Міністерство освіти і науки України

Національний технічний університет "ХПІ"

кафедра "Стратегічного управління"

**РОЗРАХУНКОВО-** **ГРАФІЧНЕ ЗАВДАННЯ**

з дисципліни "Основи Web-програмування*"*

**Тема**

"Порівняльний аналіз методів обробки масивів засобами мови JavaScript"

|  |  |
| --- | --- |
|  | Керівник РГЗ  Шуба І.В. |
|  | Виконавець  Черніков С.О. |

Харків 2022

Table of Contents

[1. ОСНОВА HTML ДОКУМЕНТУ 5](#_Toc119235100)

[1.1 HTML код 5](#_Toc119235101)

[1.2 CSS код 5](#_Toc119235102)

[2 ГОЛОВНІ JAVASCRIPT СКРИПТИ 6](#_Toc119235103)

[2.1 Функція пошуку суми елементів масиву 6](#_Toc119235104)

[2.1.1 Стандартна функція 6](#_Toc119235105)

[2.1.2 Кастомна функція 6](#_Toc119235106)

[2.2 Функція піднесення до квадрату кожного елементу 7](#_Toc119235107)

[2.2.1 Стандартна функція 7](#_Toc119235108)

[2.2.2 Кастомна функція 7](#_Toc119235109)

[2.3 Функція фільтрації масиву (тільки позитивні елементи) 8](#_Toc119235110)

[2.3.1 Стандартна функція 8](#_Toc119235111)

[2.3.2 Кастомна функція 8](#_Toc119235112)

[2.4 Функція перевірки масиву (містить хоча б один негативний елемент, всі елементи негативні) 9](#_Toc119235113)

[2.4.1 Стандартна функція 9](#_Toc119235114)

[2.4.2 Кастомна функція 9](#_Toc119235115)

[2.5 Функція пошуку першого негативного елементу 10](#_Toc119235116)

[2.5.1 Стандартна функція 10](#_Toc119235117)

[2.5.1 Кастомна функція 10](#_Toc119235118)

[2.6 Функція сортування масиву (Insertion Sort Algorithm) 11](#_Toc119235119)

[2.6.1 Стандартна функція 11](#_Toc119235120)

[2.6.2 Кастомна функція 11](#_Toc119235121)

[3 ДОПОМІЖНІ СКРИПТИ 12](#_Toc119235122)

[Код 12](#_Toc119235123)

[РЕЗУЛЬТАТ ВИКОНАННЯ 44](#_Toc119235124)

ПОСТАНОВКА ЗАДАЧІ

Організувати створення дев’яти (100, 200, 500, 1000, 5000, 10000, 25000, 50000, 100000) – елементних випадкових масив згідно до параметрів власного варіанту.



Порівняти час розрахунку статистичного параметру згідно до свого варіанту для кожного масиву, створеному у першому завданні, за допомогою ітераційних методів масивів та стандартних засобів (циклів) JS. Якщо параметр не визначається для речових значень, скористатися їх цілою частиною.



Порівняти час модифікації (див. власний варіант) для кожного масиву, створеному у першому завданні, за допомогою ітераційних методів масивів та стандартних засобів (циклів) JS. Якщо модифікація не реалізується для речових значень, скористатися їх цілою частиною.



Порівняти час фільтрації (див. власний варіант) для кожного масиву, створеному у першому завданні, за допомогою ітераційних методів масивів та стандартних засобів (циклів) JS. Якщо фільтрація не реалізується для речових значень, скористатися їх цілою частиною.



Порівняти час перевірки (див. власний варіант) для кожного масиву, створеному у першому завданні, за допомогою ітераційних методів масивів та стандартних засобів (циклів) JS. Якщо перевірка не реалізується для речових значень, скористатися їх цілою частиною.



Порівняти час пошуку (див. власний варіант) для кожного масиву, створеному у першому завданні, за допомогою ітераційних методів масивів та стандартних засобів (циклів) JS. Якщо перевірка не реалізується для речових значень, скористатися їх цілою частиною.



Порівняти час сортування (непарні номери – за зростанням, парні – за зменшенням) для кожного масиву, створеному у першому завданні, за допомогою ітераційних методів масивів та стандартних засобів сортування (наприклад сортування «бульбашкою» чи вставками) JS.

Розробити скрипт чи скрипти для проведення експериментів щодо визначення часових характеристик методів обробки масивів з попередніх завдань. Для підвищення точності для кожного методу та способу його реалізації треба проводити декілька (не менш ніж 10, а чим більше тим краще) замірів часу з подальшим розрахунком середнього значення.

# ОСНОВА HTML ДОКУМЕНТУ

## 1.1 HTML код

<!DOCTYPE html>  
<html>  
<head>  
 <title>Home</title>  
  
 <link rel="stylesheet" href="style.css">  
</head>  
<body>  
 <h3>Author of work - Sergey Chernikov, CS321a</h3>  
 <button type="button" name="button" onclick="start()">Start test</button>  
 <table id="tableWithResults">  
 <tr>  
 <th>Function</th>  
 <th>100 items</th>  
 <th>200 items</th>  
 <th>500 items</th>  
 <th>1000 items</th>  
 <th>5000 items</th>  
 <th>10000 items</th>  
 <th>25000 items</th>  
 <th>50000 items</th>  
 <th>100000 items</th>  
 </tr>  
 </table>  
 <script src="script.js"></script>  
</body>  
</html>

## 1.2 CSS код

body {  
 font-family: monospace, sans-serif;  
}  
  
#tableWithResults {  
 margin-top: 5%  
}  
  
table, tr, td, th {  
 border: 1px solid black ;  
 padding: 10px 15px;  
 border-collapse: collapse;  
}  
  
button {  
 padding: 10px 15px;  
 background: #58D68D;  
 color: black;  
 font-family: Montserrat, sans-serif;  
 border-radius: 10%;  
 border: 1px green solid;  
}  
  
button:hover {  
 background: #145A32;  
}

# 2 ГОЛОВНІ JAVASCRIPT СКРИПТИ

## 2.1 Функція пошуку суми елементів масиву

### 2.1.1 Стандартна функція

function calculateSumOfElementsStandard(array) {  
 let result = 0;  
 array.forEach((item) => {  
 result += item;  
 });  
 return result;  
}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Кількість елементів у масиві | | | | | | | | | |
| 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Середній час виконання | <0.10ms. | <0.10ms. | <0.10ms. | <0.10ms. | 0.20ms. | 0.30ms. | 0.60ms. | 0.40ms. | 0.70ms. |

### 2.1.2 Кастомна функція

function calculateSumOfElements(array) {  
 let sum = 0;  
 for (let i = 0; i < array.length; i++) {  
 sum += array[i];  
 }  
 return sum;  
}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Кількість елементів у масиві | | | | | | | | | |
| 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Середній час виконання | <0.10ms. | <0.10ms. | 0.1 ms. | <0.10ms. | 0.10ms. | 0.20ms. | 0.50ms. | 0.90ms. | 1.90ms. |

## 2.2 Функція піднесення до квадрату кожного елементу

### 2.2.1 Стандартна функція

function standardDoublyAllElements(array) {  
 array.forEach((item, index) => array[index] = item\*\*2);  
 return array;  
}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Кількість елементів у масиві | | | | | | | | | |
| 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Середній час виконання | <0.10ms. | <0.10ms. | 0.1 ms. | <0.10ms. | 0.70ms. | 1.70ms. | 2.10ms. | 4.00ms. | 9.20ms. |

