



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 «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
По курсу: "Архитектура ЭВМ"

Студент _____ Наместник Анастасия _____
Группа _____ ИУ7-53Б _____
Название предприятия _____ МГТУ им. Н. Э. Баумана, каф. ИУ7 _____
Тема _____ Работа с fs, readline-sync и express. _____

Студент:	_____	Наместник А.А..
	подпись, дата	Фамилия, И.О.
Преподаватель:	_____	Попов А. Ю.
	подпись, дата	Фамилия, И. О.

TASK_1.

Цель работы:

- Изучить fs, readline-sync и express;
- Написать программы, для демонстрации изученного материала;
- Научиться взаимодействовать с пользователем через консоль;
- Изучить и реализовать работу с файлами;
- Изучить формат JSON и научиться работать с ним.

Задание 1

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

Задание 2

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.

Задание 3

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.

Задание 4

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

Задание 5

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.

Задание 6

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.

Задание 7

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 1 — Код программы. TASK_1. Главная функция main

```
1  function main() {
2
3      const choice = readline.question("Choose the task number: ");
4      const functions = [Task1, Task2, Task3, Task4, Task5, Task7];
5
6      if (choice > 7 || choice < 1) {
7          console.log("\nWrong option!")
8          return;
9      }
10
11     functions[choice - 1]();
12
13 }
```

Листинг 2 — Код программы. TASK_1. Реализация заданий

```
1  "use strict";
2  const readline = require('readline-sync');
3  const fs = require("fs");
4
5  function Task1() {
6      const file = "file1.txt";
7
8      const N = readline.question("Enter N: ");
9      let array = [];
10     let string;
11
12     for (let i = 0; i < N; i++) {
13         string = readline.question("Enter string: ");
14         if (!(string.length % 2))
15             array.push(string);
16     }
17
18     const jsonString = JSON.stringify(array, null, 4);
19     fs.writeFileSync(file, jsonString);
20 }
21
22 function isVowel(letter) {
23     return ['a', 'e', 'i', 'o', 'u'].indexOf(letter.toLowerCase()) !== -1
24 }
25
26 function Task2() {
27     const file = "file2.txt";
28
29     const strings = fs.readFileSync(file, "utf-8");
30     //С помощью JSON.parse мы получаем объект из строки JSON
```

```

31     const obj = JSON.parse(strings);
32
33     console.log(strings);
34     let count_vowels = 0;
35
36     for (let i = 0; i < obj.length; i++) {
37         count_vowels = 0
38         for (let j = 0; j < obj[i].length; j++)
39             if (isVowel(obj[i][j]))
40                 count_vowels += 1;
41         if (count_vowels === obj[i].length)
42             console.log(obj[i]);
43     }
44 }
45
46 function Task3() {
47
48     const extension = readline.question("Enter an extension: ");
49     const folder = readline.question("Enter the folder's name: ");
50
51     if (!(fs.existsSync(folder))) {
52         console.log("\nThe folder does not exist!");
53         return;
54     }
55
56     let f = fs.readdirSync(folder);
57     //console.log(f)
58
59     for (let i = 0; i < f.length; i++) {
60         let file = f[i].split('.');
61         //console.log(file)
62         if (file[1] === extension) {
63             let file_text = fs.readFileSync(folder + "/" + f[i], "utf-8");
64             console.log(file_text);
65         }
66     }
67 }
68 function Task4() {
69
70     //folder = "nested"
71     const folder = readline.question("Enter the folder's name: ");
72     Recursion(folder);
73 }
74
75 function Recursion(folder) {
76
77     if (!fs.existsSync(folder)) {

```

```

78     console.log("\nThe folder does not exist!");
79     return;
80 }
81
82 let f = fs.readdirSync(folder);
83 let file_text;
84 //console.log(f)
85
86 for (let i = 0; i < f.length; i++) {
87     let file = f[i].split('.');
88     if (file[file.length - 1] === "txt") {
89         file_text = fs.readFileSync(folder + "/" + f[i], "utf-8");
90         if (file_text.length <= 10) {
91             console.log("Path: ", folder + "/" + f[i]);
92         }
93     }
94     else {
95         //console.log(f[i])
96         Recursion(folder + "/" + f[i]);
97     }
98 }
99 }
100
101 function Task5() {
102     const file = "file3.txt";
103     fs.writeFileSync(file, "");
104
105     const N = readline.question("Enter N: ");
106     //let array = [];
107     let f;
108
109     for (let i = 0; i < N; i++) {
110         f = readline.question("Enter file name: ");
111         if (!fs.existsSync(f)) {
112             console.log("\nThe file does not exist!");
113             i -= 1;
114         }
115         else {
116             //array.push(f);
117             let file_text = fs.readFileSync(f, "utf-8");
118             fs.appendFileSync(file, file_text);
119         }
120     }
121 }
122
123 function Task7() {
124     const obj = {};

