|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **ПРОГРАМНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.03 ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

**Отчет**

|  |  |
| --- | --- |
| **По лабораторной работе №** | 2 |

**Дисциплина:** Архитектура ЭВМ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-52Б |  |  | Н.А. Гарасев |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | А.Ю. Попов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2020

Оглавление

[**Цель работы:** 4](#_Toc53618555)

[Task 3 5](#_Toc53618556)

[Задание 1 5](#_Toc53618557)

[Задание 2 6](#_Toc53618559)

[Задание 3 7](#_Toc53618562)

[Задание 4 8](#_Toc53618565)

[Задание 5 9](#_Toc53618567)

[Задание 6 10](#_Toc53618569)

[Задание 7 11](#_Toc53618571)

[Task 4 12](#_Toc53618573)

[Задание 1 12](#_Toc53618574)

[Задание 2 13](#_Toc53618575)

[Задание 3 15](#_Toc53618576)

[Задание 4 17](#_Toc53618577)

[Вывод: 18](#_Toc53618578)

# **Цель работы:**

Приобретение и закрепление на практике следующих навыков:

1. создание файла с зависимостями;

2. опыт работы с форматом JSON;

3. работа с файлами в JavaScript;

4. считывание с клавиатуры в JavaScript;

5. запуск сервера в node.js;

6. обработка get запросов.

## Task 3

### Задание 1

**Условие задачи:**

С клавиатуры считывается число N. Далее считывается N строк. Необходимо создать массив и сохранять в него строки только с четной длинной. Получившийся массив необходимо преобразовать в строку JSON и сохранить в файл.

### Листинг кода

"use strict";

let tmp = 0;

let arr = [];

let str;

const readlineSync = require('readline-sync');

const n = parseInt(readlineSync.question("Input count of string: "));

while(tmp !== n) {

    str = readlineSync.question("Input string: ");

    if (str.length % 2 ==  0) {

         arr.push(str);

    }

    tmp++;

}

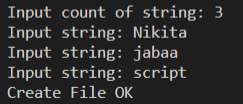
const fs = require("fs");

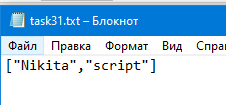
const nameString = "task31.txt";

const contentString = JSON.stringify(arr);

fs.writeFileSync(nameString, contentString);

console.log("Create File OK");





### Задание 2

**Условие задачи:**

### Необходимо считать содержимое файла, в котором хранится массив строк в формате JSON. Нужно вывести только те строки на экран, в которых содержатся только гласные буквы.

### Листинг программы:

"use strict";

function checkOnlyVowels(str) {

    let vowel = ['A', 'E', 'U', 'Y', 'I', 'O'];

    let tmp\_str = str.toUpperCase();

    for (let i = 0; i < tmp\_str.length; i++) {

        if (vowel.indexOf(tmp\_str[i]) == -1) {

            return false;

        }

    }

    return true;

}

const fs = require("fs");

const nameString = "task32.txt";

if (fs.existsSync(nameString)) {

    const contentString = fs.readFileSync(nameString, "utf8");

    const obj = JSON.parse(contentString);

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

        if (checkOnlyVowels(obj[i])) {

            console.log(obj[i]);

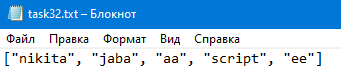
        }

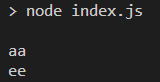
    }

} else {

    console.log("File was not found");

}





### Задание 3

**Условие задачи:**

### С клавиатуры считывается строка - название расширения файлов. Далее считывается строка - адрес папки. Необходимо перебрать все файлы в папке и вывести содержимое файлов, у которых расширение совпадает с введенным расширением.

### Листинг кода:

"use strict";

function checkExtension(extension, needCheckExtension) {

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

        if (extension[extension.length - 1 - i] != needCheckExtension[needCheckExtension.length - 1 - i]) {

            return false;

        }

    }

    return true;

}

const fs = require("fs");

const readlineSync = require('readline-sync');

const extension = readlineSync.question("Input file extension with . : ")

const folder = readlineSync.question("Input adress folder: ")

const arr = fs.readdirSync(folder);

let readData = '';

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

    const fileName = folder + arr[i];

    if (checkExtension(extension, arr[i])) {

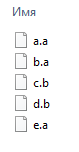
        const contentString = fs.readFileSync(fileName, "utf8");

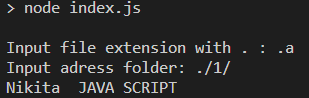
        readData += contentString;

    }

}

console.log(readData);





### Задание 4

**Условие задачи:**

Дана вложенная структура файлов и папок. Все файлы имеют раcширение "txt". Необходимо рекурсивно перебрать вложенную структуру и вывести имена файлов, у которых содержимое не превышает по длине 10 символов.