### 2.2.2 Кастомна функція

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Кількість елементів у масиві | | | | | | | | | |
| 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Середній час виконання | <0.10ms. | <0.10ms. | 0.10 ms. | 0.10ms | 0.40ms. | 0.80ms. | 0.90ms. | 0.70ms. | 0.60ms. |

function doublyAllElements(array) {  
 for (let i = 0; i < array.length; i++) {  
 array[i] = ***Math***.pow(array[i], 2);  
 }  
  
 return array;  
}

## 2.3 Функція фільтрації масиву (тільки позитивні елементи)

### 2.3.1 Стандартна функція

function standardPositiveFilter(array) {  
 let result = array.filter(item => item > 0);  
 return result;  
}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Кількість елементів у масиві | | | | | | | | | |
| 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Середній час виконання | <0.10ms. | <0.10ms. | 0.1 ms. | <0.10ms. | 0.10ms. | 0.30ms. | 1.40ms. | 1.70ms. | 3.60ms. |

### 2.3.2 Кастомна функція

function filterPositiveElemenents(array) {  
 let positiveArray = [];  
 for (let i = 0; i < array.length; i++) {  
 if (array[i] > 0) {  
 positiveArray.push(array[i]);  
 }  
 }  
  
 return positiveArray;  
}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Кількість елементів у масиві | | | | | | | | | |
| 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Середній час виконання | 0.10ms. | <0.10ms. | 0.1 ms. | 0.10ms. | 0.30ms. | 0.70ms. | 0.90ms. | 1.30ms. | 1.20ms. |

## 2.4 Функція перевірки масиву (містить хоча б один негативний елемент, всі елементи негативні)

### 2.4.1 Стандартна функція

function standardCheck(array) {  
 const isNegative = (currentValue) => currentValue < 0;  
 let allNegative = array.every(isNegative);  
 let oneNegative = array.some(item => item < 0) == true;  
  
 return {  
 allNegative: allNegative,  
 oneNegative: oneNegative  
 };  
}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Кількість елементів у масиві | | | | | | | | | |
| 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Середній час виконання | 0.10ms. | <0.10ms. | <0.1 ms. | <0.10ms. | <0.10ms.. | <0.10ms. | <0.10ms. | <0.10ms. | <0.10ms. |

### 2.4.2 Кастомна функція

function checkArrayElements(array) {  
 let containsAtLeastOneNegativeElement = false;  
 let allElementsAreNegative = false;  
 let info = {};  
  
 let numberOfNegativeElements = 0;  
 for (let i = 0; i < array.length; i++) {  
 if (array[i] < 0) {  
 numberOfNegativeElements++;  
 containsAtLeastOneNegativeElement = true;  
 }  
 }  
  
 if (numberOfNegativeElements == array.length) allElementsAreNegative = true;  
  
 info.data = {  
 atLeastOneNegative: containsAtLeastOneNegativeElement,  
 allNegative: allElementsAreNegative  
 }  
  
 return info;  
}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Кількість елементів у масиві | | | | | | | | | |
| 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Середній час виконання | 0.10ms. | <0.10ms. | <0.1 ms. | <0.10ms. | 0.20ms. | 0.50ms. | 1.30ms. | 0.30ms. | 0.60ms. |

## 2.5 Функція пошуку першого негативного елементу

### 2.5.1 Стандартна функція

function standardFindFirstNegative(array) {  
 return array.find(item => item < 0);  
}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Кількість елементів у масиві | | | | | | | | | |
| 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Середній час виконання | <0.10ms. | <0.10ms. | <0.1 ms. | <0.10ms. | <0.10ms. | <0.10ms. | <0.10ms. | <0.10ms. | <0.10ms. |

### 2.5.1 Кастомна функція

function findFirstNegativeElement(array) {  
 for (let i = 0; i < array.length; i++) {  
 if (array[i] < 0) {  
 return {  
 firstNegativeValue: array[i],  
 index: i  
 };  
 }  
 }  
}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Кількість елементів у масиві | | | | | | | | | |
| 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Середній час виконання | <0.10ms. | <0.10ms. | <0.1 ms. | <0.10ms. | <0.10ms. | <0.10ms. | <0.10ms. | 0.10ms. | <0.10ms. |

## 2.6 Функція сортування масиву (Insertion Sort Algorithm)

### 2.6.1 Стандартна функція

function sortArrayStandard(array) {  
 array.sort(function(a, b){return a - b});  
}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Кількість елементів у масиві | | | | | | | | | |
| 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Середній час виконання | 0.10ms. | <0.10ms. | 0.30 ms. | 0.40ms. | 2.00ms. | 3.30ms. | 8.70ms. | 27.50ms. | 58.70ms. |

### 2.6.2 Кастомна функція

function insertionSort(inputArr) {  
 let n = inputArr.length;  
 for (let i = 1; i < n; i++) {  
 // Choosing the first element in our unsorted subarray  
 let current = inputArr[i];  
 // The last element of our sorted subarray  
 let j = i-1;  
 while ((j > -1) && (current < inputArr[j])) {  
 inputArr[j+1] = inputArr[j];  
 j--;  
 }  
 inputArr[j+1] = current;  
 }  
 return inputArr;  
}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Кількість елементів у масиві | | | | | | | | | |
| 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Середній час виконання | 0.20ms. | 0.50ms. | 1.80 ms. | 0.30ms. | 8.10ms. | 30.30ms. | 184.30ms. | 751.30ms. | 2947.70ms. |

# 3 ДОПОМІЖНІ СКРИПТИ

## Код

// constants  
const ***MIN\_ELEMENT\_VALUE*** = -100;  
const ***MAX\_ELEMENT\_VALUE*** = 100;  
  
// HTML elements  
let ***table*** = ***document***.getElementById('tableWithResults');  
  
// arrays  
let ***array100*** = [];  
let ***array200*** = [];  
let ***array500*** = [];  
let ***array1000*** = [];  
let ***array5000*** = [];  
let ***array10000*** = [];  
let ***array25000*** = [];  
let ***array50000*** = [];  
let ***array100000*** = [];  
  
*/\*\*  
 \* Custom functions for arrays  
 \*/****console***.log('CUSTOM FUNCTIONS');  
  
// filling the arrays  
function fillArrayRandomly(array, size) {  
 let difference = ***MAX\_ELEMENT\_VALUE*** - ***MIN\_ELEMENT\_VALUE***;  
 let randomNumber;  
  
 if (array instanceof ***Array***) {  
 for (let i = 0; i < size; i++) {  
 randomNumber = ***Math***.random();  
 randomNumber = ***Math***.floor(randomNumber \* difference);  
 randomNumber = randomNumber + ***MIN\_ELEMENT\_VALUE***;  
 array[i] = randomNumber;  
 }  
 }  
}  
  
function fillArrays() {  
 fillArrayRandomly(***array100***, 100);  
 fillArrayRandomly(***array200***, 200);  
 fillArrayRandomly(***array500***, 500);  
 fillArrayRandomly(***array1000***, 1000);  
 fillArrayRandomly(***array5000***, 5000);  
 fillArrayRandomly(***array10000***, 10000);  
 fillArrayRandomly(***array25000***, 25000);  
 fillArrayRandomly(***array50000***, 50000);  
 fillArrayRandomly(***array100000***, 100000);  
}  
  
function calculateSumOfElements(array) {  
 let sum = 0;  
 for (let i = 0; i < array.length; i++) {  
 sum += array[i];  
 }  
 return sum;  
}  
  
function calculateSumOfEachArray() {  
 let startTime;  
 let endTime;  
 let info = {};  
  
 startTime = ***performance***.now();  
 let sum100 = calculateSumOfElements(***array100***);  
 endTime = ***performance***.now();  
 info.data100 = {  
 sum: sum100,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let sum200 = calculateSumOfElements(***array200***);  
 endTime = ***performance***.now();  
 info.data200 = {  
 sum: sum200,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let sum500 = calculateSumOfElements(***array500***);  
 endTime = ***performance***.now();  
 info.data500 = {  
 sum: sum500,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let sum1000 = calculateSumOfElements(***array1000***);  
 endTime = ***performance***.now();  
 info.data1000 = {  
 sum: sum1000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let sum5000 = calculateSumOfElements(***array5000***);  
 endTime = ***performance***.now();  
 info.data5000 = {  
 sum: sum5000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let sum10000 = calculateSumOfElements(***array10000***);  
 endTime = ***performance***.now();  
 info.data10000 = {  
 sum: sum10000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let sum25000 = calculateSumOfElements(***array25000***);  
 endTime = ***performance***.now();  
 info.data25000 = {  
 sum: sum25000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let sum50000 = calculateSumOfElements(***array50000***);  
 endTime = ***performance***.now();  
 info.data50000 = {  
 sum: sum50000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let sum100000 = calculateSumOfElements(***array100000***);  
 endTime = ***performance***.now();  
 info.data100000 = {  
 sum: sum100000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 addRowToTableCustomSum(info);  
 return info;  
}  
  