```

```

125     obj.x = 17;
126     obj.y = -45;
127     obj.z = 0;
128     obj.data = {};
129     obj.data.param1 = 10;
130     obj.data.param2 = 20;
131     obj.data.extra = {};
132     obj.data.extra.param3 = 30;
133     obj.data.extra.param4 = 40;
134
135     //С помощью формата JSON.stringify можно представить информацию об объе
        кте в виде строки
136     const jsonString = JSON.stringify(obj, null, 4);
137     console.log(jsonString);
138
139     //С помощью JSON.parse мы получаем объект из строки JSON
140     const obj_copy = JSON.parse(jsonString);
141     console.log(obj_copy);
142
143 }

```

Вывод:

- Были изучены fs, readline-sync и express;
- Были написаны программы, для демонстрации изученного материала;
- Было изучено и реализовано взаимодействие с пользователем через консоль;
- Было изучено и реализована работа с файлами;
- Был изучен формат JSON, а также реализована работа с ним.

Пример работы:

```

MBP-Anastasia:project3_4 anastasia$ node task3.js
Choose the task number: 1
Enter N: 5
Enter string: qwer
Enter string: qw rtyu
Enter string: a a a
Enter string: aaa bb c
Enter string: q
MBP-Anastasia:project3_4 anastasia$ █

```

Рисунок 0.1 — Пример работы программы

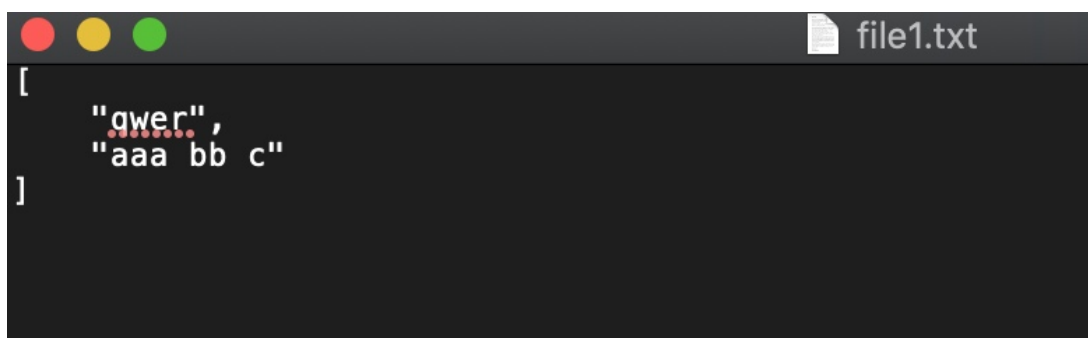


Рисунок 0.2 — Пример работы программы

TASK_2.

Цель работы:

- Научиться запускать собственные сервера;
- Изучить и реализовать хранение данных на стороне сервера;
- Реализовать генерацию HTML страниц;
- Изучить и реализовать взаимодействие с пользователем.

Задание 1

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.

Задание 2

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.

Задание 3

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.

Задание 4

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 3 — Код программы. TASK_2. Реализация заданий

```
1  "use strict";
2  const path = require("path");
3  const fs = require("fs");
4  //Фреймворк Express сам использует модуль http, но вместе с тем предоставля
   ет ряд готовых абстракций,
5  //которые упрощают создание сервера и серверной логики, в частности, обрабо
   тка отправленных форм, работа с куками, CORS и т.д.
6
7  // подключение express
8  const express = require("express");
9
10 function CreatePage(app, path, file) {
11   app.get(path, function (request, response) {
12     //const file = request.query.p;
```



```

13     //console.log(nameString);
14     if (fs.existsSync(file)) {
15         const contentFile = fs.readFileSync(file, "utf8");
16         response.end(contentFile);
17     } else {
18         const contentFile = fs.readFileSync("bad.html", "utf8");
19         response.end(contentFile);
20     }
21 });
22 }
23
24 function Task1(app) {
25     CreatePage(app, "/compare", "compare.html")
26
27     app.get("/compare/numbers", function (request, response) {
28         const a = request.query.a;
29         const b = request.query.b;
30         const c = request.query.c;
31         const aInt = parseInt(a);
32         const bInt = parseInt(b);
33         const cInt = parseInt(c);
34
35         if (!aInt || !bInt || !cInt) {
36             response.end("Input error!");
37             return;
38         }
39
40         let maxInt = cInt;
41         if (aInt >= bInt && aInt >= cInt)
42             maxInt = aInt;
43         else if (bInt >= aInt && bInt >= cInt)
44             maxInt = bInt;
45
46         const answerJSON = JSON.stringify({ result: maxInt });
47         response.end(answerJSON);
48     });
49 }
50
51 function Task2(app) {
52     CreatePage(app, "/array", "array.html")
53
54     app.get("/array/objects", (request, response) => {
55         const index = request.query.index;
56         const indexInt = parseInt(index);
57
58         if (!indexInt) {
59             response.end("Input error!");

