|  |  |
| --- | --- |
| **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 **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 Программная инженерия**

**ОТЧЕТ**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 2 |

**Название:**

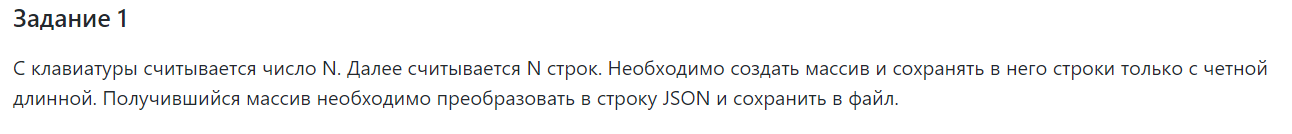
JSON файлы, сервер

**Дисциплина:** Архитектура ЭВМ

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

Москва, 2020

TASK 3-4.



"use strict"

let table = [];

let fs = require("fs");

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

let count = readlineSync.question("Input N: ");

for (let i = 0; i < count; i++) {

let reqString = readlineSync.question("Input string: ");

if ((reqString.length) % 2 === 0) {

table.push(reqString);

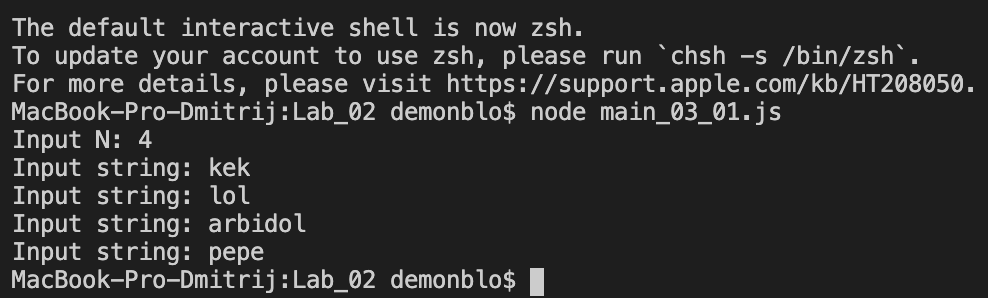
}

}

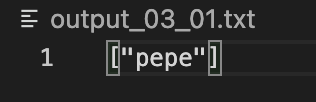
let outputString = JSON.stringify(table);

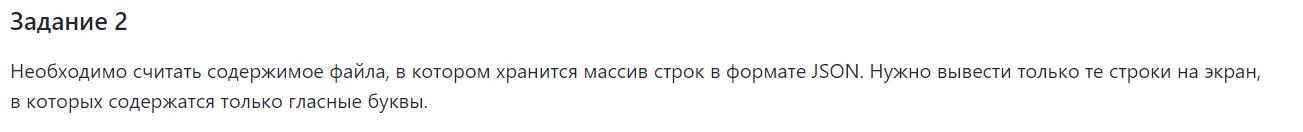
let fileName = "output\_03\_01.txt";

fs.writeFileSync(fileName, outputString);

ТЕСТЫ

Файл:





"use strict"

let fs = require("fs");

let fileName = "input\_03\_02.txt";

let inputString = fs.readFileSync(fileName, "utf8");

let jsonString = JSON.parse(inputString);

function CompareToVowels(letter) {

let vowels = ["a", "u", "e", "o", "y", "i", "A", "U", "E", "O", "Y", "I"];

for (let key in vowels) {

if (letter === vowels[key]) {

return true;

}

}

return false;

}

for (let i in jsonString.mass) {

let flag = true;

for (let j in jsonString.mass[i]) {

if (CompareToVowels(jsonString.mass[i].charAt(j)) === false) {

flag = false;

}

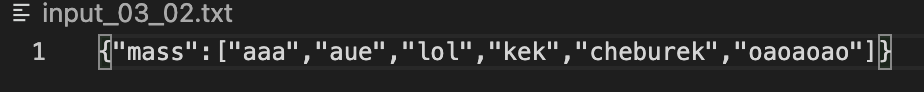
}

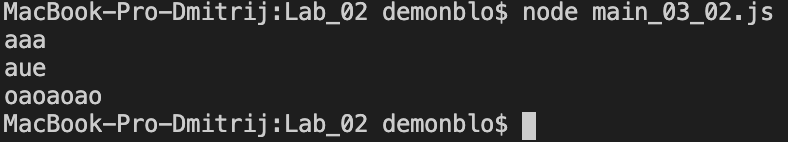
if (flag === true) {

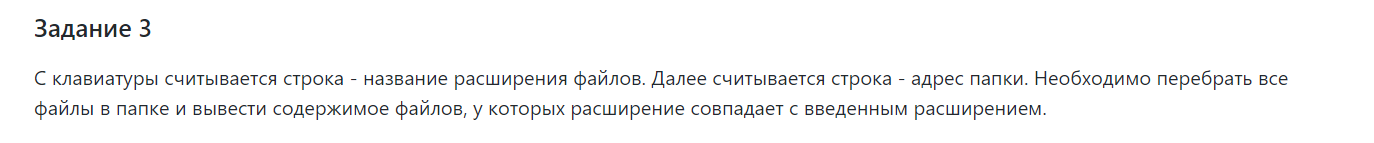
console.log(jsonString.mass[i]);