function addRowToTableCustomSum(info) {  
 let row = ***document***.createElement('tr');  
 row.style.background = 'green';  
  
 let tdFunctionName = ***document***.createElement('td');  
 tdFunctionName.innerHTML = "Custom Sum";  
  
 let td100 = ***document***.createElement('td');  
 td100.innerHTML = info.data100.timeTaken + 'ms.';  
  
 let td200 = ***document***.createElement('td');  
 td200.innerHTML = info.data200.timeTaken + 'ms.';  
  
 let td500 = ***document***.createElement('td');  
 td500.innerHTML = info.data500.timeTaken + 'ms.';  
  
 let td1000 = ***document***.createElement('td');  
 td1000.innerHTML = info.data1000.timeTaken + 'ms.';  
  
 let td5000 = ***document***.createElement('td');  
 td5000.innerHTML = info.data5000.timeTaken + 'ms.';  
  
 let td10000 = ***document***.createElement('td');  
 td10000.innerHTML = info.data10000.timeTaken + 'ms.';  
  
 let td25000 = ***document***.createElement('td');  
 td25000.innerHTML = info.data25000.timeTaken + 'ms.';  
  
 let td50000 = ***document***.createElement('td');  
 td50000.innerHTML = info.data50000.timeTaken + 'ms.';  
  
 let td100000 = ***document***.createElement('td');  
 td100000.innerHTML = info.data100000.timeTaken + 'ms.';  
  
 row.appendChild(tdFunctionName);  
 row.appendChild(td100);  
 row.appendChild(td200);  
 row.appendChild(td500);  
 row.appendChild(td1000);  
 row.appendChild(td5000);  
 row.appendChild(td10000);  
 row.appendChild(td25000);  
 row.appendChild(td50000);  
 row.appendChild(td100000);  
  
 ***table***.appendChild(row);  
}  
  
function doublyAllElements(array) {  
 for (let i = 0; i < array.length; i++) {  
 array[i] = ***Math***.pow(array[i], 2);  
 }  
  
 return array;  
}  
  
function doublyEachArray() {  
 let startTime;  
 let endTime;  
 let info = {};  
  
 startTime = ***performance***.now();  
 let doubly100 = doublyAllElements(***array100***);  
 endTime = ***performance***.now();  
 info.data100 = {  
 array: doubly100,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let doubly200 = doublyAllElements(***array200***);  
 endTime = ***performance***.now();  
 info.data200 = {  
 array: doubly200,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let doubly500 = doublyAllElements(***array500***);  
 endTime = ***performance***.now();  
 info.data500 = {  
 array: doubly500,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let doubly1000 = doublyAllElements(***array1000***);  
 endTime = ***performance***.now();  
 info.data1000 = {  
 array: doubly1000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let doubly5000 = doublyAllElements(***array5000***);  
 endTime = ***performance***.now();  
 info.data5000 = {  
 array: doubly5000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let doubly10000 = doublyAllElements(***array10000***);  
 endTime = ***performance***.now();  
 info.data10000 = {  
 array: doubly10000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let doubly25000 = doublyAllElements(***array25000***);  
 endTime = ***performance***.now();  
 info.data25000 = {  
 array: doubly25000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let doubly50000 = doublyAllElements(***array50000***);  
 endTime = ***performance***.now();  
 info.data50000 = {  
 array: doubly50000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let doubly100000 = doublyAllElements(***array100000***);  
 endTime = ***performance***.now();  
 info.data100000 = {  
 array: doubly100000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 addRowToTableCustomDouble(info);  
 return info;  
}  
  
function addRowToTableCustomDouble(info) {  
 let row = ***document***.createElement('tr');  
 row.style.background = 'yellow';  
  
 let tdFunctionName = ***document***.createElement('td');  
 tdFunctionName.innerHTML = "Double All Elements";  
  
 let td100 = ***document***.createElement('td');  
 td100.innerHTML = info.data100.timeTaken + 'ms.';  
  
 let td200 = ***document***.createElement('td');  
 td200.innerHTML = info.data200.timeTaken + 'ms.';  
  
 let td500 = ***document***.createElement('td');  
 td500.innerHTML = info.data500.timeTaken + 'ms.';  
  
 let td1000 = ***document***.createElement('td');  
 td1000.innerHTML = info.data1000.timeTaken + 'ms.';  
  
 let td5000 = ***document***.createElement('td');  
 td5000.innerHTML = info.data5000.timeTaken + 'ms.';  
  
 let td10000 = ***document***.createElement('td');  
 td10000.innerHTML = info.data10000.timeTaken + 'ms.';  
  
 let td25000 = ***document***.createElement('td');  
 td25000.innerHTML = info.data25000.timeTaken + 'ms.';  
  
 let td50000 = ***document***.createElement('td');  
 td50000.innerHTML = info.data50000.timeTaken + 'ms.';  
  
 let td100000 = ***document***.createElement('td');  
 td100000.innerHTML = info.data100000.timeTaken + 'ms.';  
  
 row.appendChild(tdFunctionName);  
 row.appendChild(td100);  
 row.appendChild(td200);  
 row.appendChild(td500);  
 row.appendChild(td1000);  
 row.appendChild(td5000);  
 row.appendChild(td10000);  
 row.appendChild(td25000);  
 row.appendChild(td50000);  
 row.appendChild(td100000);  
  
 ***table***.appendChild(row);  
}  
  
function filterPositiveElemenents(array) {  
 let positiveArray = [];  
 for (let i = 0; i < array.length; i++) {  
 if (array[i] > 0) {  
 positiveArray.push(array[i]);  
 }  
 }  
  
 return positiveArray;  
}  
  
function filterAllArrays() {  
 let startTime;  
 let endTime;  
 let info = {};  
  
 startTime = ***performance***.now();  
 let positiveArray100 = filterPositiveElemenents(***array100***);  
 endTime = ***performance***.now();  
 info.data100 = {  
 positiveArray: positiveArray100,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let positiveArray200 = filterPositiveElemenents(***array200***);  
 endTime = ***performance***.now();  
 info.data200 = {  
 positiveArray: positiveArray200,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let positiveArray500 = filterPositiveElemenents(***array500***);  
 endTime = ***performance***.now();  
 info.data500 = {  
 positiveArray: positiveArray500,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let positiveArray1000 = filterPositiveElemenents(***array1000***);  
 endTime = ***performance***.now();  
 info.data1000 = {  
 positiveArray: positiveArray1000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let positiveArray5000 = filterPositiveElemenents(***array5000***);  
 endTime = ***performance***.now();  
 info.data5000 = {  
 positiveArray: positiveArray5000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let positiveArray10000 = filterPositiveElemenents(***array10000***);  
 endTime = ***performance***.now();  
 info.data10000 = {  
 positiveArray: positiveArray10000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let positiveArray25000 = filterPositiveElemenents(***array25000***);  
 endTime = ***performance***.now();  
 info.data25000 = {  
 positiveArray: positiveArray25000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let positiveArray50000 = filterPositiveElemenents(***array50000***);  
 endTime = ***performance***.now();  
 info.data50000 = {  
 positiveArray: positiveArray50000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let positiveArray100000 = filterPositiveElemenents(***array100000***);  
 endTime = ***performance***.now();  
 info.data100000 = {  
 positiveArray: positiveArray100000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 addRowToTableCustomFilter(info);  
 return info;  
}  
  