### Листинг кода:

"use strict";

function recursionReadFile(dir) {

    const arr = fs.readdirSync(dir);

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

        try {

            const fileText = fs.readFileSync(dir + arr[i], "utf8");

            if (fileText.length <= 10) {

                console.log(fileText);

            }

        } catch(err) {

            recursionReadFile(dir + arr[i] + '/');

        }

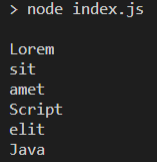
    }

}

const fs = require("fs");

const dir = "./1/"

recursionReadFile(dir);



### Задание 5

**Условие задачи:**

С клавиатуры считывается число N. Далее считывается N строк - имена текстовых файлов. Необходимо склеить всё содержимое введенных файлов в одну большую строку и сохранить в новый файл.

### Листинг кода:

"use strict";

const readlineSync = require('readline-sync');

const fs = require("fs");

const n = parseInt(readlineSync.question("Input count of file: "));

console.log(n);

let tmp = 0;

let str = '';

while(tmp !== n) {

    const file\_name = readlineSync.question("Input file: ");

    if (fs.existsSync(file\_name)) {

        str += fs.readFileSync(file\_name, "utf8");

        tmp++;

    } else {

        console.log("File was not found");

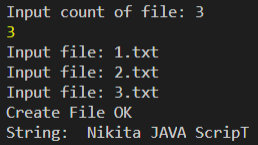
    }

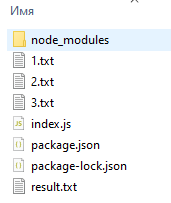
}

fs.writeFileSync("result.txt", str);

console.log("Create File OK");

console.log("String: ", str);





### Задание 6

**Условие задачи:**

Написать код, который позволяет определить максимальный возможный уровень вложенности друг в друга полей в объекте, чтобы данный объект можно было преобразовать в строку формата JSON. Ответом является целое число.

### Листинг кода:

"use strict";

let exObj = {name: "obj0", child:null};

for (let i = 1;;i++) {

    try {

        const obj = {name: "obj" + i, child: exObj};

        JSON.stringify(obj);

        exObj = obj;

    } catch (err) {

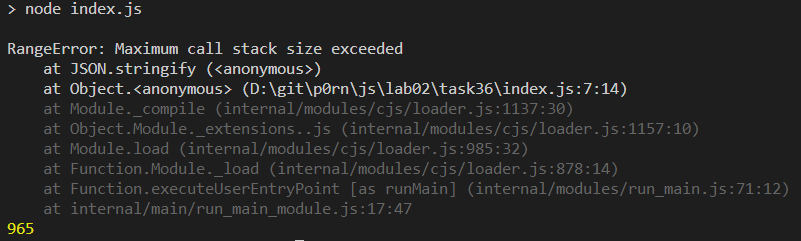
        console.log(err);

        console.log(i);

        break;

    }

}



### Задание 7

**Условие задачи:**

Из файла считывается строка в формате JSON. В этой строке информация об объекте, в котором находится большое количество вложенных друг в друга полей. Объект представляет из себя дерево. Необходимо рекурсивно обработать дерево и найти максимальную вложенность в дереве. Необходимо вывести на экран ветку с максимальной вложенностью.