}

}







"use strict"

let fs = require("fs");

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

let expansion = readlineSync.question("Input expansion: ");

let fullExp = "." + expansion;

let folder = readlineSync.question("Input folder adress: ");

let arr = fs.readdirSync(folder);

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

if (arr[i].includes(fullExp)) {

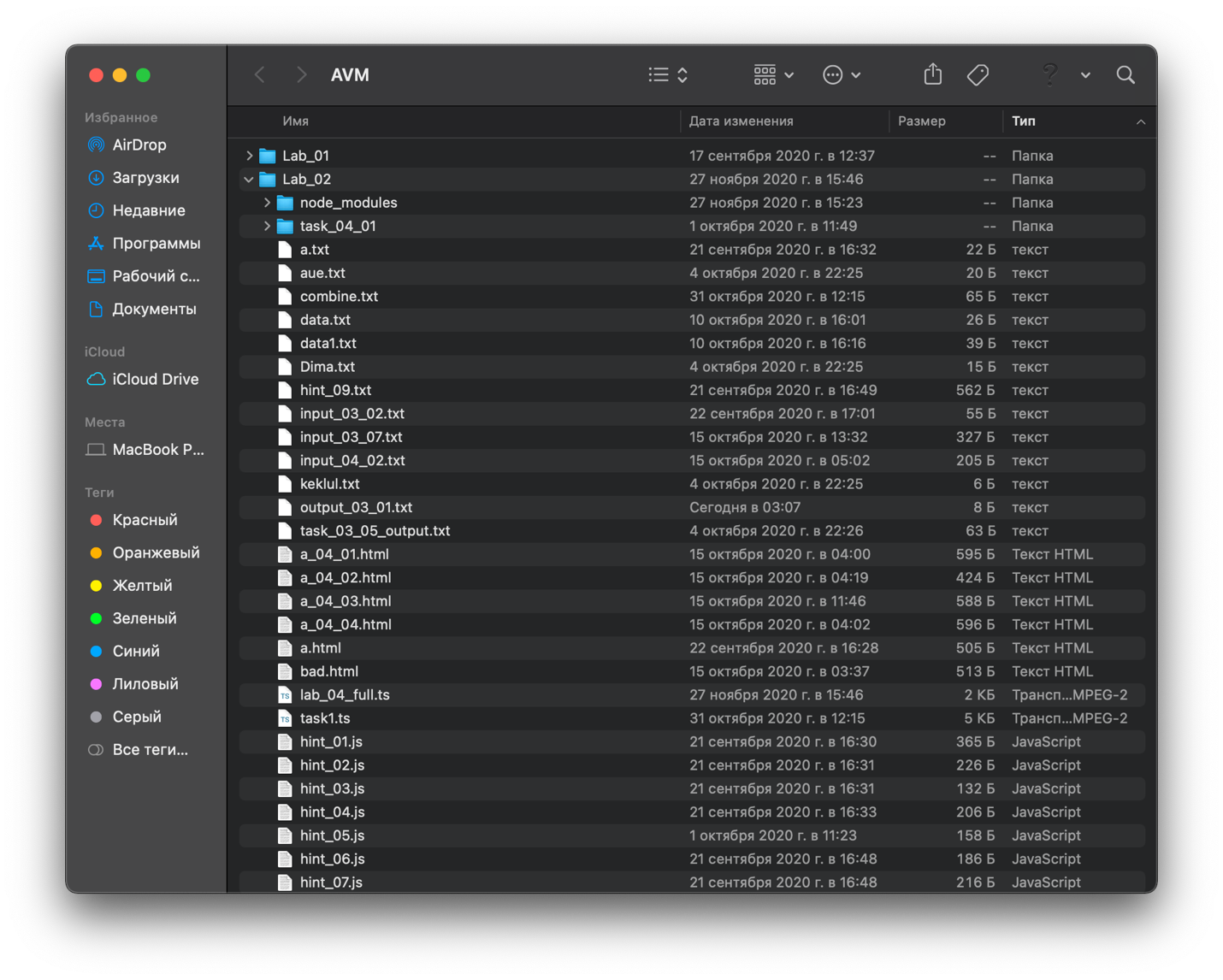
console.log("file name: " + arr[i]);