function addRowToTableCustomFilter(info) {  
 let row = ***document***.createElement('tr');  
 row.style.background = 'lightblue';  
  
 let tdFunctionName = ***document***.createElement('td');  
 tdFunctionName.innerHTML = "Custom Positive Filter";  
  
 let td100 = ***document***.createElement('td');  
 td100.innerHTML = info.data100.timeTaken + 'ms.';  
  
 let td200 = ***document***.createElement('td');  
 td200.innerHTML = info.data200.timeTaken + 'ms.';  
  
 let td500 = ***document***.createElement('td');  
 td500.innerHTML = info.data500.timeTaken + 'ms.';  
  
 let td1000 = ***document***.createElement('td');  
 td1000.innerHTML = info.data1000.timeTaken + 'ms.';  
  
 let td5000 = ***document***.createElement('td');  
 td5000.innerHTML = info.data5000.timeTaken + 'ms.';  
  
 let td10000 = ***document***.createElement('td');  
 td10000.innerHTML = info.data10000.timeTaken + 'ms.';  
  
 let td25000 = ***document***.createElement('td');  
 td25000.innerHTML = info.data25000.timeTaken + 'ms.';  
  
 let td50000 = ***document***.createElement('td');  
 td50000.innerHTML = info.data50000.timeTaken + 'ms.';  
  
 let td100000 = ***document***.createElement('td');  
 td100000.innerHTML = info.data100000.timeTaken + 'ms.';  
  
 row.appendChild(tdFunctionName);  
 row.appendChild(td100);  
 row.appendChild(td200);  
 row.appendChild(td500);  
 row.appendChild(td1000);  
 row.appendChild(td5000);  
 row.appendChild(td10000);  
 row.appendChild(td25000);  
 row.appendChild(td50000);  
 row.appendChild(td100000);  
  
 ***table***.appendChild(row);  
}  
  
function checkArrayElements(array) {  
 let containsAtLeastOneNegativeElement = false;  
 let allElementsAreNegative = false;  
 let info = {};  
  
 let numberOfNegativeElements = 0;  
 for (let i = 0; i < array.length; i++) {  
 if (array[i] < 0) {  
 numberOfNegativeElements++;  
 containsAtLeastOneNegativeElement = true;  
 }  
 }  
  
 if (numberOfNegativeElements == array.length) allElementsAreNegative = true;  
  
 info.data = {  
 atLeastOneNegative: containsAtLeastOneNegativeElement,  
 allNegative: allElementsAreNegative  
 }  
  
 return info;  
}  
  
function checkAllArrays() {  
 let startTime;  
 let endTime;  
 let info = {};  
  
 startTime = ***performance***.now();  
 let check100 = checkArrayElements(***array100***);  
 endTime = ***performance***.now();  
 info.data100 = {  
 data: check100,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let check200 = checkArrayElements(***array200***);  
 endTime = ***performance***.now();  
 info.data200 = {  
 data: check200,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let check500 = checkArrayElements(***array500***);  
 endTime = ***performance***.now();  
 info.data500 = {  
 data: check500,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let check1000 = checkArrayElements(***array1000***);  
 endTime = ***performance***.now();  
 info.data1000 = {  
 data: check1000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let check5000 = checkArrayElements(***array5000***);  
 endTime = ***performance***.now();  
 info.data5000 = {  
 data: check5000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let check10000 = checkArrayElements(***array10000***);  
 endTime = ***performance***.now();  
 info.data10000 = {  
 data: check10000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let check25000 = checkArrayElements(***array25000***);  
 endTime = ***performance***.now();  
 info.data25000 = {  
 data: check25000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let check50000 = checkArrayElements(***array50000***);  
 endTime = ***performance***.now();  
 info.data50000 = {  
 data: check50000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let check100000 = checkArrayElements(***array100000***);  
 endTime = ***performance***.now();  
 info.data100000 = {  
 data: check100000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 addRowToTableCustomCheck(info);  
 return info;  
}  
  
function addRowToTableCustomCheck(info) {  
 let row = ***document***.createElement('tr');  
 row.style.background = 'orange';  
  
 let tdFunctionName = ***document***.createElement('td');  
 tdFunctionName.innerHTML = "Custom Check For Negatives";  
  
 let td100 = ***document***.createElement('td');  
 td100.innerHTML = info.data100.timeTaken + 'ms.';  
  
 let td200 = ***document***.createElement('td');  
 td200.innerHTML = info.data200.timeTaken + 'ms.';  
  
 let td500 = ***document***.createElement('td');  
 td500.innerHTML = info.data500.timeTaken + 'ms.';  
  
 let td1000 = ***document***.createElement('td');  
 td1000.innerHTML = info.data1000.timeTaken + 'ms.';  
  
 let td5000 = ***document***.createElement('td');  
 td5000.innerHTML = info.data5000.timeTaken + 'ms.';  
  
 let td10000 = ***document***.createElement('td');  
 td10000.innerHTML = info.data10000.timeTaken + 'ms.';  
  
 let td25000 = ***document***.createElement('td');  
 td25000.innerHTML = info.data25000.timeTaken + 'ms.';  
  
 let td50000 = ***document***.createElement('td');  
 td50000.innerHTML = info.data50000.timeTaken + 'ms.';  
  
 let td100000 = ***document***.createElement('td');  
 td100000.innerHTML = info.data100000.timeTaken + 'ms.';  
  
 row.appendChild(tdFunctionName);  
 row.appendChild(td100);  
 row.appendChild(td200);  
 row.appendChild(td500);  
 row.appendChild(td1000);  
 row.appendChild(td5000);  
 row.appendChild(td10000);  
 row.appendChild(td25000);  
 row.appendChild(td50000);  
 row.appendChild(td100000);  
  
 ***table***.appendChild(row);  
}  
  
function findFirstNegativeElement(array) {  
 for (let i = 0; i < array.length; i++) {  
 if (array[i] < 0) {  
 return {  
 firstNegativeValue: array[i],  
 index: i  
 };  
 }  
 }  
}  
  
function findAllFirstNegativeElements() {  
 let startTime;  
 let endTime;  
 let info = {};  
  
 startTime = ***performance***.now();  
 let find100 = findFirstNegativeElement(***array100***);  
 endTime = ***performance***.now();  
 info.find100 = {  
 data: find100,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let find200 = findFirstNegativeElement(***array200***);  
 endTime = ***performance***.now();  
 info.find200 = {  
 data: find200,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let find500 = findFirstNegativeElement(***array500***);  
 endTime = ***performance***.now();  
 info.find500 = {  
 data: find500,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let find1000 = findFirstNegativeElement(***array1000***);  
 endTime = ***performance***.now();  
 info.find1000 = {  
 data: find1000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let find5000 = findFirstNegativeElement(***array5000***);  
 endTime = ***performance***.now();  
 info.find5000 = {  
 data: find5000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let find10000 = findFirstNegativeElement(***array10000***);  
 endTime = ***performance***.now();  
 info.find10000 = {  
 data: find10000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let find25000 = findFirstNegativeElement(***array25000***);  
 endTime = ***performance***.now();  
 info.find25000 = {  
 data: find25000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let find50000 = findFirstNegativeElement(***array50000***);  
 endTime = ***performance***.now();  
 info.find50000 = {  
 data: find50000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let find100000 = findFirstNegativeElement(***array100000***);  
 endTime = ***performance***.now();  
 info.find100000 = {  
 data: find100000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 addRowToTableCustomSearch(info);  
 return info;  
}  
  
function addRowToTableCustomSearch(info) {  
 let row = ***document***.createElement('tr');  
 row.style.background = 'gray';  
  
 let tdFunctionName = ***document***.createElement('td');  
 tdFunctionName.innerHTML = "Custom First Negative Search";  
  
