Міністерство освіти і науки України

Національний технічний університет

«Харківський політехнічний інститут»

кафедра “Інформатика та інтелектуальна власність”

**Розрахуноково-ГРАФІЧНЕ ЗАВДАННЯ**

з дисципліни “Основи WEB технологій”

**Тема**

“ Порівняльний аналіз методів обробки масивів засобами мови JavaScript(JS)”

Виконавець

студент групи КН-321а Андрій КАРАУЛАНОВ

Керівник розрахунково-графічного завдання

ст. викладач кафедри ІІВ Ірина ШУБА

Харків 2022

# ПОСТАНОВКА ЗАДАЧІ

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



2. 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.



3. 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.



4. 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.



5. 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.



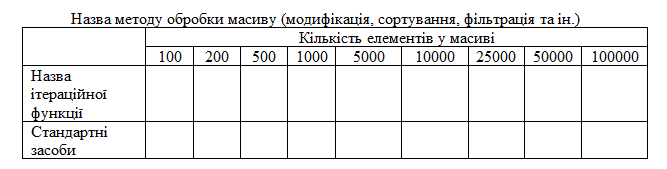
6. Порівняти час пошуку (див. власний варіант) для кожного масиву, створеному у першому завданні, за допомогою ітераційних методів масивів та стандартних засобів (циклів) JS. Якщо перевірка не реалізується для речових значень, скористатися їх цілою частиною.



7. 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.

8. 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.

9. Провести експерименти за представити їх результати в таблицях, форма яких може мати наприклад такий вигляд



10. На основі табличної інформації побудувати порівняльні діаграми, наприклад гістограми, створивши для цього відповідний скрипт, що змінює розміри блоків діаграми (додаткові 10 балів) чи використавши один із стандартних табличних процесорів, наприклад MS Excel. Зробити відповідні висновки.

**ЗМІСТ**

[ПОСТАНОВКА ЗАДАЧІ 2](#_Toc119347166)

[1 ВИКОНАННЯ ЗАВДАННЯ 5](#_Toc119347167)

[1.1 Створення відповідних масивів 5](#_Toc119347168)

[1.2 Час підрахунку пошуку максимального елемента у масиві 6](#_Toc119347169)

[1.3 Час отримання коренів усіх елементів 7](#_Toc119347170)

[1.4 Час пошуку усіх негативних елементів масиву 8](#_Toc119347171)

[1.5 Час пошуку першого позитивного та усіх негативних 10](#_Toc119347172)

[1.6 Час пошуку першого позитивного елементу 11](#_Toc119347173)

[1.7 Час сортування масиву за зменшенням 12](#_Toc119347174)

[Висновок 14](#_Toc119347175)

[Додаток А 15](#_Toc119347176)

# 1 ВИКОНАННЯ ЗАВДАННЯ

## 1.1 Створення відповідних масивів

Згідно завдання було розроблено код, який генерує масив n місткості речових чисел від -200 до 200. Приклад коду який що генерує відповідний масив на 100 елементів наведено нижче.

function gfg\_Run()

{

var l = Math.floor(100);

arr = Array.from({length: l}, ( ) => Math.fround(200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

Повний код наведено у додатку А.

У WEB сторінці э меню, згідно якого можна створити на вивести масив з обраною кількістю елементів. У по дальшому і дії, що описані у завданнях 1 ­– 6. Приклад веб сторінки першого завдання наведено на рисунку 1.1.

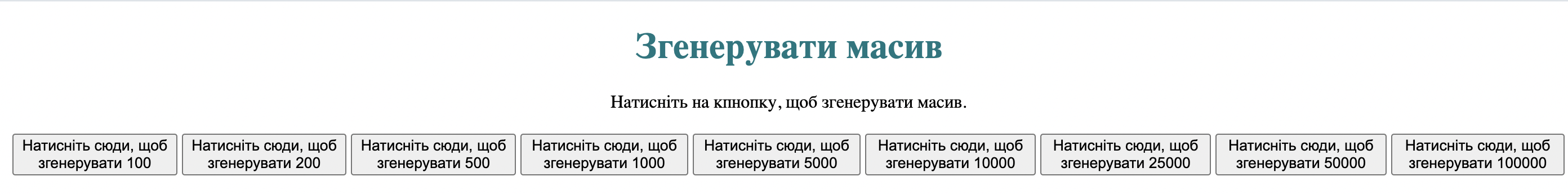


Рисунок 1.1 – WEB сторінка першого завдання

## 1.2 Час підрахунку пошуку максимального елемента у масиві

Приклад функції для підрахунку усіх максимальних елементів:

function find\_max()