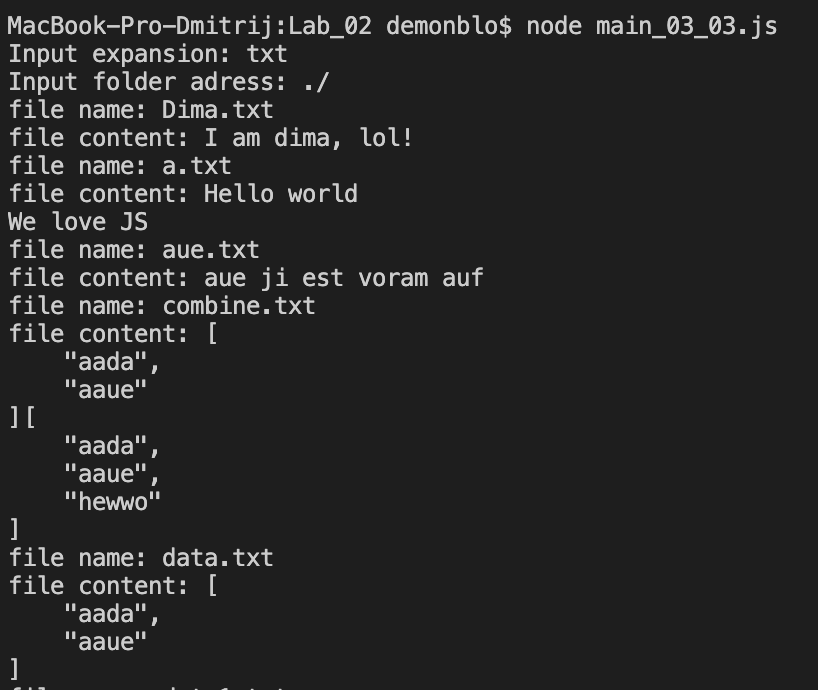
let contentString = fs.readFileSync(arr[i], "utf8");

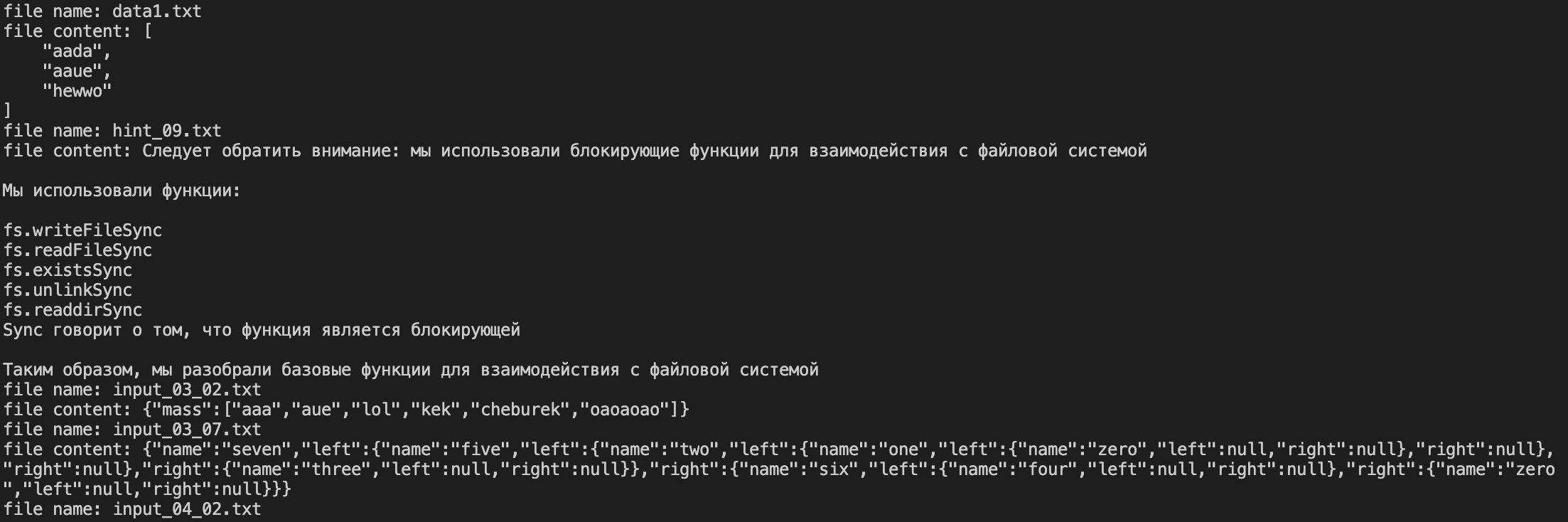
console.log("file content: " + contentString);

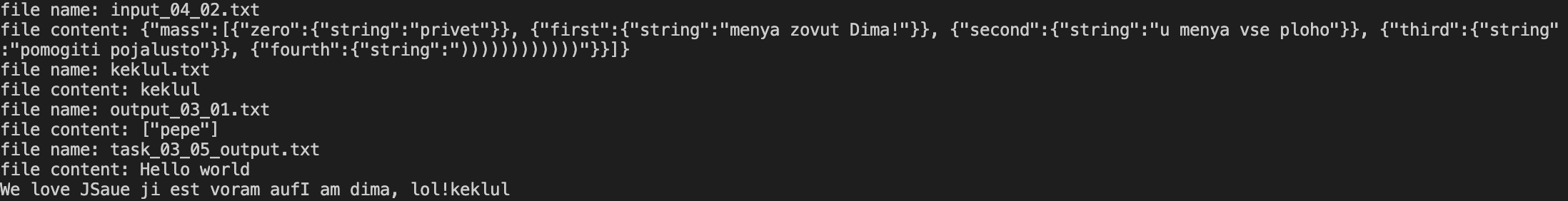
}

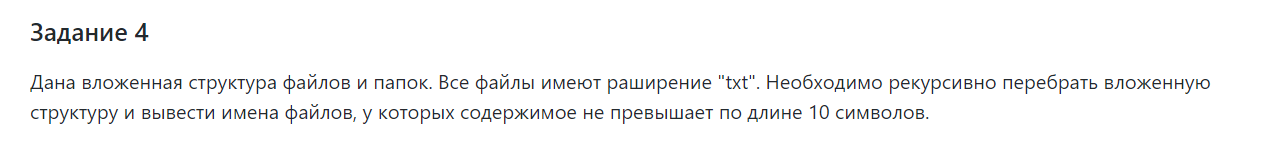
}











"use strict"

let expansion = ".txt";

let fs = require("fs");

function check\_out(folder, spaces) {

let arr = fs.readdirSync(folder);