```

```

60         return;
61     }
62
63     const array = JSON.parse(fs.readFileSync("string.json"));
64
65     if (indexInt < 0 || indexInt > array.length) {
66         response.end("Input index error!");
67         return;
68     }
69
70     response.end("Index = " + indexInt + "\nElement = " +
71         array[indexInt - 1]);
72 }
73
74 function Task3(app) {
75     CreatePage(app, "/markup", "markup.html")
76
77     app.get("/markup/generate", (request, response) => {
78         const field_names = request.query.fields;
79         const address = request.query.address;
80         const fields = field_names.split(' ');
81
82         const pathBegin = "start.txt";
83         const pathEnd = "end.txt";
84
85         const fileBegin = fs.readFileSync(pathBegin, "utf8")
86         const fileEnd = fs.readFileSync(pathEnd, "utf8")
87
88         let fileContent = '<form method="GET" action="${address}">\n'
89         for (let i = 0; i < fields.length; i++) {
90             fileContent += '
91 <h2><p>Введите ${fields[i]}</p>\n\
92 <input name = "${fields[i]}" spellcheck = "false" autocomplete = "off"
93     class="colortextInput">'
94         }
95
96         fileContent += '\<br>\n<br>\n<input type = "submit" value =
97             "Отправить" class="colortextSubmit"><h2>\n\
98 </form >\n'
99
100         response.end(fileBegin + fileContent + fileEnd);
101     });
102 }
103
104 function Task4(app) {
105     CreatePage(app, "/interval", "interval.html")

```

```

104
105     app.get("/interval/res", (request, response) => {
106         const a = request.query.a;
107         const b = request.query.b;
108         const c = request.query.c;
109         const aInt = parseInt(a);
110         const bInt = parseInt(b);
111         const cInt = parseInt(c);
112
113         if (!aInt || !bInt || !cInt) {
114             response.end("Input error!");
115             return;
116         }
117
118         let arr = [];
119         for (let i = aInt; i <= bInt; i++)
120             if (!(i % cInt))
121                 arr.push(i)
122         if (!arr[0]) {
123             response.end("Array is empty");
124             return;
125         }
126
127         response.end("Array = " + arr);
128     });
129 }
130
131 function Main() {
132     // создаем объект приложения
133     const app = express();
134     //Настройка порта
135     const port = 5015;
136     app.listen(port);
137     console.log("My server on port " + port);
138
139     //app.use(express.static(path.join(__dirname, '..', 'public')));
140     CreatePage(app, "/me/page", "main.html")
141
142     Task1(app);
143     Task2(app);
144     Task3(app);
145     Task4(app);
146
147 }

```

```

1      <!DOCTYPE html>
2      <html lang="ru">
3
4      <head>
5          <meta charset="UTF-8">
6          <link rel="stylesheet" type="text/css" href="../main.css">
7          <title>Выбор задания</title>
8          <style>
9              .colortextInput {
10                  background: -webkit-gradient(linear, 0 0, 0 100%,
11                      from(#D0ECF4), to(#D0ECF4), color-stop(0.5, rgb(0, 0, 0)));
12                  filter:
13                      progid:DXImageTransform.Microsoft.gradient(startColorstr='#00BBD6',
14                          endColorstr='#EBFFFF');
15                  padding: 5px 100px;
16                  font-size: 20px;
17                  text-align: center;
18                  color: rgb(13, 202, 38);
19                  -webkit-border-radius: 5px;
20                  border-radius: 5px;
21                  border: 1px solid #666;
22              }
23
24              .colortextSubmit {
25                  background: -webkit-gradient(linear, 0 0, 0 100%,
26                      from(#D0ECF4), to(#D0ECF4), color-stop(0.5, rgb(0, 0, 0)));
27                  filter:
28                      progid:DXImageTransform.Microsoft.gradient(startColorstr='#00BBD6',
29                          endColorstr='#EBFFFF');
30                  padding: 5px 30px;
31                  font-size: 20px;
32                  color: rgb(13, 202, 38);
33                  -webkit-border-radius: 5px;
34                  border-radius: 5px;
35                  border: 1px solid #666;
36              }
37
38              body {
39                  font-family: Geneva, Arial, Helvetica, sans-serif;
40                  text-align: center;
41                  padding: 50px;
42                  /* Цвет фона */
43                  background: black;
44              }
45
46              h1 {
47                  color: rgb(0, 0, 0);

