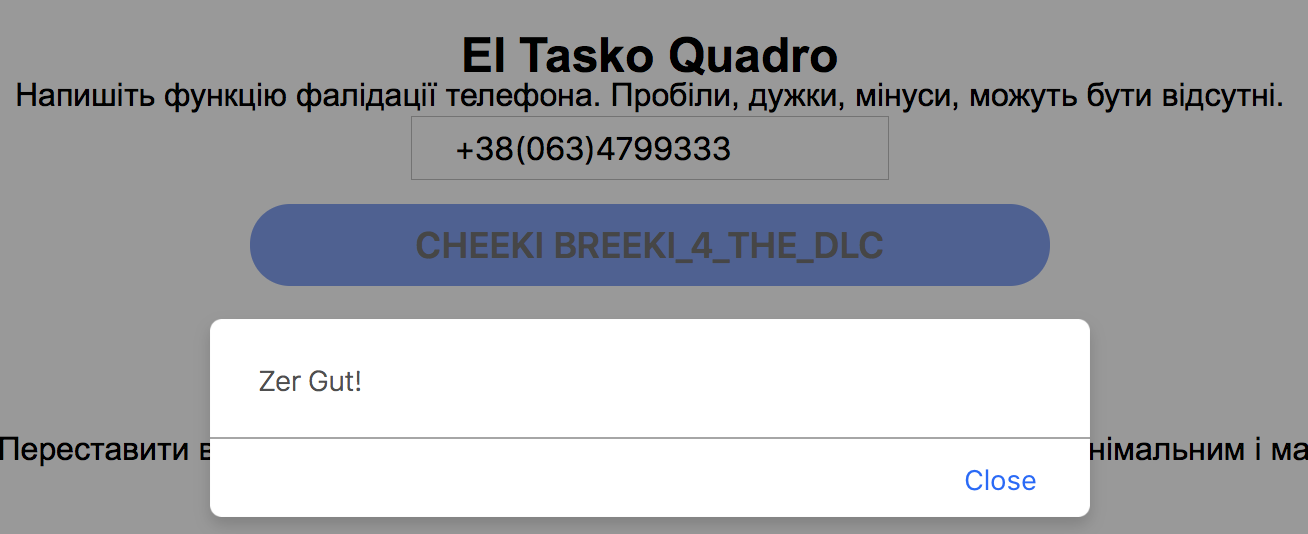


Рисунок 4. Завдання 4

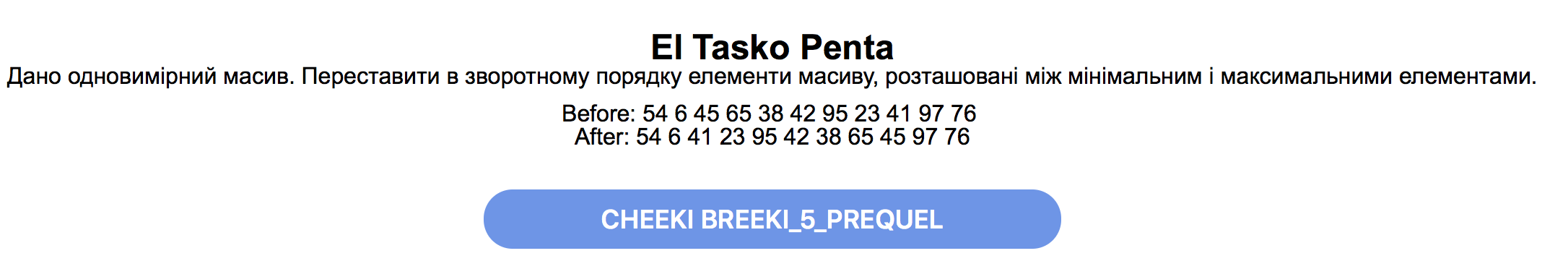


Рисунок 5. Завдання 5