{

var max = Math.max.apply(null, arr);

el\_down.innerHTML=max;

}

function test\_find\_max()

{

test(find\_max);

}

function test(function\_to\_test)

{

var start = window.performance.now();

for( let i = 0; i < 100; i++)

{

function\_to\_test();

}

var end = window.performance.now();

var time =(end - start)/10;

alert(`Execution time: ${time} ms`);

}

Таблица 1.1 – Час підрахунку пошуку максимального елемента

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ємкість масиву | 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Час у мс | 0.3 | 0.44 | 0.85 | 1.09 | 5.1 | 8.03 | 14.3 | 24.1 | 44.459 |

## 1.3 Час отримання коренів усіх елементів

Приклад функції отримання квадратних коренів усіх елементів масиву:

function cube()

{

var new\_arr =[];

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

{

new\_arr.push(arr[i]\*arr[i]\*arr[i]);

}

arr=new\_arr;

el\_down.innerHTML=arr;

}

function test\_cube()

{

test(cube);

}

function test(function\_to\_test)

{

var start = window.performance.now();

for( let i = 0; i < 100; i++)

{

function\_to\_test();

}

var end = window.performance.now();

var time =(end - start)/10;

alert(`Execution time: ${time} ms`);

}

Таблица 1.2 – Час отримання квадратних коренів усіх елементів

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ємкість масиву | 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Час у мс | 0.73 | 0.96 | 1.8 | 3.12 | 8.93 | 13.89 | 29.08 | 54.72 | 115.8299 |

## 1.4 Час пошуку усіх негативних елементів масиву

Приклад функції пошуку негативних елементів:

function filter()

{

var new\_arr =[];

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

{

if(arr[i]<0)

{

new\_arr.push(arr[i]);

}

}

arr=new\_arr;

el\_down.innerHTML=arr;

}

function test\_filter()

{

test(filter);

}

function test(function\_to\_test)

{

var start = window.performance.now();

for( let i = 0; i < 100; i++)

{

function\_to\_test();

}

var end = window.performance.now();

var time =(end - start)/10;

alert(`Execution time: ${time} ms`);

}

Таблица 1.3 – Час пошуку усіх негативних елементів

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ємкість масиву | 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Час у мс | 0,35 | 0,439 | 1,1 | 1,51 | 5,40 | 5,55 | 8,489 | 26,03 | 49,01 |

## 1.5 Час пошуку першого позитивного та усіх негативних

Приклад функції пошуку першого позитивного, а інші негативні:

function check()

{

var positiv\_elem = 0;

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

{

if(arr[i]>0)

{

positiv\_elem++;

}

}

if(positiv\_elem==arr.length)

{

alert("All elements are positive");

}

else if (positiv\_elem>0)

{

alert("This array has at least one positive element");

}

}

function test\_check()

{

test(check);

}

function test(function\_to\_test)

{

var start = window.performance.now();

for( let i = 0; i < 100; i++)

{

function\_to\_test();

}

var end = window.performance.now();

var time =(end - start)/10;

alert(`Execution time: ${time} ms`);

}

Таблица 1.4 – Час пошуку позитивного та усіх негативних

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ємкість масиву | 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Час у мс | 0,77 | 1,3 | 3,25 | 6,62 | 24,1 | 49,81 | 128,7 | 261,22 | 522,529 |

## 1.6 Час пошуку першого позитивного елементу

Приклад функції пошуку першого позитивного:

function find()

{

var first\_positiv=0;

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

{

if(arr[i]>0)

{

first\_positiv= arr[i];

break;

}

}

el\_down.innerHTML=first\_positiv;

}

function test\_find()

{

test(find);

}

function test(function\_to\_test)

{

var start = window.performance.now();

for( let i = 0; i < 100; i++)

{

function\_to\_test();

}

var end = window.performance.now();

var time =(end - start)/10;

alert(`Execution time: ${time} ms`);

}

Таблица 1.5 – Час пошуку першого позитивного елементу

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ємкість масиву | 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Час у мс | 0,11 | 0,08 | 0,09 | 0,09 | 0,1 | 0,25 | 0,259 | 0,25 | 0,2 |