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

if (arr[i].includes(expansion)) {

let fileName = folder + "/" + arr[i];

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

if (contentString.length <= 10) {

console.log(spaces + arr[i]);

}

}

if (!arr[i].includes(expansion) && arr[i] != ".DS\_Store" && !(arr[i].includes(".js"))) {

console.log(spaces + arr[i]);

let nextDir = "" + folder + "/" + arr[i];

check\_out(nextDir, spaces + " ");

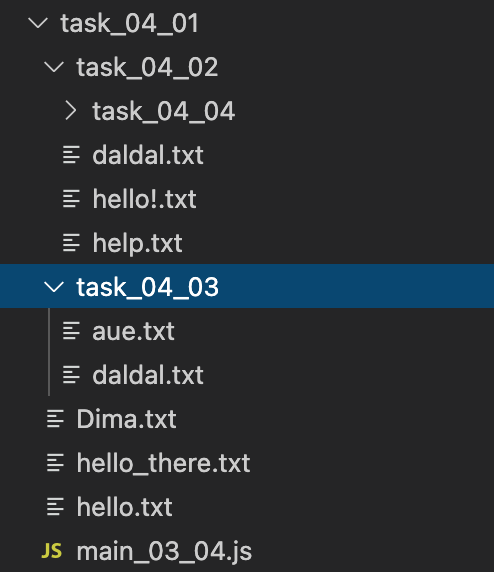
}

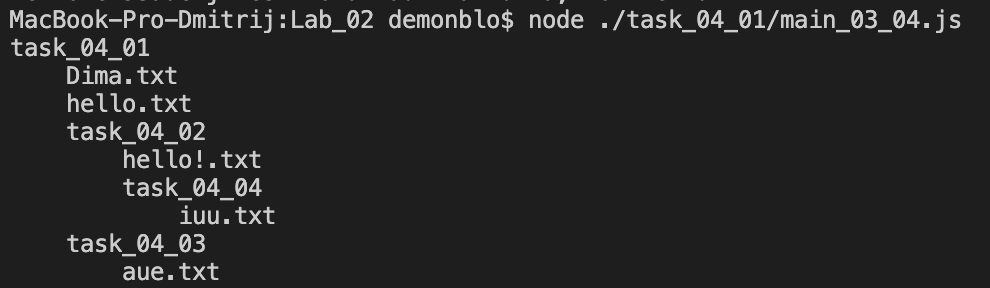
}

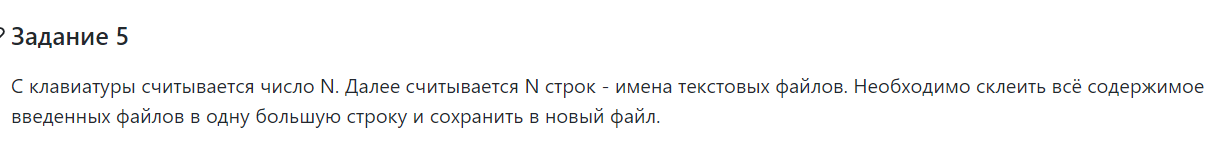
}

console.log("task\_04\_01");

check\_out("./task\_04\_01", " ");







"use strict"

let fs = require("fs");

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

let N = readlineSync.question("Input N: ");

let outp = "";

let arr = fs.readdirSync("./");

for (let i = 0; i < N; i++) {

let fileName = readlineSync.question("Input file's name: ");

if(!(arr.includes(fileName))) {

console.log("There is no such file!");

return -1;