```

```

42     }
43
44     h2 {
45         color: rgb(13, 202, 38);
46     }
47 </style>
48 </head>
49
50 <body>
51     <div id="header">
52         <!-- <h1 class="header-title">Выбор задания</h1> -->
53         <br>
54     </div>
55
56     <div id="main">
57
58         <h2>#1</h2>
59         <form method="GET" action="/compare/">
60             <p><button type="submit" class="colortextSubmit">Найти наибольш
61                 ее среди трех чисел</button></p>
62
63         </form>
64
65         <h2>#2</h2>
66         <form method="GET" action="/array/">
67             <p><button type="submit" class="colortextSubmit">Получить значе
68                 ние массива по индексу</button></p>
69
70         </form>
71
72         <h2>#3</h2>
73         <form method="GET" action="/markup/">
74             <p><button type="submit" class="colortextInput">Получить размет
75                 ку</button></p>
76
77         </form>
78
79         <h2>#4</h2>
80         <form method="GET" action="/interval/">
81             <p><button type="submit" class="colortextInput">Получить массив
82                 чисел</button></p>
83
84         </form>
85
86     </div>

```

```

83 <p></p>
85 <p></p>
87 </body>
88
89 </html>

```

Листинг 5 — Код программы. TASK_2. Страница с заданием 1

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <link rel="stylesheet" type="text/css" href="../task1.css">
7   <title>Поиск максимума</title>
8   <style>
9     .colortextInput {
10       background: -webkit-gradient(linear, 0 0, 0 100%, from(#D0ECF4),
11         to(#D0ECF4), color-stop(0.5, rgb(0, 0, 0)));
12       filter:
13         progid:DXImageTransform.Microsoft.gradient(startColorstr='#00BBD6',
14         endColorstr='#EBFFFF');
15       padding: 5px 10px;
16       font-size: 20px;
17       text-align: center;
18       color: rgb(13, 202, 38);
19       -webkit-border-radius: 5px;
20       border-radius: 5px;
21       border: 1px solid #666;
22     }
23
24     .colortextSubmit {
25       background: -webkit-gradient(linear, 0 0, 0 100%, from(#D0ECF4),
26         to(#D0ECF4), color-stop(0.5, rgb(0, 0, 0)));
27       filter:
28         progid:DXImageTransform.Microsoft.gradient(startColorstr='#00BBD6',
29         endColorstr='#EBFFFF');
30       padding: 5px 40px;
31       font-size: 20px;
32       color: rgb(13, 202, 38);
33       -webkit-border-radius: 5px;

```

```

28     border-radius: 5px;
29     border: 1px solid #666;
30 }
31
32 body {
33     font-family: Geneva, Arial, Helvetica, sans-serif;
34     text-align: center;
35     padding: 50px;
36     /* Цвет фона */
37     background: black;
38 }
39
40 h1 {
41     color: rgb(0, 0, 0);
42 }
43
44 h2 {
45     color: rgb(13, 202, 38);
46 }
47 </style>
48 </head>
49
50 <body>
51     <div id="header">
52         <!-- <h1 class="header-title">Максимум</h1> -->
53     </div>
54
55     <div id="main">
56         <form method="GET" action="/compare/numbers">
57             <h2>
58                 <p>Введите первое число</p>
59                 <h2>
60                     <input name="a" spellcheck="false" autocomplete="off"
61                         class="colortextInput">
62                     <p>Введите второе число</p>
63                     <input name="b" spellcheck="false" autocomplete="off"
64                         class="colortextInput">
65                     <p>Введите третье число</p>
66                     <input name="c" spellcheck="false" autocomplete="off"
67                         class="colortextInput">
68                     <br>
69                     <br>
67                     <input type="submit" value="Получить максимум"
68                         class="colortextSubmit">
69                 </form>
70             </div>

```

```

70     <p></p>
72     <p></p>
74
75
76 </body>
77
78 </html>
79
80 </html>

```

Листинг 6 — Код программы. TASK_2. Страница с заданием 2