## 1.7 Час сортування масиву за зменшенням

Приклад функції сортування:

function sort()

{

arr.sort((a, b) => a-b);

arr.reverse();

el\_down.innerHTML=arr;

}

function test\_sort()

{

test(sort);

}

function test(function\_to\_test)

{

var start = window.performance.now();

for( let i = 0; i < 100; i++)

{

function\_to\_test();

}

var end = window.performance.now();

var time =(end - start)/10;

alert(`Execution time: ${time} ms`);

}

Таблица 1.6 – Час сортування за спаданням

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ємкість масиву | 100 | 200 | 500 | 1000 | 5000 | 10000 | 25000 | 50000 | 100000 |
| Час у мс | 0,69 | 1,26 | 3,32 | 5 | 23,78 | 49,58 | 129,1 | 260,3 | 532,84 |

# Висновок

Для підрахунку максимального елемента збільшується час самого підрахунку, залежності від розміру масиву, це стосується інших даних, отриманих під час дослідження, за виключенням пошуку першого елементу. Час у всіх майже однаковий, але з розміром масиву збільшується можливий пошук першого позитивного елементу, якщо перед ним усі негативні.

Під час розрахунково-графічного завдання було закріплено низку вивченого матеріалу с курсу основ WEB-технологій. У цій робіти було застосовано раніше вивчені алгоритми за допомогою мови JS.

# Додаток А

<!DOCTYPE HTML>

<html>

<head>

<title>exercise 1</title>

</head>

<body style="text-align:center;">

<h1 style="color: rgb(0,255,127)"> Generate array </h1>

<p id="GFG\_UP"> </p>

<table>

<tr>

<td> <button onclick="gfg\_Run()"> Click Here to generate 100 </button> </td>

<td><button onclick="gf\_Run()"> Click Here to generate 200 </button></td>

<td><button onclick="g\_Run()"> Click Here to generate 500 </button></td>

<td><button onclick="Run()"> Click Here to generate 1000 </button></td>

<td><button onclick="un()"> Click Here to generate 5000 </button></td>

<td><button onclick="n()"> Click Here to generate 10000 </button></td>

<td><button onclick="fg\_Run()"> Click Here to generate 25000 </button></td>

<td><button onclick="gfgRun()"> Click Here to generate 50000 </button></td>

<td><button onclick="gfg\_R()"> Click Here to generate 100000 </button></td>

</tr>

</table>

<p id="GFG\_DOWN" style="color:rgb(219,112,147);"></p>

<script>

var el\_up = document.getElementById("GFG\_UP");

var el\_down = document.getElementById("GFG\_DOWN");

el\_up.innerHTML = "Click on the button to generate the array.";

function gfg\_Run()

{

var l = Math.floor(100);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gf\_Run()

{

var l = Math.floor(200);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function g\_Run()

{

var l = Math.floor(500);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function Run()

{

var l = Math.floor(1000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function un()

{

var l = Math.floor(5000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function n()

{

var l = Math.floor(10000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function fg\_Run()

{

var l = Math.floor(25000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gfgRun()

{

var l = Math.floor(50000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gfg\_R()

{

var l = Math.floor(100000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

</script>

</body>

</html>

<!DOCTYPE HTML>

<html>

<head>

<title>exercise 2</title>

</head>

<body style="text-align:center;">

<h1 style="color: rgb(0,255,127)"> Generate array </h1>

<p id="GFG\_UP"> </p>

<table>

<tr>

<td> <button onclick="gfg\_Run()"> Click Here to generate 100 </button>

</td>

<td><button onclick="gf\_Run()"> Click Here to generate 200 </button>

</td>

<td><button onclick="g\_Run()"> Click Here to generate 500 </button>

</td>

<td><button onclick="Run()"> Click Here to generate 1000 </button>

</td>

<td><button onclick="un()"> Click Here to generate 5000 </button>

</td>

<td><button onclick="n()"> Click Here to generate 10000 </button>

</td>

<td><button onclick="fg\_Run()"> Click Here to generate 25000 </button>

</td>

<td><button onclick="gfgRun()"> Click Here to generate 50000 </button>

</td>

<td><button onclick="gfg\_R()"> Click Here to generate 100000 </button>

</td>

<td><button onclick="test\_find\_max()"> Click Here to find max value </button>

</td>

</tr>

</table>

<p id="GFG\_DOWN" style="color:rgb(219,112,147);"></p>

<script>

var el\_up = document.getElementById("GFG\_UP");

var el\_down = document.getElementById("GFG\_DOWN");

el\_up.innerHTML = "Click on the button to generate the array.";

function gfg\_Run()