 let td100 = ***document***.createElement('td');  
 td100.innerHTML = info.find100.timeTaken + 'ms.';  
  
 let td200 = ***document***.createElement('td');  
 td200.innerHTML = info.find200.timeTaken + 'ms.';  
  
 let td500 = ***document***.createElement('td');  
 td500.innerHTML = info.find500.timeTaken + 'ms.';  
  
 let td1000 = ***document***.createElement('td');  
 td1000.innerHTML = info.find1000.timeTaken + 'ms.';  
  
 let td5000 = ***document***.createElement('td');  
 td5000.innerHTML = info.find5000.timeTaken + 'ms.';  
  
 let td10000 = ***document***.createElement('td');  
 td10000.innerHTML = info.find10000.timeTaken + 'ms.';  
  
 let td25000 = ***document***.createElement('td');  
 td25000.innerHTML = info.find25000.timeTaken + 'ms.';  
  
 let td50000 = ***document***.createElement('td');  
 td50000.innerHTML = info.find50000.timeTaken + 'ms.';  
  
 let td100000 = ***document***.createElement('td');  
 td100000.innerHTML = info.find100000.timeTaken + 'ms.';  
  
 row.appendChild(tdFunctionName);  
 row.appendChild(td100);  
 row.appendChild(td200);  
 row.appendChild(td500);  
 row.appendChild(td1000);  
 row.appendChild(td5000);  
 row.appendChild(td10000);  
 row.appendChild(td25000);  
 row.appendChild(td50000);  
 row.appendChild(td100000);  
  
 ***table***.appendChild(row);  
}  
  
function insertionSort(inputArr) {  
 let n = inputArr.length;  
 for (let i = 1; i < n; i++) {  
 // Choosing the first element in our unsorted subarray  
 let current = inputArr[i];  
 // The last element of our sorted subarray  
 let j = i-1;  
 while ((j > -1) && (current < inputArr[j])) {  
 inputArr[j+1] = inputArr[j];  
 j--;  
 }  
 inputArr[j+1] = current;  
 }  
 return inputArr;  
}  
  
function sortEachArray() {  
 let startTime;  
 let endTime;  
 let info = {};  
  
 startTime = ***performance***.now();  
 let sort100 = insertionSort(***array100***);  
 endTime = ***performance***.now();  
 info.insertionSort100 = {  
 sortedArray: sort100,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let sort200 = insertionSort(***array200***);  
 endTime = ***performance***.now();  
 info.insertionSort200 = {  
 sortedArray: sort200,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let sort500 = insertionSort(***array500***);  
 endTime = ***performance***.now();  
 info.insertionSort500 = {  
 sortedArray: sort500,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let sort1000 = insertionSort(***array1000***);  
 endTime = ***performance***.now();  
 info.insertionSort1000 = {  
 sortedArray: sort1000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let sort5000 = insertionSort(***array5000***);  
 endTime = ***performance***.now();  
 info.insertionSort5000 = {  
 sortedArray: sort5000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let sort10000 = insertionSort(***array10000***);  
 endTime = ***performance***.now();  
 info.insertionSort10000 = {  
 sortedArray: sort10000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let sort25000 = insertionSort(***array25000***);  
 endTime = ***performance***.now();  
 info.insertionSort25000 = {  
 sortedArray: sort25000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let sort50000 = insertionSort(***array50000***);  
 endTime = ***performance***.now();  
 info.insertionSort50000 = {  
 sortedArray: sort50000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let sort100000 = insertionSort(***array100000***);  
 endTime = ***performance***.now();  
 info.insertionSort100000 = {  
 sortedArray: sort100000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 addRowToTableInsertionSort(info);  
 return info;  
}  
  
function addRowToTableInsertionSort(info) {  
 let row = ***document***.createElement('tr');  
 row.style.background = 'gold';  
  
 let tdFunctionName = ***document***.createElement('td');  
 tdFunctionName.innerHTML = "Insertion sort";  
  
 let td100 = ***document***.createElement('td');  
 td100.innerHTML = info.insertionSort100.timeTaken + 'ms.';  
  
 let td200 = ***document***.createElement('td');  
 td200.innerHTML = info.insertionSort200.timeTaken + 'ms.';  
  
 let td500 = ***document***.createElement('td');  
 td500.innerHTML = info.insertionSort500.timeTaken + 'ms.';  
  
 let td1000 = ***document***.createElement('td');  
 td1000.innerHTML = info.insertionSort1000.timeTaken + 'ms.';  
  
 let td5000 = ***document***.createElement('td');  
 td5000.innerHTML = info.insertionSort5000.timeTaken + 'ms.';  
  
 let td10000 = ***document***.createElement('td');  
 td10000.innerHTML = info.insertionSort10000.timeTaken + 'ms.';  
  
 let td25000 = ***document***.createElement('td');  
 td25000.innerHTML = info.insertionSort25000.timeTaken + 'ms.';  
  
 let td50000 = ***document***.createElement('td');  
 td50000.innerHTML = info.insertionSort50000.timeTaken + 'ms.';  
  
 let td100000 = ***document***.createElement('td');  
 td100000.innerHTML = info.insertionSort100000.timeTaken + 'ms.';  
  
 row.appendChild(tdFunctionName);  
 row.appendChild(td100);  
 row.appendChild(td200);  
 row.appendChild(td500);  
 row.appendChild(td1000);  
 row.appendChild(td5000);  
 row.appendChild(td10000);  
 row.appendChild(td25000);  
 row.appendChild(td50000);  
 row.appendChild(td100000);  
  
 ***table***.appendChild(row);  
}  
  
function startCustomTest() {  
 // fillArrays();  
 // let info1 = calculateSumOfEachArray();  
 // console.log("Sum of all elements:");  
 // console.log(info1);  
 // fillArrays();  
 // let info2 = doublyEachArray();  
 // console.log("Doubling all the elements:");  
 // console.log(info2);  
 // fillArrays();  
 // let info3 = filterAllArrays();  
 // console.log("Filtering all arrays:");  
 // console.log(info3);  
 // fillArrays();  
 // let info4 = checkAllArrays();  
 // console.log("Check of all arrays:");  
 // console.log(info4);  
 // fillArrays();  
 // let info5 = findAllFirstNegativeElements();  
 // console.log("Searching for all negative elements:");  
 // console.log(info5);  
 // fillArrays();  
 // let info6 = sortEachArray();  
 // console.log("Insertion Sort for each array:");  
 // console.log(info6);  
}  
  
// startCustomTest();  
  
***console***.log('STANDARD FUNCTIONS');  
  
function sortArrayStandard(array) {  
 array.sort(function(a, b){return a - b});  
}  
  
function sortAllArraysStandard() {  
 let startTime;  
 let endTime;  
 let info = {};  
  
 startTime = ***performance***.now();  
 sortArrayStandard(***array100***);  
 endTime = ***performance***.now();  
 info.standardSort100 = {  
 ***sortedArray***: ***array100***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 sortArrayStandard(***array200***);  
 endTime = ***performance***.now();  
 info.standardSort200 = {  
 ***sortedArray***: ***array200***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 sortArrayStandard(***array500***);  
 endTime = ***performance***.now();  
 info.standardSort500 = {  
 ***sortedArray***: ***array500***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 sortArrayStandard(***array1000***);  
 endTime = ***performance***.now();  
 info.standardSort1000 = {  
 ***sortedArray***: ***array1000***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 sortArrayStandard(***array5000***);  
 endTime = ***performance***.now();  
 info.standardSort5000 = {  
 ***sortedArray***: ***array5000***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 sortArrayStandard(***array10000***);  
 endTime = ***performance***.now();  
 info.standardSort10000 = {  
 ***sortedArray***: ***array10000***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 sortArrayStandard(***array25000***);  
 endTime = ***performance***.now();  
 info.standardSort25000 = {  
 ***sortedArray***: ***array25000***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 sortArrayStandard(***array50000***);  
 endTime = ***performance***.now();  
 info.standardSort50000 = {  
 ***sortedArray***: ***array50000***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 sortArrayStandard(***array100000***);  
 endTime = ***performance***.now();  
 info.standardSort100000 = {  
 ***sortedArray***: ***array100000***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 addRowToTableStandardSort(info);  
 return info;  
}  
  