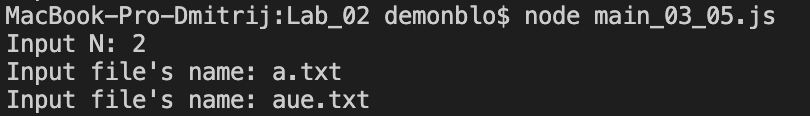
}

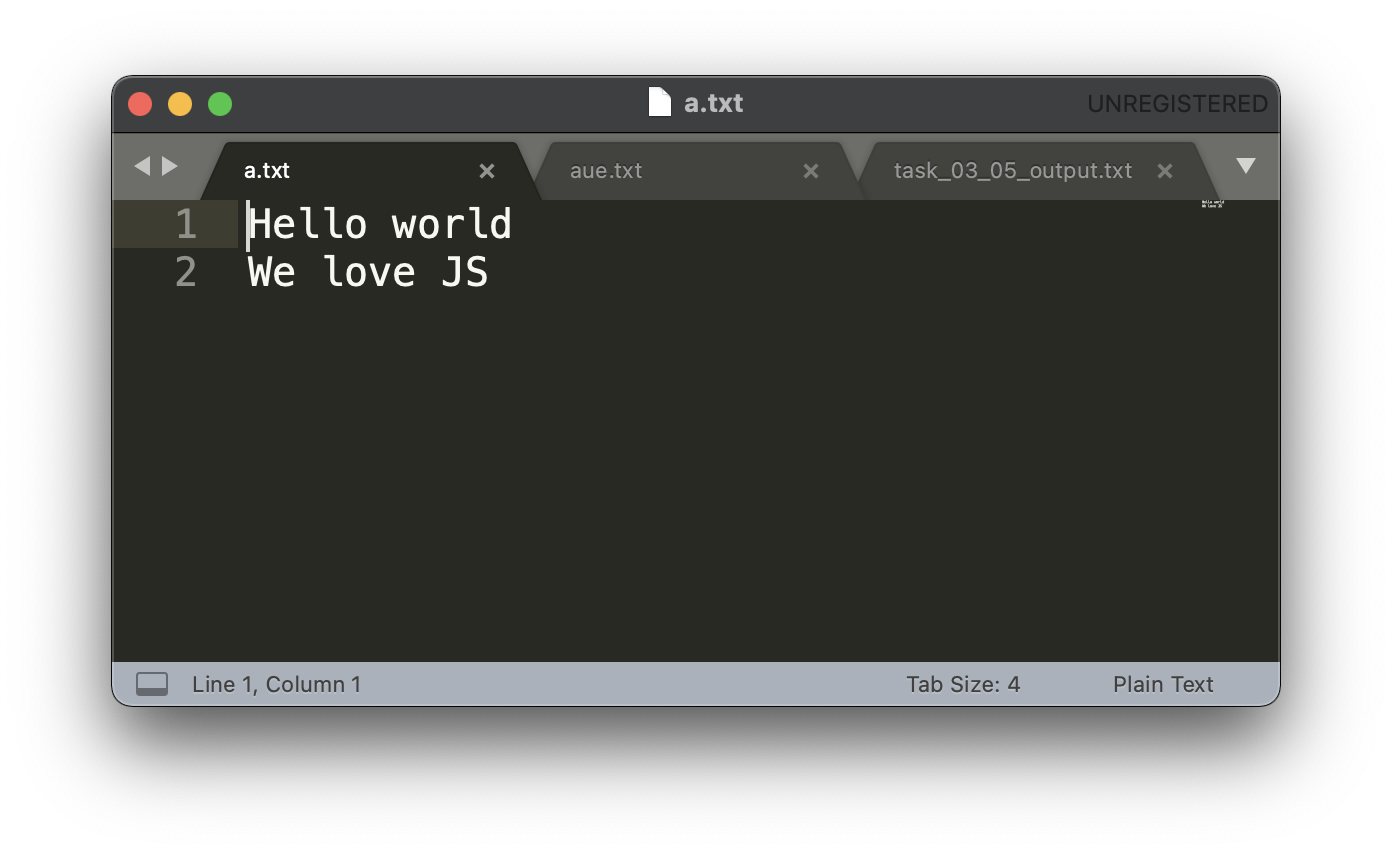
let contentString = fs.readFileSync(fileName, "utf8");

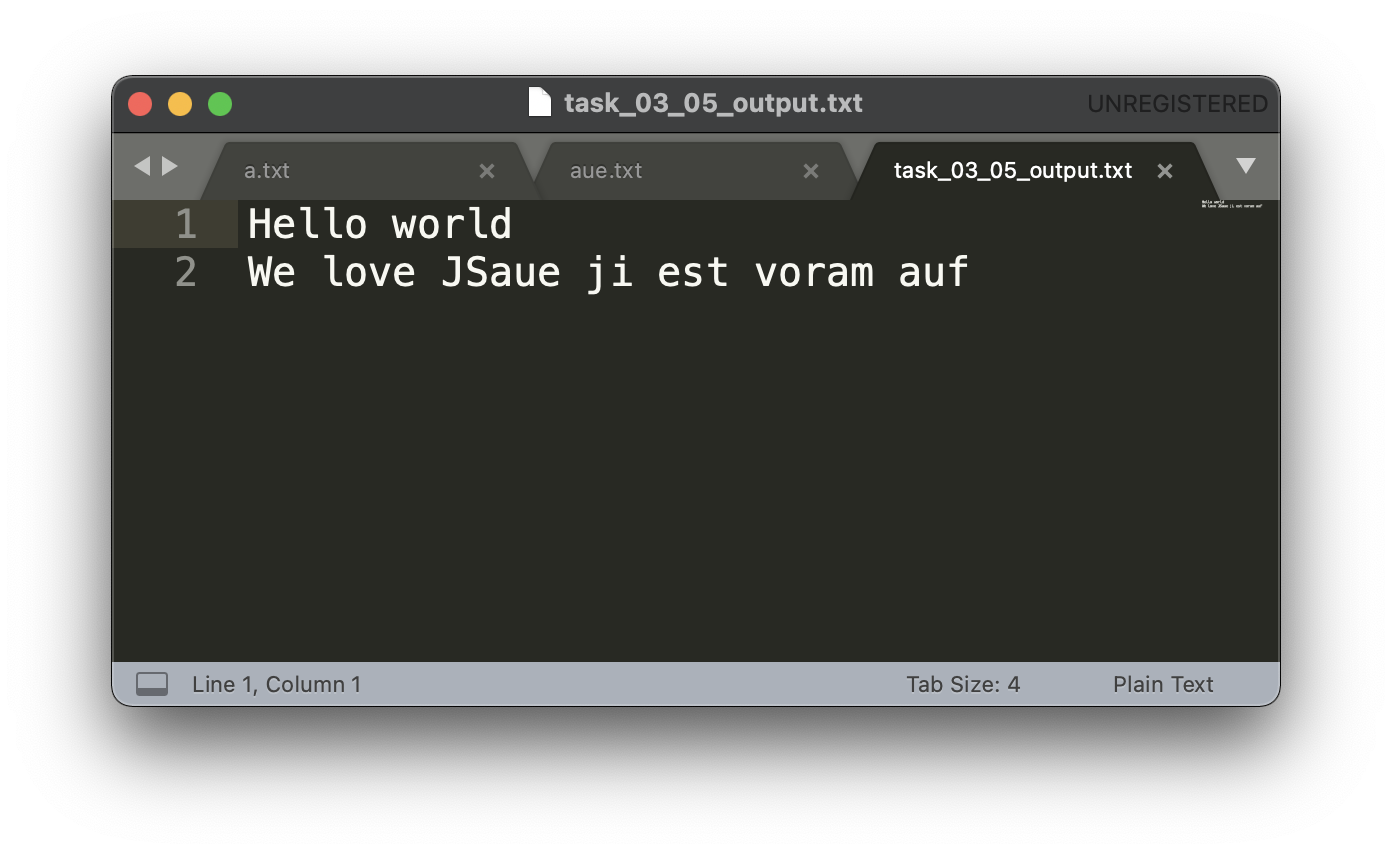
outp += contentString;