{

var l = Math.floor(100);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gf\_Run()

{

var l = Math.floor(200);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function g\_Run()

{

var l = Math.floor(500);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function Run()

{ var l = Math.floor(1000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function un()

{

var l = Math.floor(5000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function n()

{ var l = Math.floor(10000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function fg\_Run()

{

var l = Math.floor(25000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gfgRun()

{ var l = Math.floor(50000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gfg\_R()

{

var l = Math.floor(100000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function find\_max()

{

var max = Math.max.apply(null, arr);

el\_down.innerHTML=max;

}

function test\_find\_max()

{

test(find\_max);

}

function test(function\_to\_test)

{

var start = window.performance.now();

for( let i = 0; i < 100; i++)

{

function\_to\_test();

}

var end = window.performance.now();

var time =(end - start)/10;

alert(`Execution time: ${time} ms`);

}

</script>

</body>

</html>

<!DOCTYPE HTML>

<html>

<head>

<title>exercise 3</title>

</head>

<body style="text-align:center;">

<h1 style="color: rgb(0,255,127)"> Generate array </h1>

<p id="GFG\_UP"> </p>

<table>

<tr>

<td> <button onclick="gfg\_Run()"> Click Here to generate 100 </button>

</td>

<td><button onclick="gf\_Run()"> Click Here to generate 200 </button>

</td>

<td><button onclick="g\_Run()"> Click Here to generate 500 </button>

</td>

<td><button onclick="Run()"> Click Here to generate 1000 </button>

</td>

<td><button onclick="un()"> Click Here to generate 5000 </button>

</td>

<td><button onclick="n()"> Click Here to generate 10000 </button>

</td>

<td><button onclick="fg\_Run()"> Click Here to generate 25000 </button>

</td>

<td><button onclick="gfgRun()"> Click Here to generate 50000 </button>

</td>

<td><button onclick="gfg\_R()"> Click Here to generate 100000 </button>

</td>

<td><button onclick="test\_cube()"> Count cube </button>

</td>

</tr>

</table>

<p id="GFG\_DOWN" style="color:rgb(219,112,147);"></p>

<script>

var el\_up = document.getElementById("GFG\_UP");

var el\_down = document.getElementById("GFG\_DOWN");

el\_up.innerHTML = "Click on the button to generate the array.";

function gfg\_Run()

{

var l = Math.floor(100);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gf\_Run()

{

var l = Math.floor(200);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function g\_Run()

{

var l = Math.floor(500);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function Run()

{ var l = Math.floor(1000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function un()

{

var l = Math.floor(5000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function n()

{ var l = Math.floor(10000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function fg\_Run()

{

var l = Math.floor(25000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gfgRun()

{ var l = Math.floor(50000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gfg\_R()

{

var l = Math.floor(100000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function cube()

{

var new\_arr =[];

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

{

new\_arr.push(arr[i]\*arr[i]\*arr[i]);

}

arr=new\_arr;

el\_down.innerHTML=arr;

}

function test\_cube()

{

test(cube);

}

function test(function\_to\_test)

{

var start = window.performance.now();

for( let i = 0; i < 100; i++)

{

function\_to\_test();

}

var end = window.performance.now();

var time =(end - start)/10;

alert(`Execution time: ${time} ms`);

}

</script>

</body>

</html>

<!DOCTYPE HTML>

<html>

<head>

<title>exercise 4</title>

</head>

<body style="text-align:center;">

<h1 style="color: rgb(0,255,127)"> Generate array </h1>

<p id="GFG\_UP"> </p>

<table>

<tr>

<td> <button onclick="gfg\_Run()"> Click Here to generate 100 </button>

</td>

<td><button onclick="gf\_Run()"> Click Here to generate 200 </button>

</td>

<td><button onclick="g\_Run()"> Click Here to generate 500 </button>

</td>

<td><button onclick="Run()"> Click Here to generate 1000 </button>

</td>

<td><button onclick="un()"> Click Here to generate 5000 </button>

</td>

<td><button onclick="n()"> Click Here to generate 10000 </button>

</td>

<td><button onclick="fg\_Run()"> Click Here to generate 25000 </button>

</td>

<td><button onclick="gfgRun()"> Click Here to generate 50000 </button>

</td>

<td><button onclick="gfg\_R()"> Click Here to generate 100000 </button>

</td>

<td><button onclick="test\_filter()"> Left only negative </button>

</td>

</tr>

</table>

<p id="GFG\_DOWN" style="color:rgb(219,112,147);"></p>

<script>

var el\_up = document.getElementById("GFG\_UP");

var el\_down = document.getElementById("GFG\_DOWN");

el\_up.innerHTML = "Click on the button to generate the array.";

function gfg\_Run()

