Содержание

[1 Текст задания 3](#_Toc501904511)

[2 Исходные тексты программы 4](#_Toc501904512)

[3 Скриншоты выполнения программы 12](#_Toc501904513)

[4 Заключение 13](#_Toc501904514)

[5 Используемая литература 14](#_Toc501904515)

1. Текст задания

Задача 10

Есть два отсортированных в порядке неубывания массива A[1,N] и B[1,M]. Получить отсортированный по неубыванию массив C[1,N+M], состоящий из элементов массивов A и B («слить» вместе массивы A и B).

Ввод входных параметров осуществляется с помощью клавиатуры и мыши.

1. Исходные тексты программы

**index.html**

<!DOCTYPE html>

<html lang="ru">

<head>

<title>Лабораторная работа №4 - Слияние упорядоченныз массивов</title>

<meta charset="UTF-8">

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

<link rel="stylesheet" type="text/css" href="css/style.css">

<script type="text/javascript" src="lib/jquery-2.2.4.min.js"></script>

<script type="text/javascript" src="js/merge.js"></script>

</head>

<body>

<div class="wrapper">

<div id="condition">

<h2>Условие задачи</h2>

<p>Есть два отсортированных в порядке неубывания массива A[1,N] и B[1,M].</p>

<p>Получить отсортированный по неубыванию массив C[1,N+M], состоящий из элементов массивов A и B ("слить" вместе массивы A и B).</p>

</div>

<div id="input">

<h2>Входные параметры</h2>

<p><label>Массив A: <input id="arrayA" type="text"></label></p>

<p><label>Массив B: <input id="arrayB" type="text"></label></p>

<p>Введите элементы массивов из целых или вещественных чисел (разделитель точка "."), через пробельный символ. Например ввод массива может быть: <b>1.5 2.4 3.3 4.2 5.1</b>.</p>

</div>

<div id="output">

<h2>Выходные параметры</h2>

<p id="answer">Входные параметры не введены...</p>

</div>

</div>

</body>

</html>

**style.css**

@font-face

{

font-family:Roboto;

src:url(fonts/roboto/Roboto.woff2) format(woff2), url(fonts/roboto/Roboto.woff) format(woff), url(fonts/roboto/Roboto.ttf) format(truetype);

font-weight:400;

font-style:normal;

}

\*

{

margin:0;

padding:0;

}

html

{

font-size:62.5%;

}

body

{

font-family:Roboto;

}

.wrapper

{

max-width:800px;

width:calc(100%-20px);

border-left:1px solid #39c;

border-right:1px solid #39c;

margin:0 auto;

}

h2

{

font-size:2em;

color:#fff;

text-align:center;

padding:5px 8px;

background:#39c;

}

p

{

font-size:1.6em;

margin:5px;

}

input

{

font-size:1em;

}

label

{

display:block;

width:100%;

text-align:center;

}

label > input

{

width:100%;

box-sizing:border-box;

text-align:center;

outline:0;

padding:4px 2px;

}

#condition p

{

text-align:justify;

}

#output

{

border-bottom:1px solid #39c;

}

**merge.js**

/\*\*

\* Получить элементы массива A в виде строки

\*

\* @returns {string}

\* Элементы массива

\*/

function getA()

{

return jQuery('#arrayA').val();

}

/\*\*

\* Получить элементы массива B в виде строки

\*

\* @returns {string}

\* Элементы массива

\*/

function getB()

{

return jQuery('#arrayB').val();

}

/\*\*

\* Конвертирует строковые значения массива в вещественные

\*

\* @param {array} arr

\* Входной массив

\* @returns {array}

\* Вещественный массив

\*/

function converArrayToFloat(arr)

{

if (Array.isArray(arr))

{

var output = arr.map(function(item)

{

return parseFloat(item);

});

}

else

{

output = arr;

}

return output;

}

/\*\*

\* Проверяет упорядоченность массива по-возрастанию

\*

\* @param {double} arr

\* Передаваемый массив массив

\* @return {boolean}

\* Результат проверки

\*/

function checkSortUp(arr)

{

var result = true;

if (Array.isArray(arr) && arr.length > 1)

{

jQuery.each(arr, function(index)

{

if (arr[index] > arr[index + 1])

{

result = false;

}

});

}

return result;

}

/\*\*

\* Слияние упорядоченных массивов

\*

\* @param {double} a

\* Cортированный по-возрастанию массив A

\* @param {double} b

\* Cортированный по-возрастанию массив B

\* @return {array}

\* Объединенный массив

\*/

function merge(a, b)

{

var output = [];

if (Array.isArray(a) && Array.isArray(b))

{

// Выполняем опрерации до тех пор пока есть что сравнивать

while (a.length > 0 && b.length > 0)

{

// Метод shift() удаляет первый элемент из массива

// и возвращает его значение.