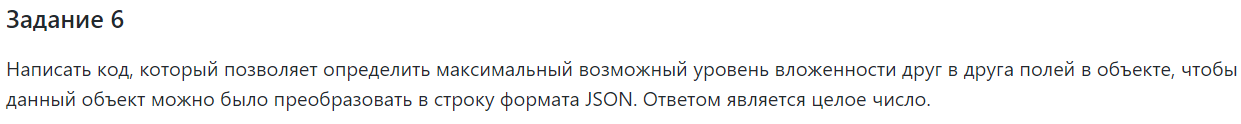
}

fs.writeFileSync("task\_03\_05\_output.txt", outp);









"use strict"

let studentObject = {};

let right\_now = studentObject;

for(let i = 1; i < 10000; i++) {

let cur\_obj = {};

cur\_obj.count = i;

let cur\_1 = right\_now;

cur\_1.link = cur\_obj;

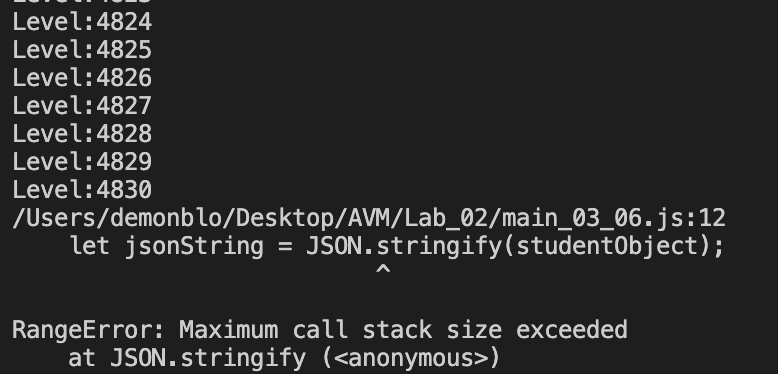
right\_now = cur\_1.link;

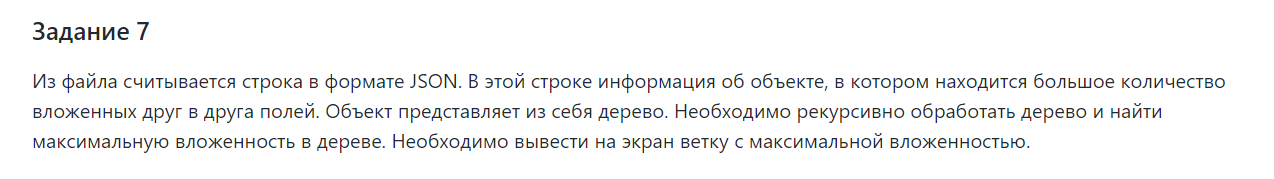
let jsonString = JSON.stringify(studentObject);

console.log("Level:" + i);

}

console.log(jsonString);





"use strict"

let fs = require("fs");

let node\_0 = {name:"zero", left:null, right:null};

let node\_1 = {name:"one", left:node\_0, right:null};

let node\_2 = {name:"two", left:node\_1, right:null};

let node\_3 = {name:"three", left:null, right:null};

let node\_4 = {name:"four", left:null, right:null};

let node\_5 = {name:"five", left:node\_2, right:node\_3};

let node\_6 = {name:"six", left:node\_4, right:node\_0};

let node\_7 = {name:"seven", left:node\_5, right:node\_6};

fs.writeFileSync("input\_03\_07.txt", JSON.stringify(node\_7));

const tree = JSON.parse(fs.readFileSync("input\_03\_07.txt", "utf8"));

let max\_branch = "";

function find\_longest\_branch(link, branch) {

branch += link.name + " ";

if (branch > max\_branch) {

max\_branch = branch;