function addRowToTableStandardSort(info) {  
 let row = ***document***.createElement('tr');  
 row.style.background = 'gold';  
  
 let tdFunctionName = ***document***.createElement('td');  
 tdFunctionName.innerHTML = "Standard Sort";  
  
 let td100 = ***document***.createElement('td');  
 td100.innerHTML = info.standardSort100.timeTaken + 'ms.';  
  
 let td200 = ***document***.createElement('td');  
 td200.innerHTML = info.standardSort200.timeTaken + 'ms.';  
  
 let td500 = ***document***.createElement('td');  
 td500.innerHTML = info.standardSort500.timeTaken + 'ms.';  
  
 let td1000 = ***document***.createElement('td');  
 td1000.innerHTML = info.standardSort1000.timeTaken + 'ms.';  
  
 let td5000 = ***document***.createElement('td');  
 td5000.innerHTML = info.standardSort5000.timeTaken + 'ms.';  
  
 let td10000 = ***document***.createElement('td');  
 td10000.innerHTML = info.standardSort10000.timeTaken + 'ms.';  
  
 let td25000 = ***document***.createElement('td');  
 td25000.innerHTML = info.standardSort25000.timeTaken + 'ms.';  
  
 let td50000 = ***document***.createElement('td');  
 td50000.innerHTML = info.standardSort50000.timeTaken + 'ms.';  
  
 let td100000 = ***document***.createElement('td');  
 td100000.innerHTML = info.standardSort100000.timeTaken + 'ms.';  
  
 row.appendChild(tdFunctionName);  
 row.appendChild(td100);  
 row.appendChild(td200);  
 row.appendChild(td500);  
 row.appendChild(td1000);  
 row.appendChild(td5000);  
 row.appendChild(td10000);  
 row.appendChild(td25000);  
 row.appendChild(td50000);  
 row.appendChild(td100000);  
  
 ***table***.appendChild(row);  
}  
  
function calculateSumOfElementsStandard(array) {  
 let result = 0;  
 array.forEach((item) => {  
 result += item;  
 });  
 return result;  
}  
  
function calculateSumOfAllArraysStandard() {  
 let startTime;  
 let endTime;  
 let info = {};  
  
 startTime = ***performance***.now();  
 let sum100 = calculateSumOfElementsStandard(***array100***);  
 endTime = ***performance***.now();  
 info.sumStandard100 = {  
 sum: sum100,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 startTime = ***performance***.now();  
 let sum200 = calculateSumOfElementsStandard(***array200***);  
 endTime = ***performance***.now();  
 info.sumStandard200 = {  
 sum: sum200,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 startTime = ***performance***.now();  
 let sum500 = calculateSumOfElementsStandard(***array500***);  
 endTime = ***performance***.now();  
 info.sumStandard500 = {  
 sum: sum500,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 startTime = ***performance***.now();  
 let sum1000 = calculateSumOfElementsStandard(***array1000***);  
 endTime = ***performance***.now();  
 info.sumStandard1000 = {  
 sum: sum1000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 startTime = ***performance***.now();  
 let sum5000 = calculateSumOfElementsStandard(***array5000***);  
 endTime = ***performance***.now();  
 info.sumStandard5000 = {  
 sum: sum5000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 startTime = ***performance***.now();  
 let sum10000 = calculateSumOfElementsStandard(***array10000***);  
 endTime = ***performance***.now();  
 info.sumStandard10000 = {  
 sum: sum10000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 startTime = ***performance***.now();  
 let sum25000 = calculateSumOfElementsStandard(***array25000***);  
 endTime = ***performance***.now();  
 info.sumStandard25000 = {  
 sum: sum25000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 startTime = ***performance***.now();  
 let sum50000 = calculateSumOfElementsStandard(***array50000***);  
 endTime = ***performance***.now();  
 info.sumStandard50000 = {  
 sum: sum50000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 startTime = ***performance***.now();  
 let sum100000 = calculateSumOfElementsStandard(***array100000***);  
 endTime = ***performance***.now();  
 info.sumStandard100000 = {  
 sum: sum100000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 addRowToTableStandardSum(info);  
 return info;  
}  
  
function addRowToTableStandardSum(info) {  
 let row = ***document***.createElement('tr');  
 row.style.background = 'green';  
  
 let tdFunctionName = ***document***.createElement('td');  
 tdFunctionName.innerHTML = "Standard Sum";  
  
 let td100 = ***document***.createElement('td');  
 td100.innerHTML = info.sumStandard100.timeTaken + 'ms.';  
  
 let td200 = ***document***.createElement('td');  
 td200.innerHTML = info.sumStandard200.timeTaken + 'ms.';  
  
 let td500 = ***document***.createElement('td');  
 td500.innerHTML = info.sumStandard500.timeTaken + 'ms.';  
  
 let td1000 = ***document***.createElement('td');  
 td1000.innerHTML = info.sumStandard1000.timeTaken + 'ms.';  
  
 let td5000 = ***document***.createElement('td');  
 td5000.innerHTML = info.sumStandard5000.timeTaken + 'ms.';  
  
 let td10000 = ***document***.createElement('td');  
 td10000.innerHTML = info.sumStandard10000.timeTaken + 'ms.';  
  
 let td25000 = ***document***.createElement('td');  
 td25000.innerHTML = info.sumStandard25000.timeTaken + 'ms.';  
  
 let td50000 = ***document***.createElement('td');  
 td50000.innerHTML = info.sumStandard50000.timeTaken + 'ms.';  
  
 let td100000 = ***document***.createElement('td');  
 td100000.innerHTML = info.sumStandard100000.timeTaken + 'ms.';  
  
 row.appendChild(tdFunctionName);  
 row.appendChild(td100);  
 row.appendChild(td200);  
 row.appendChild(td500);  
 row.appendChild(td1000);  
 row.appendChild(td5000);  
 row.appendChild(td10000);  
 row.appendChild(td25000);  
 row.appendChild(td50000);  
 row.appendChild(td100000);  
  
 ***table***.appendChild(row);  
}  
  
function standardDoublyAllElements(array) {  
 array.forEach((item, index) => array[index] = item\*\*2);  
 return array;  
}  
  
function doublyAllArraysStandard() {  
 let startTime;  
 let endTime;  
 let info = {};  
  
 startTime = ***performance***.now();  
 standardDoublyAllElements(***array100***);  
 endTime = ***performance***.now();  
 info.doubly100 = {  
 ***doubledArray***: ***array100***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 standardDoublyAllElements(***array200***);  
 endTime = ***performance***.now();  
 info.doubly200 = {  
 ***doubledArray***: ***array200***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 standardDoublyAllElements(***array500***);  
 endTime = ***performance***.now();  
 info.doubly500 = {  
 ***doubledArray***: ***array500***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 standardDoublyAllElements(***array1000***);  
 endTime = ***performance***.now();  
 info.doubly1000 = {  
 ***doubledArray***: ***array1000***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 standardDoublyAllElements(***array5000***);  
 endTime = ***performance***.now();  
 info.doubly5000 = {  
 ***doubledArray***: ***array5000***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 standardDoublyAllElements(***array10000***);  
 endTime = ***performance***.now();  
 info.doubly10000 = {  
 ***doubledArray***: ***array10000***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 standardDoublyAllElements(***array25000***);  
 endTime = ***performance***.now();  
 info.doubly25000 = {  
 ***doubledArray***: ***array25000***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 standardDoublyAllElements(***array50000***);  
 endTime = ***performance***.now();  
 info.doubly50000 = {  
 ***doubledArray***: ***array50000***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 standardDoublyAllElements(***array100000***);  
 endTime = ***performance***.now();  
 info.doubly100000 = {  
 ***doubledArray***: ***array100000***,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 addRowToTableStandardDouble(info);  
 return info;  
}  
  