// Метод push() добавляет элемент в массив

output.push(a[0] < b[0] ? a.shift() : b.shift());

}

// Метод concat() возвращает новый массив, состоящий из массива,

// на котором он был вызван, соединённого с другими массивами и/или

// значениями, переданными в качестве аргументов.

return output.concat(a.length ? a : b);

}

return output;

}

/\*\*

\* Выполняем действия когда DOM полностью загружен

\*/

jQuery(document).ready(function()

{

// Отслеживаем изменение числа элементов любого из массивов

jQuery('#arrayA, #arrayB').on('change keyup', function()

{

var inputA = getA(); // Ввведеные данные для массива A

var inputB = getB(); // Ввведеные данные для массива B

var regsym = /(^[-0-9 .]+)$/g; // Проверка на допустимые символы

var regdot = /[.]{2,}/; // Проверка на множество точек

var message = ''; // Сообщение пользователю

// Обработка входных данных

if (inputA === '')

{ // Текстовое поле массива A не заполнено

message += 'Введите элементы массива A. ';

}

else if (inputB === '')

{ // Текстовое поле массива B не заполнено

message = 'Введите элементы массива B. ';

}

else if (inputA.match(regsym) === null || inputA.match(regdot) !== null)

{

message = 'В массиве А содержатся недопустимые символы. ';

}

else if (inputB.match(regsym) === null || inputB.match(regdot) !== null)

{

message = 'В массиве B содержатся недопустимые символы. ';

}

if (message === '')

{

var arrayA = []; // Массив A

var arrayB = []; // Массив B

var arrayC = []; // Массив C

var regval = /(-?[0-9]+(\.[0-9]+)?)/g;

arrayA = converArrayToFloat(inputA.match(regval));

arrayB = converArrayToFloat(inputB.match(regval));

console.log(arrayA);

if (checkSortUp(arrayA) !== true)

{

message = 'Массив A не упорядочен по-возрастанию';

}

else if (checkSortUp(arrayB) !== true)

{

message = 'Массив B не упорядочен по-возрастанию';

}

else

{

arrayC = merge(arrayA, arrayB);

message = 'Результат слияния упрорядоченных массивов:';

message = 'Массив С: <b>' + arrayC.join(' ') + '</b>';

}

}

jQuery('#answer').html(message);

});

});

1. Скриншоты выполнения программы

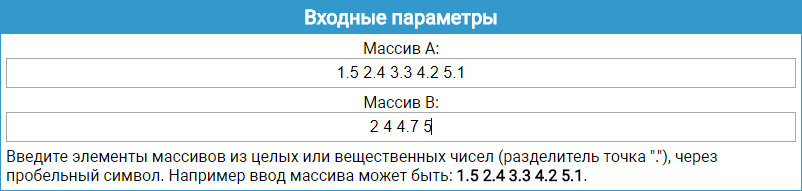


Рисунок 1 – Ввод входных параметров с клавиатуры

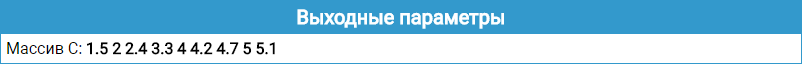


Рисунок 2 – Результат выполнения программы

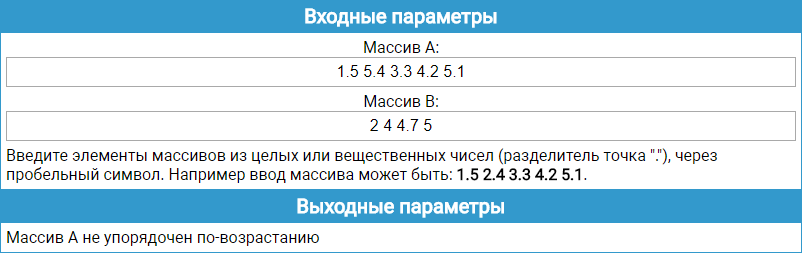


Рисунок 3 – Обработка ошибок в случае неверного ввода входных параметров. Введен неупорядоченный массив

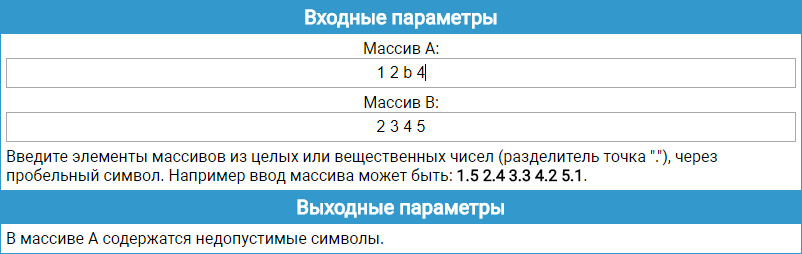


Рисунок 4 – Обработка ошибок в случае неверного ввода входных параметров. Введены недопустимые символы

1. Заключение

В ходе выполнения лабораторной работы изучены методы сортировки, в том числе сортировка упорядоченных массивов методом «слияния».

.

1. Используемая литература

1 Методические указания по выполнению лабораторных работ по дисциплине «Технологии программирования» – Нижний Новгород.: НГТУ, 2015. – 25 с.

2 Стандарт предприятия СТП 1-У-НГТУ-2004. Общие требования к оформлению пояснительных записок дипломных и курсовых проектов. – Нижний Новгород.: НГТУ, 2004. – 22 с.