}

if(link.right != null) {

find\_longest\_branch(link.right, branch);

}

if(link.left != null) {

find\_longest\_branch(link.left, branch);

}

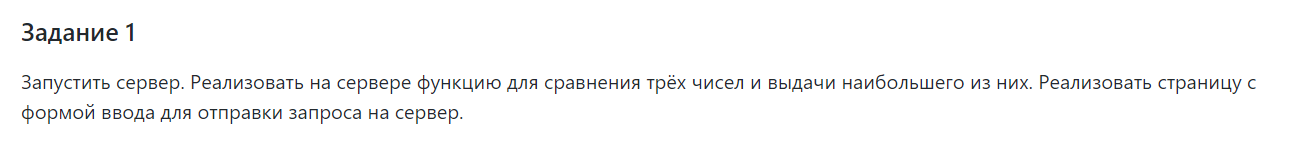
}

find\_longest\_branch(tree, "");

console.log("The longest branch: " + max\_branch);



ЧАСТЬ 2.



"use strict"

const express = require("express");

const fs = require("fs");

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;

let max = 0;

const aInt = parseInt(a);

const bInt = parseInt(b);

const cInt = parseInt(c);

if (aInt >= bInt && aInt >= cInt) {

max = aInt;

}

else if (bInt >= aInt && bInt >= cInt) {

max = bInt;

}

else if (cInt >= aInt && cInt >= bInt) {

max = cInt;

}

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

response.end(answerJSON);

});

<!DOCTYPE html>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Страница A</title>

</head>

<body>

<h1>Страница A</h1>

<form method="GET" action="/find/max">

<p>Введите A</p>

<input name="a" spellcheck="false" autocomplete="off">

<p>Введите B</p>

<input name="b" spellcheck="false" autocomplete="off">

<p>Введите C</p>

<input name="c" spellcheck="false" autocomplete="off">

<br>

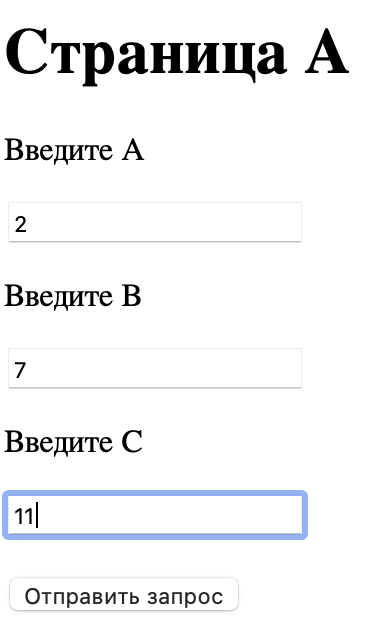
<br>

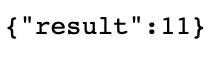
<input type="submit" value="Отправить запрос">

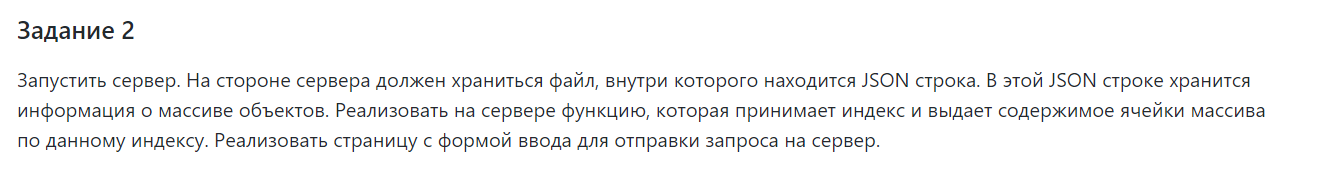
</form>

</body>

</html>







"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);

let inputString = fs.readFileSync("input\_04\_02.txt", "utf8");

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/byindex", function(request, response) {

let index = request.query.index;

let indexInt = parseInt(index);

let jSonString = JSON.parse(inputString);

let answ = jSonString.mass[indexInt];

let answerJSON = JSON.stringify({result: answ});

response.end(answerJSON);

});

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Страница A</title>

</head>

<body>

<h1>Страница A</h1>

<form method="GET" action="/find/byindex">

<p>Введите индекс</p>

<input name="index" spellcheck="false" autocomplete="off">

<br>

<br>

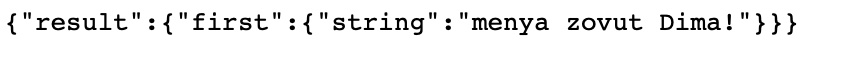
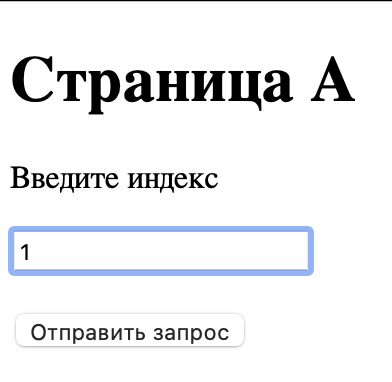
<input type="submit" value="Отправить запрос">