{

var l = Math.floor(100);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gf\_Run()

{

var l = Math.floor(200);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function g\_Run()

{

var l = Math.floor(500);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function Run()

{ var l = Math.floor(1000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function un()

{

var l = Math.floor(5000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function n()

{ var l = Math.floor(10000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function fg\_Run()

{

var l = Math.floor(25000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gfgRun()

{ var l = Math.floor(50000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gfg\_R()

{

var l = Math.floor(100000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function filter()

{

var new\_arr =[];

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

{

if(arr[i]<0)

{

new\_arr.push(arr[i]);

}

}

arr=new\_arr;

el\_down.innerHTML=arr;

}

function test\_filter()

{

test(filter);

}

function test(function\_to\_test)

{

var start = window.performance.now();

for( let i = 0; i < 100; i++)

{

function\_to\_test();

}

var end = window.performance.now();

var time =(end - start)/10;

alert(`Execution time: ${time} ms`);

}

</script>

</body>

</html>

!DOCTYPE HTML>

<html>

<head>

<title>exercise 5</title>

</head>

<body style="text-align:center;">

<h1 style="color: rgb(0,255,127)"> Generate array </h1>

<p id="GFG\_UP"> </p>

<table>

<tr>

<td> <button onclick="gfg\_Run()"> Click Here to generate 100 </button> </td>

<td><button onclick="gf\_Run()"> Click Here to generate 200 </button></td>

<td><button onclick="g\_Run()"> Click Here to generate 500 </button></td>

<td><button onclick="Run()"> Click Here to generate 1000 </button></td>

<td><button onclick="un()"> Click Here to generate 5000 </button></td>

<td><button onclick="n()"> Click Here to generate 10000 </button></td>

<td><button onclick="fg\_Run()"> Click Here to generate 25000 </button></td>

<td><button onclick="gfgRun()"> Click Here to generate 50000 </button></td>

<td><button onclick="gfg\_R()"> Click Here to generate 100000 </button></td>

<td><button onclick="test\_check()"> Check </button></td>

</tr>

</table>

<p id="GFG\_DOWN" style="color:rgb(219,112,147);"></p>

<script>

var el\_up = document.getElementById("GFG\_UP");

var el\_down = document.getElementById("GFG\_DOWN");

el\_up.innerHTML = "Click on the button to generate the array.";

function gfg\_Run()

{

var l = Math.floor(100);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gf\_Run()

{

var l = Math.floor(200);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function g\_Run()

{

var l = Math.floor(500);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function Run()

{ var l = Math.floor(1000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function un()

{

var l = Math.floor(5000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function n()

{ var l = Math.floor(10000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function fg\_Run()

{

var l = Math.floor(25000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gfgRun()

{ var l = Math.floor(50000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gfg\_R()

{

var l = Math.floor(100000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function check()

{

var positiv\_elem = 0;

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

{

if(arr[i]>0)

{

positiv\_elem++;

}

}

if(positiv\_elem==arr.length)

{

alert("All elements are positive");

}

else if (positiv\_elem>0)

{

alert("This array has at least one positive element");

}

}

function test\_check()

{

test(check);

}

function test(function\_to\_test)

{

var start = window.performance.now();

for( let i = 0; i < 100; i++)

{

function\_to\_test();

}

var end = window.performance.now();

var time =(end - start)/10;

alert(`Execution time: ${time} ms`);