function addRowToTableStandardDouble(info) {  
 let row = ***document***.createElement('tr');  
 row.style.background = 'yellow';  
  
 let tdFunctionName = ***document***.createElement('td');  
 tdFunctionName.innerHTML = "Standard Double";  
  
 let td100 = ***document***.createElement('td');  
 td100.innerHTML = info.doubly100.timeTaken + 'ms.';  
  
 let td200 = ***document***.createElement('td');  
 td200.innerHTML = info.doubly200.timeTaken + 'ms.';  
  
 let td500 = ***document***.createElement('td');  
 td500.innerHTML = info.doubly500.timeTaken + 'ms.';  
  
 let td1000 = ***document***.createElement('td');  
 td1000.innerHTML = info.doubly1000.timeTaken + 'ms.';  
  
 let td5000 = ***document***.createElement('td');  
 td5000.innerHTML = info.doubly5000.timeTaken + 'ms.';  
  
 let td10000 = ***document***.createElement('td');  
 td10000.innerHTML = info.doubly10000.timeTaken + 'ms.';  
  
 let td25000 = ***document***.createElement('td');  
 td25000.innerHTML = info.doubly25000.timeTaken + 'ms.';  
  
 let td50000 = ***document***.createElement('td');  
 td50000.innerHTML = info.doubly50000.timeTaken + 'ms.';  
  
 let td100000 = ***document***.createElement('td');  
 td100000.innerHTML = info.doubly100000.timeTaken + 'ms.';  
  
 row.appendChild(tdFunctionName);  
 row.appendChild(td100);  
 row.appendChild(td200);  
 row.appendChild(td500);  
 row.appendChild(td1000);  
 row.appendChild(td5000);  
 row.appendChild(td10000);  
 row.appendChild(td25000);  
 row.appendChild(td50000);  
 row.appendChild(td100000);  
  
 ***table***.appendChild(row);  
}  
  
function standardPositiveFilter(array) {  
 let result = array.filter(item => item > 0);  
 return result;  
}  
  
function filterAllArraysStandard() {  
 let startTime;  
 let endTime;  
 let info = {};  
  
 startTime = ***performance***.now();  
 let filter100 = standardPositiveFilter(***array100***);  
 endTime = ***performance***.now();  
 info.standardFilter100 = {  
 filteredArray: filter100,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let filter200 = standardPositiveFilter(***array200***);  
 endTime = ***performance***.now();  
 info.standardFilter200 = {  
 filteredArray: filter200,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let filter500 = standardPositiveFilter(***array500***);  
 endTime = ***performance***.now();  
 info.standardFilter500 = {  
 filteredArray: filter500,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let filter1000 = standardPositiveFilter(***array1000***);  
 endTime = ***performance***.now();  
 info.standardFilter1000 = {  
 filteredArray: filter1000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let filter5000 = standardPositiveFilter(***array5000***);  
 endTime = ***performance***.now();  
 info.standardFilter5000 = {  
 filteredArray: filter5000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let filter10000 = standardPositiveFilter(***array10000***);  
 endTime = ***performance***.now();  
 info.standardFilter10000 = {  
 filteredArray: filter10000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let filter25000 = standardPositiveFilter(***array25000***);  
 endTime = ***performance***.now();  
 info.standardFilter25000 = {  
 filteredArray: filter25000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let filter50000 = standardPositiveFilter(***array50000***);  
 endTime = ***performance***.now();  
 info.standardFilter50000 = {  
 filteredArray: filter50000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let filter100000 = standardPositiveFilter(***array100000***);  
 endTime = ***performance***.now();  
 info.standardFilter100000 = {  
 filteredArray: filter100000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 addRowToTableStandardFilter(info);  
 return info;  
}  
  
function addRowToTableStandardFilter(info) {  
 let row = ***document***.createElement('tr');  
 row.style.background = 'lightblue';  
  
 let tdFunctionName = ***document***.createElement('td');  
 tdFunctionName.innerHTML = "Standard Positive Filter";  
  
 let td100 = ***document***.createElement('td');  
 td100.innerHTML = info.standardFilter100.timeTaken + 'ms.';  
  
 let td200 = ***document***.createElement('td');  
 td200.innerHTML = info.standardFilter200.timeTaken + 'ms.';  
  
 let td500 = ***document***.createElement('td');  
 td500.innerHTML = info.standardFilter500.timeTaken + 'ms.';  
  
 let td1000 = ***document***.createElement('td');  
 td1000.innerHTML = info.standardFilter1000.timeTaken + 'ms.';  
  
 let td5000 = ***document***.createElement('td');  
 td5000.innerHTML = info.standardFilter5000.timeTaken + 'ms.';  
  
 let td10000 = ***document***.createElement('td');  
 td10000.innerHTML = info.standardFilter10000.timeTaken + 'ms.';  
  
 let td25000 = ***document***.createElement('td');  
 td25000.innerHTML = info.standardFilter25000.timeTaken + 'ms.';  
  
 let td50000 = ***document***.createElement('td');  
 td50000.innerHTML = info.standardFilter50000.timeTaken + 'ms.';  
  
 let td100000 = ***document***.createElement('td');  
 td100000.innerHTML = info.standardFilter100000.timeTaken + 'ms.';  
  
 row.appendChild(tdFunctionName);  
 row.appendChild(td100);  
 row.appendChild(td200);  
 row.appendChild(td500);  
 row.appendChild(td1000);  
 row.appendChild(td5000);  
 row.appendChild(td10000);  
 row.appendChild(td25000);  
 row.appendChild(td50000);  
 row.appendChild(td100000);  
  
 ***table***.appendChild(row);  
}  
  
function standardCheck(array) {  
 const isNegative = (currentValue) => currentValue < 0;  
 let allNegative = array.every(isNegative);  
 let oneNegative = array.some(item => item < 0) == true;  
  
 return {  
 allNegative: allNegative,  
 oneNegative: oneNegative  
 };  
}  
  
function checkAllArraysStandard() {  
 let startTime;  
 let endTime;  
 let info = {};  
  
 startTime = ***performance***.now();  
 let check100 = standardCheck(***array100***);  
 endTime = ***performance***.now();  
 info.standardCheck100 = {  
 data: check100,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 startTime = ***performance***.now();  
 let check200 = standardCheck(***array200***);  
 endTime = ***performance***.now();  
 info.standardCheck200 = {  
 data: check200,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 startTime = ***performance***.now();  
 let check500 = standardCheck(***array500***);  
 endTime = ***performance***.now();  
 info.standardCheck500 = {  
 data: check500,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 startTime = ***performance***.now();  
 let check1000 = standardCheck(***array1000***);  
 endTime = ***performance***.now();  
 info.standardCheck1000 = {  
 data: check1000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 startTime = ***performance***.now();  
 let check5000 = standardCheck(***array5000***);  
 endTime = ***performance***.now();  
 info.standardCheck5000 = {  
 data: check5000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 startTime = ***performance***.now();  
 let check10000 = standardCheck(***array10000***);  
 endTime = ***performance***.now();  
 info.standardCheck10000 = {  
 data: check10000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 startTime = ***performance***.now();  
 let check25000 = standardCheck(***array25000***);  
 endTime = ***performance***.now();  
 info.standardCheck25000 = {  
 data: check25000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 startTime = ***performance***.now();  
 let check50000 = standardCheck(***array50000***);  
 endTime = ***performance***.now();  
 info.standardCheck50000 = {  
 data: check50000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 startTime = ***performance***.now();  
 let check100000 = standardCheck(***array100000***);  
 endTime = ***performance***.now();  
 info.standardCheck100000 = {  
 data: check100000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 };  
  