### Листинг кода:

"use strict";

const fs = require("fs");

const node9 = {name: "9", child1: null, child2: null};

const node8 = {name: "8", child1: node9, child2: null};

const node7 = {name: "7", child1: null, child2: null};

const node6 = {name: "6", child1: null, child2: null};

const node5 = {name: "5", child1: null, child2: null};

const node4 = {name: "4", child1: node7, child2: node8};

const node3 = {name: "3", child1: node5, child2: node6};

const node2 = {name: "2", child1: node3, child2: node4};

const node1 = {name: "1", child1: node2, child2: null};

console.log(JSON.stringify(node1));

fs.writeFileSync("tree.txt", JSON.stringify(node1));

let longestbranch = "";

function search(node, branch) {

    branch += node.name;

    if (branch.length > longestbranch.length)

        longestbranch = branch;

    if (node.child1 !== null)

        search(node.child1, branch)

    if (node.child2 !== null)

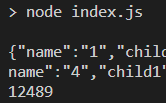
        search(node.child2, branch)

}

const obj = JSON.parse(fs.readFileSync("tree.txt", "utf8"));

search(obj, "");

console.log(longestbranch);



## Task 4

### Задание 1

**Условие задания:**

Запустить сервер. Реализовать на сервере функцию для сравнения трёх чисел и выдачи наибольшего из них. Реализовать страницу с формой ввода для отправки запроса на сервер.

**Листинг кода:**

"use strict";

const fs = require("fs");

const express = require("express");

const app = express();

const port = 5015;

app.listen(port);

console.log("My server on port " + port);

app.get("/me/page", function(request, response) {

    const nameString = request.query.p;

    if (fs.existsSync(nameString)) {

        const contentString = fs.readFileSync(nameString, "utf8");

        response.end(contentString);

    } else {

        const contentString = fs.readFileSync("bad.html", "utf8");

        response.end(contentString);

    }

});

app.get("/find/max", function(request, response) {

    const a = request.query.a;

    const b = request.query.b;

    const c = request.query.c;

    const aInt = parseInt(a);

    const bInt = parseInt(b);

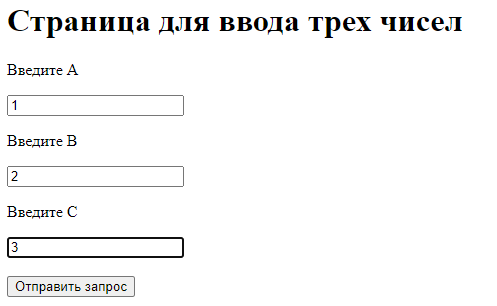
    const cInt = parseInt(c);

    const maxInt = Math.max(Math.max(aInt, bInt), cInt);

    const answerJSON = JSON.stringify({result: maxInt});

    response.end(answerJSON);

});





### Задание 2

**Условие задачи:**

Запустить сервер. На стороне сервера должен храниться файл, внутри которого находится JSON строка. В этой JSON строке хранится информация о массиве объектов. Реализовать на сервере функцию, которая принимает индекс и выдает содержимое ячейки массива по данному индексу. Реализовать страницу с формой ввода для отправки запроса на сервер.

**Листинг кода:**

"use strict";

const fs = require("fs");

const express = require("express");

const app = express();

const port = 5015;

app.listen(port);

console.log("My server on port " + port);

app.get("/me/page", function(request, response) {

    const nameString = request.query.p;

    if (fs.existsSync(nameString)) {

        const contentString = fs.readFileSync(nameString, "utf8");

        response.end(contentString);

    } else {

        const contentString = fs.readFileSync("bad.html", "utf8");

        response.end(contentString);

    }

});

app.get("/getelem", function(request, response) {

    const index = parseInt(request.query.index);

    let res = JSON.parse(fs.readFileSync("arr.txt", "utf-8"));

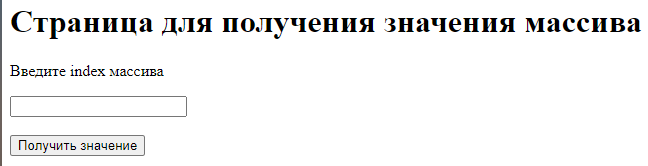
    if (index < 0 || index >= res.length)

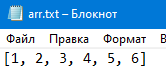
        response.end('Wrong index');

    const result = JSON.stringify(res[index]);

    response.end(result);

});

****





### Задание 3

**Условие задачи:**

Написать программу, которая на вход получает массив названий полей и адрес запроса (куда отправлять). Программа должна генерировать HTML разметку страницы, в которую встроена форма для отправки запроса.