}

</script>

</body>

</html>

<!DOCTYPE HTML>

<html>

<head>

<title>exercise 6</title>

</head>

<body style="text-align:center;">

<h1 style="color: rgb(0,255,127)"> Generate array </h1>

<p id="GFG\_UP"> </p>

<table>

<tr>

<td> <button onclick="gfg\_Run()"> Click Here to generate 100 </button> </td>

<td><button onclick="gf\_Run()"> Click Here to generate 200 </button></td>

<td><button onclick="g\_Run()"> Click Here to generate 500 </button></td>

<td><button onclick="Run()"> Click Here to generate 1000 </button></td>

<td><button onclick="un()"> Click Here to generate 5000 </button></td>

<td><button onclick="n()"> Click Here to generate 10000 </button></td>

<td><button onclick="fg\_Run()"> Click Here to generate 25000 </button></td>

<td><button onclick="gfgRun()"> Click Here to generate 50000 </button></td>

<td><button onclick="gfg\_R()"> Click Here to generate 100000 </button></td>

<td><button onclick="test\_find()">Find first positive </button></td>

</tr>

</table>

<p id="GFG\_DOWN" style="color:rgb(219,112,147);"></p>

<script>

var el\_up = document.getElementById("GFG\_UP");

var el\_down = document.getElementById("GFG\_DOWN");

el\_up.innerHTML = "Click on the button to generate the array.";

function gfg\_Run()

{

var l = Math.floor(100);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gf\_Run()

{

var l = Math.floor(200);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function g\_Run()

{

var l = Math.floor(500);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function Run()

{ var l = Math.floor(1000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function un()

{

var l = Math.floor(5000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function n()

{ var l = Math.floor(10000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function fg\_Run()

{

var l = Math.floor(25000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gfgRun()

{ var l = Math.floor(50000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gfg\_R()

{

var l = Math.floor(100000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function find()

{

var first\_positiv=0;

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

{

if(arr[i]>0)

{

first\_positiv= arr[i];

break;

}

}

el\_down.innerHTML=first\_positiv;

}

function test\_find()

{

test(find);

}

function test(function\_to\_test)

{

var start = window.performance.now();

for( let i = 0; i < 100; i++)

{

function\_to\_test();

}

var end = window.performance.now();

var time =(end - start)/10;

alert(`Execution time: ${time} ms`);

}

</script>

</body>

</html>

<!DOCTYPE HTML>

<html>

<head>

<title>exercise 7</title>

</head>

<body style="text-align:center;">

<h1 style="color: rgb(0,255,127)"> Generate array </h1>

<p id="GFG\_UP"> </p>

<table>

<tr>

<td> <button onclick="gfg\_Run()"> Click Here to generate 100 </button> </td>

<td><button onclick="gf\_Run()"> Click Here to generate 200 </button></td>

<td><button onclick="g\_Run()"> Click Here to generate 500 </button></td>

<td><button onclick="Run()"> Click Here to generate 1000 </button></td>

<td><button onclick="un()"> Click Here to generate 5000 </button></td>

<td><button onclick="n()"> Click Here to generate 10000 </button></td>

<td><button onclick="fg\_Run()"> Click Here to generate 25000 </button></td>

<td><button onclick="gfgRun()"> Click Here to generate 50000 </button></td>

<td><button onclick="gfg\_R()"> Click Here to generate 100000 </button></td>

<td><button onclick="test\_sort()">Sort array by descending </button></td>

</tr>

</table>

<p id="GFG\_DOWN" style="color:rgb(219,112,147);"></p>

<script>

var el\_up = document.getElementById("GFG\_UP");

var el\_down = document.getElementById("GFG\_DOWN");

el\_up.innerHTML = "Click on the button to generate the array.";

function gfg\_Run()

{

var l = Math.floor(100);

arr = Array.from({length: l}, ( ) =>Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gf\_Run()

{

var l = Math.floor(200);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function g\_Run()

{

var l = Math.floor(500);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function Run()

{ var l = Math.floor(1000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function un()

{

var l = Math.floor(5000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function n()

{ var l = Math.floor(10000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function fg\_Run()

{

var l = Math.floor(25000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gfgRun()

{ var l = Math.floor(50000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function gfg\_R()

{

var l = Math.floor(100000);

arr = Array.from({length: l}, ( ) => Math.fround(-200+Math.random()\* (200 + 200)).toFixed(2));

el\_down.innerHTML=arr;

}

function sort()

{

arr.sort((a, b) => a-b);

arr.reverse();

el\_down.innerHTML=arr;

}

function test\_sort()

{

test(sort);

}

function test(function\_to\_test)

{

var start = window.performance.now();

for( let i = 0; i < 100; i++)

{

function\_to\_test();

}

var end = window.performance.now();

var time =(end - start)/10;

alert(`Execution time: ${time} ms`);

}

</script>

</body>

</html>