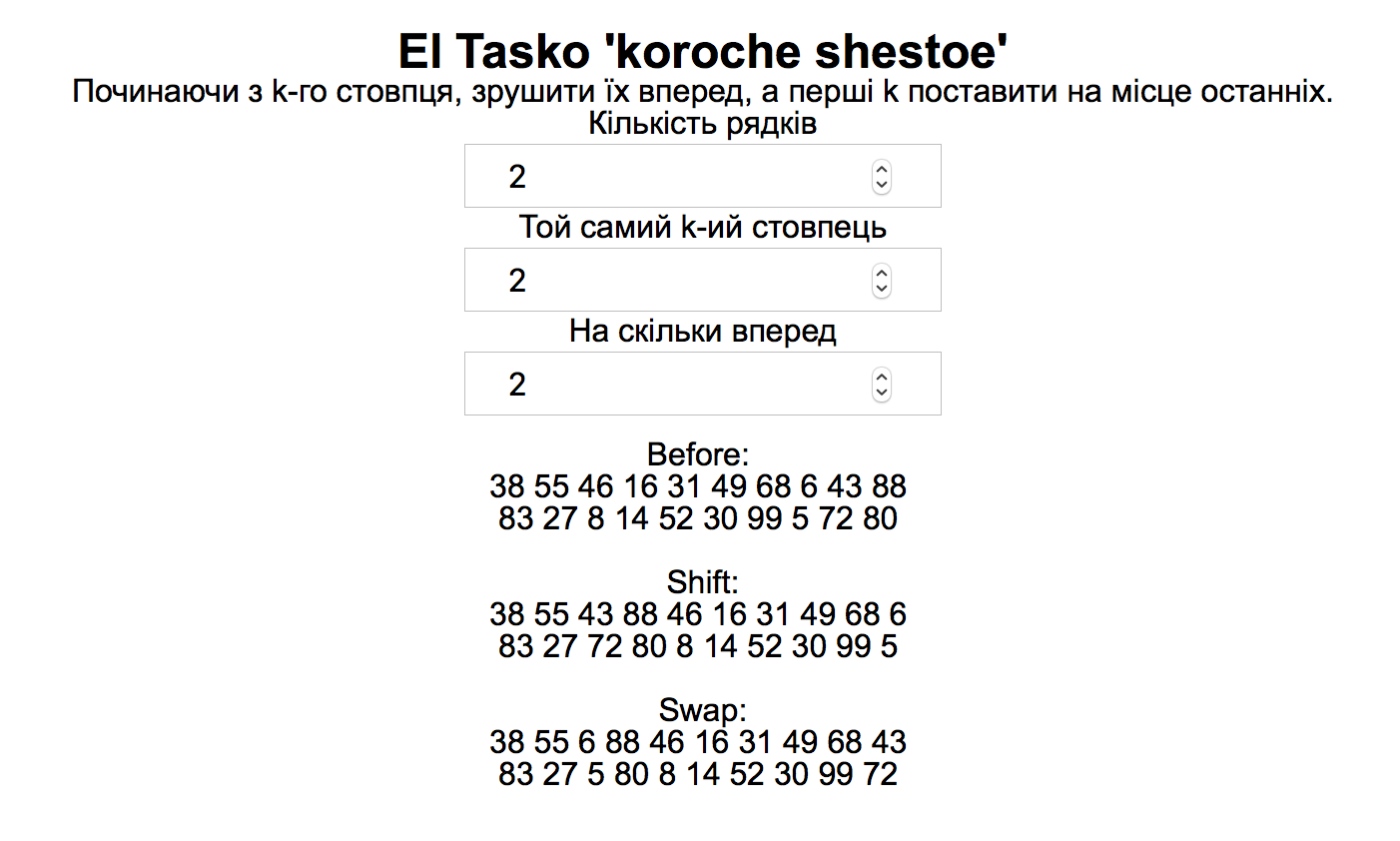


Рисунок 6. Завдання 6

**Висновок:** на даній лабораторній роботі я навчився працювати з базовими елементами JavaScript.