**Листинг кода:**

"use strict";

const fs = require("fs");

const express = require("express");

const app = express();

const port = 5015;

app.listen(port);

console.log("My server on port " + port);

app.get("/me/page", function(request, response) {

    const nameString = "a.html";

    console.log(nameString);

    if (fs.existsSync(nameString)) {

        const contentString = fs.readFileSync(nameString, "utf8");

        response.end(contentString);

    } else {

        const contentString = fs.readFileSync("bad.html", "utf8");

        response.end(contentString);

    }

});

app.get("/input\_fields", function(request, response) {

    let fields = request.query.fields;

    fields = fields.split(", ");

    const adress = request.query.adress;

    const nameString = "result.html";

    let html\_page = '<!DOCTYPE html>\n\

<html>\n\

<head>\n\

    <meta charset="UTF-8">\n\

    <title>Запросы</title>\n\

</head>\n\

<body>\n\

    <h1>Новая страница запроса</h1>\n\

    <form method="GET" action="/' + adress + '">\n'

    for (let field of fields){

        html\_page += '        <p>' + field + '</p>\n';

        html\_page += '        <input name="' + field + '" spellcheck="false" autocomplete="off">\n'

    }

    html\_page += '        <br>\n\

        <br>\n\

        <input type="submit" value="Отправить запрос">\n\

    </form>\n\

</body>\n\

</html>'

    app.get("/" + adress, function(request, response) {

        let arr = request.query;

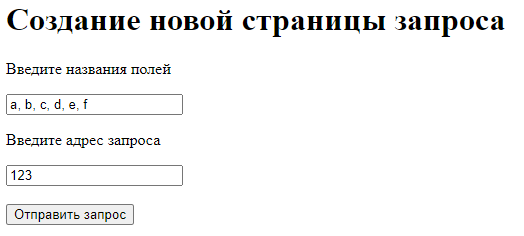
        let contentString = JSON.stringify(arr);

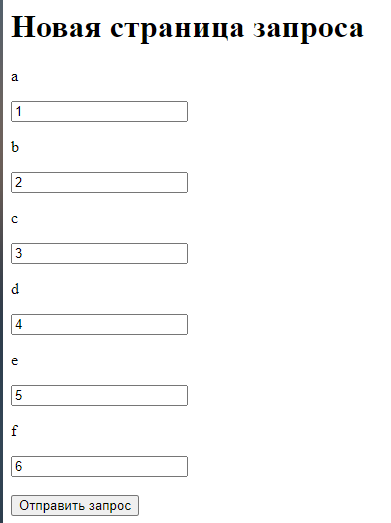
        response.end(contentString);

    });

    response.end(html\_page);

});

****

****

****

### Задание 4

**Условие задачи:**

Запустить сервер. Реализовать на сервере функцию, которая принимает на вход числа A, B и C. Функция должна выдавать массив целых чисел на отрезке от A до B, которые делятся на C нацело.

**Листинг кода:**

"use strict";

const fs = require("fs");

const express = require("express");

const app = express();

const port = 5015;

app.listen(port);

console.log("My server on port " + port);

app.get("/me/page", function(request, response) {

    const nameString = request.query.p;

    if (fs.existsSync(nameString)) {

        const contentString = fs.readFileSync(nameString, "utf8");

        response.end(contentString);

    } else {

        const contentString = fs.readFileSync("bad.html", "utf8");

        response.end(contentString);

    }

});

app.get("/array", function(request, response) {

    const a = request.query.a;

    const b = request.query.b;

    const c = request.query.c;

    const aInt = parseInt(a);

    const bInt = parseInt(b);

    const cInt = parseInt(c);

    let arr = [];

    let n = bInt - aInt;

    for (let i = 0; i <= n; i++) {

        let tmp = aInt + i;

        if ( tmp % cInt == 0) {

            arr.push(tmp);

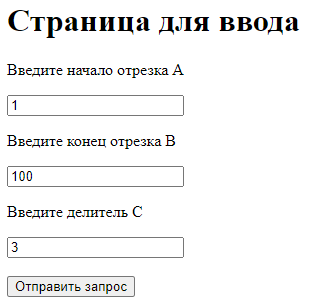
        }

    }

    const answerJSON = JSON.stringify({result: arr});

    response.end(answerJSON);

});

****

****

## Вывод:

Все поставленные задачи были выполнены. Цель лабораторной работы достигнута.