</form>

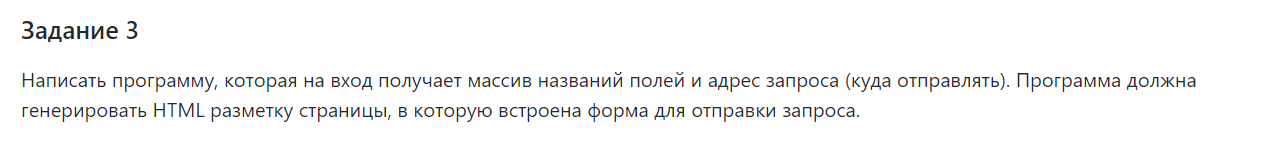
</body>

</html>

</html>



Исходный файл: {"mass":[{"zero":{"string":"privet"}}, {"first":{"string":"menya zovut Dima!"}}, {"second":{"string":"u menya vse ploho"}}, {"third":{"string":"pomogiti pojalusto"}}, {"fourth":{"string":"))))))))))))"}}]}



"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);

function build\_html(fields, address){

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

<html>\n\

<head>\n\

<meta charset = "UTF-8">\n\

<title>Результат</title>\n\

</head>\n\

<body>\n\

<h1> Реузльтат</h1>\n\

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

for (let field of fields){

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

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

}

html\_file += ' <br>\n\

<br>\n\

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

</form>\n\

</body>\n\

</html>'

return html\_file;

}

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

const fileName = request.query.p;

console.log(fileName);

if (fs.existsSync(fileName)) {

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

response.end(contentString);

} else {

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

response.end(contentString);

}

});

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

let fields = request.query.fields;

let addresses = request.query.address;

fields = fields.split(",");

let result\_html = build\_html(fields, addresses);

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

let arr = request.query;

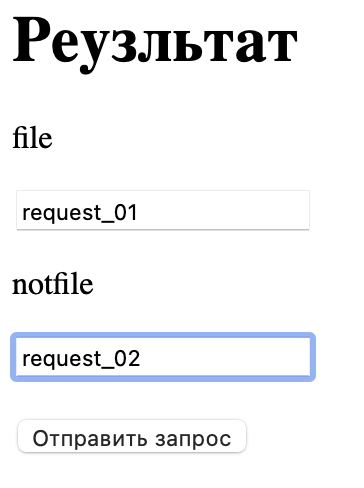
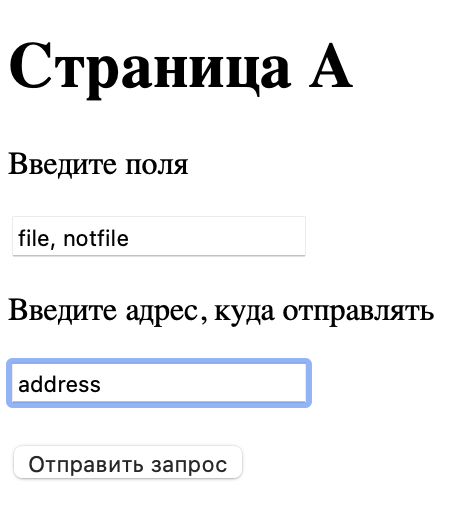
let string = JSON.stringify(arr);

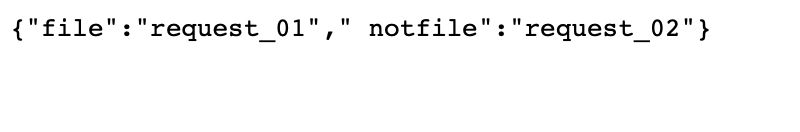
response.end(string);

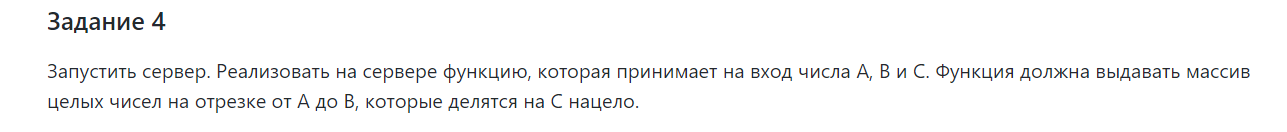
});

response.end(result\_html);

})







"use strict"

const express = require("express");

const fs = require("fs");

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/mass", function(request, response) {

const a = request.query.a;

const b = request.query.b;

const c = request.query.c;

let mass = [];

const aInt = parseInt(a);

const bInt = parseInt(b);

const cInt = parseInt(c);

for (let i = aInt; i <= bInt; i++) {

if (i % cInt === 0) {

mass.push(i);

}

}

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

response.end(answerJSON);

});

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Страница A</title>

</head>

<body>

<h1>Страница A</h1>

<form method="GET" action="/find/mass">

<p>Введите A</p>

<input name="a" spellcheck="false" autocomplete="off">

<p>Введите B</p>

<input name="b" spellcheck="false" autocomplete="off">

<p>Введите C</p>

<input name="c" spellcheck="false" autocomplete="off">

<br>

<br>

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

</form>

</body>

</html>