 addRowToTableStandardCheck(info);  
 return info;  
}  
  
function addRowToTableStandardCheck(info) {  
 let row = ***document***.createElement('tr');  
 row.style.background = 'orange';  
  
 let tdFunctionName = ***document***.createElement('td');  
 tdFunctionName.innerHTML = "Standard Negative Check";  
  
 let td100 = ***document***.createElement('td');  
 td100.innerHTML = info.standardCheck100.timeTaken + 'ms.';  
  
 let td200 = ***document***.createElement('td');  
 td200.innerHTML = info.standardCheck200.timeTaken + 'ms.';  
  
 let td500 = ***document***.createElement('td');  
 td500.innerHTML = info.standardCheck500.timeTaken + 'ms.';  
  
 let td1000 = ***document***.createElement('td');  
 td1000.innerHTML = info.standardCheck1000.timeTaken + 'ms.';  
  
 let td5000 = ***document***.createElement('td');  
 td5000.innerHTML = info.standardCheck5000.timeTaken + 'ms.';  
  
 let td10000 = ***document***.createElement('td');  
 td10000.innerHTML = info.standardCheck10000.timeTaken + 'ms.';  
  
 let td25000 = ***document***.createElement('td');  
 td25000.innerHTML = info.standardCheck25000.timeTaken + 'ms.';  
  
 let td50000 = ***document***.createElement('td');  
 td50000.innerHTML = info.standardCheck50000.timeTaken + 'ms.';  
  
 let td100000 = ***document***.createElement('td');  
 td100000.innerHTML = info.standardCheck100000.timeTaken + 'ms.';  
  
 row.appendChild(tdFunctionName);  
 row.appendChild(td100);  
 row.appendChild(td200);  
 row.appendChild(td500);  
 row.appendChild(td1000);  
 row.appendChild(td5000);  
 row.appendChild(td10000);  
 row.appendChild(td25000);  
 row.appendChild(td50000);  
 row.appendChild(td100000);  
  
 ***table***.appendChild(row);  
}  
  
function standardFindFirstNegative(array) {  
 return array.find(item => item < 0);  
}  
  
function findAllFirstNegativeElementsStandard() {  
 let startTime;  
 let endTime;  
 let info = {};  
  
 startTime = ***performance***.now();  
 let firstNegative100 = standardFindFirstNegative(***array100***);  
 endTime = ***performance***.now();  
 info.firstNeg100 = {  
 elementValue: firstNegative100,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let firstNegative200 = standardFindFirstNegative(***array200***);  
 endTime = ***performance***.now();  
 info.firstNeg200 = {  
 elementValue: firstNegative200,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let firstNegative500 = standardFindFirstNegative(***array500***);  
 endTime = ***performance***.now();  
 info.firstNeg500 = {  
 elementValue: firstNegative500,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let firstNegative1000 = standardFindFirstNegative(***array1000***);  
 endTime = ***performance***.now();  
 info.firstNeg1000 = {  
 elementValue: firstNegative1000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let firstNegative5000 = standardFindFirstNegative(***array5000***);  
 endTime = ***performance***.now();  
 info.firstNeg5000 = {  
 elementValue: firstNegative5000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let firstNegative10000 = standardFindFirstNegative(***array10000***);  
 endTime = ***performance***.now();  
 info.firstNeg10000 = {  
 elementValue: firstNegative10000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let firstNegative25000 = standardFindFirstNegative(***array25000***);  
 endTime = ***performance***.now();  
 info.firstNeg25000 = {  
 elementValue: firstNegative25000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let firstNegative50000 = standardFindFirstNegative(***array50000***);  
 endTime = ***performance***.now();  
 info.firstNeg50000 = {  
 elementValue: firstNegative50000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 startTime = ***performance***.now();  
 let firstNegative100000 = standardFindFirstNegative(***array100000***);  
 endTime = ***performance***.now();  
 info.firstNeg100000 = {  
 elementValue: firstNegative100000,  
 timeTaken: (endTime - startTime).toFixed(2)  
 }  
  
 addRowToTableStandardNegativeCheck(info);  
 return info;  
}  
  
function addRowToTableStandardNegativeCheck(info) {  
 let row = ***document***.createElement('tr');  
 row.style.background = 'gray';  
  
 let tdFunctionName = ***document***.createElement('td');  
 tdFunctionName.innerHTML = "Standard Negative Check";  
  
 let td100 = ***document***.createElement('td');  
 td100.innerHTML = info.firstNeg100.timeTaken + 'ms.';  
  
 let td200 = ***document***.createElement('td');  
 td200.innerHTML = info.firstNeg200.timeTaken + 'ms.';  
  
 let td500 = ***document***.createElement('td');  
 td500.innerHTML = info.firstNeg500.timeTaken + 'ms.';  
  
 let td1000 = ***document***.createElement('td');  
 td1000.innerHTML = info.firstNeg1000.timeTaken + 'ms.';  
  
 let td5000 = ***document***.createElement('td');  
 td5000.innerHTML = info.firstNeg5000.timeTaken + 'ms.';  
  
 let td10000 = ***document***.createElement('td');  
 td10000.innerHTML = info.firstNeg10000.timeTaken + 'ms.';  
  
 let td25000 = ***document***.createElement('td');  
 td25000.innerHTML = info.firstNeg25000.timeTaken + 'ms.';  
  
 let td50000 = ***document***.createElement('td');  
 td50000.innerHTML = info.firstNeg50000.timeTaken + 'ms.';  
  
 let td100000 = ***document***.createElement('td');  
 td100000.innerHTML = info.firstNeg100000.timeTaken + 'ms.';  
  
 row.appendChild(tdFunctionName);  
 row.appendChild(td100);  
 row.appendChild(td200);  
 row.appendChild(td500);  
 row.appendChild(td1000);  
 row.appendChild(td5000);  
 row.appendChild(td10000);  
 row.appendChild(td25000);  
 row.appendChild(td50000);  
 row.appendChild(td100000);  
  
 ***table***.appendChild(row);  
}  
  
function startStandardTest() {  
 // fillArrays();  
 // let standardSortInfo = sortAllArraysStandard();  
 // console.log("Standard Sort:");  
 // console.log(standardSortInfo);  
 // fillArrays();  
 // let standardSumInfo = calculateSumOfAllArraysStandard();  
 // console.log("Standard Sum:");  
 // console.log(standardSumInfo);  
 // fillArrays();  
 // let standardDoublyInfo = doublyAllArraysStandard();  
 // console.log("Standard Doubly:");  
 // console.log(standardDoublyInfo);  
 // fillArrays();  
 // let standardFilterInfo = filterAllArraysStandard();  
 // console.log("Standard Filter:");  
 // console.log(standardFilterInfo);  
 // fillArrays();  
 // let standardCheckInfo = checkAllArraysStandard();  
 // console.log("Standard Check:");  
 // console.log(standardCheckInfo);  
 // fillArrays();  
 // let standardFirstNegativeInfo = findAllFirstNegativeElementsStandard();  
 // console.log("Standard First Negative:");  
 // console.log(standardFirstNegativeInfo);  
}  
  
// startStandardTest();  
  
function start() {  
 // startCustomTest();  
 // startStandardTest();  
 fillArrays();  
 calculateSumOfEachArray();  
 fillArrays();  
 calculateSumOfAllArraysStandard();  
 fillArrays();  
 doublyEachArray();  
 fillArrays();  
 doublyAllArraysStandard();  
 fillArrays();  
 filterAllArrays();  
 fillArrays();  
 filterAllArraysStandard();  
 fillArrays();  
 checkAllArrays();  
 fillArrays();  
 checkAllArraysStandard();  
 fillArrays();  
 findAllFirstNegativeElements();  
 fillArrays();  
 findAllFirstNegativeElementsStandard();  
 fillArrays();  
 sortEachArray();  
 fillArrays();  
 sortAllArraysStandard();  
}

# РЕЗУЛЬТАТ ВИКОНАННЯ