```

1     <!DOCTYPE html>
2     <html>
3
4     <head>
5         <meta charset="UTF-8">
6         <link rel="stylesheet" type="text/css" href="../task2.css">
7         <title>Поиск объекта массива по индексу</title>
8         <style>
9             .colortextInput {
10                 background: -webkit-gradient(linear, 0 0, 0 100%,
11                     from(#D0ECF4), to(#D0ECF4), color-stop(0.5, rgb(0, 0,
12                     0)));
13                 filter:
14                     progid:DXImageTransform.Microsoft.gradient(startColorstr='#00BBD6',
15                     endColorstr='#EBFFFF');
16                 padding: 5px 24px;
17                 font-size: 20px;
18                 text-align: center;
19                 color: rgb(13, 202, 38);
20                 -webkit-border-radius: 5px;
21                 border-radius: 5px;
22                 border: 1px solid #666;
23             }
24
25             .colortextSubmit {
26                 background: -webkit-gradient(linear, 0 0, 0 100%,
27                     from(#D0ECF4), to(#D0ECF4), color-stop(0.5, rgb(0, 0,
28                     0)));

```



```

23         filter :
                progid:DXImageTransform.Microsoft.gradient(startColorstr='#00BBD6',
                endColorstr='#EBFFFF');
24         padding: 5px 40px;
25         font-size: 18px;
26         color: rgb(13, 202, 38);
27         -webkit-border-radius: 5px;
28         border-radius: 5px;
29         border: 1px solid #666;
30     }
31
32     body {
33         font-family: Geneva, Arial, Helvetica, sans-serif;
34         text-align: center;
35         padding: 50px;
36         /* Цвет фона */
37         background: black;
38     }
39
40     h1 {
41         color: rgb(0, 0, 0);
42     }
43
44     h2 {
45         color: rgb(13, 202, 38);
46     }
47 </style>
48 </head>
49
50 <body>
51     <div id="header">
52         <!-- <h1 class="header-title">Массив</h1> -->
53     </div>
54
55     <div id="main">
56         <form method="GET" action="/array/objects">
57             <h2>
58                 <p>Введите индекс</p>
59                 <h2>
60                     <input name="index" spellcheck="false"
61                         autocomplete="off" class="colortextInput">
62                     <br>
63                     <br>
64                     <input type="submit" value="Получить элемент массив
65                         a" class="colortextSubmit">

```

```

66     <p></p>
68     <p></p>
70
71
72 </body>
73
74 </html>

```

Листинг 7 — Код программы. TASK_2. Страница с заданием 3

```

1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta charset="UTF-8">
6      <link rel="stylesheet" type="text/css" href="../task3.css">
7      <title>Получить разметку</title>
8      <style>
9          .colortextInput {
10              background: -webkit-gradient(linear, 0 0, 0 100%,
11                  from(#D0ECF4), to(#D0ECF4), color-stop(0.5, rgb(0, 0, 0)));
12              filter:
13                  progid:DXImageTransform.Microsoft.gradient(startColorstr='#00BBD6',
14                      endColorstr='#EBFFFF');
15              padding: 5px 10px;
16              font-size: 20px;
17              text-align: center;
18              color: rgb(13, 202, 38);
19              -webkit-border-radius: 5px;
20              border-radius: 5px;
21              border: 1px solid #666;
22          }
23
24          .colortextSubmit {
25              background: -webkit-gradient(linear, 0 0, 0 100%,
26                  from(#D0ECF4), to(#D0ECF4), color-stop(0.5, rgb(0, 0, 0)));
27              filter:
28                  progid:DXImageTransform.Microsoft.gradient(startColorstr='#00BBD6',
29                      endColorstr='#EBFFFF');
30              padding: 5px 40px;
31              font-size: 20px;

```

```

26         color: rgb(13, 202, 38);
27         -webkit-border-radius: 5px;
28         border-radius: 5px;
29         border: 1px solid #666;
30     }
31
32     body {
33         font-family: Geneva, Arial, Helvetica, sans-serif;
34         text-align: center;
35         padding: 50px;
36         /* Цвет фона */
37         background: black;
38     }
39
40     h1 {
41         color: rgb(0, 0, 0);
42     }
43
44     h2 {
45         color: rgb(13, 202, 38);
46     }
47 </style>
48 </head>
49
50 <body>
51     <div id="header">
52         <!-- <h1 class="header-title">Разметка</h1> -->
53     </div>
54
55     <div id="main">
56         <form method="GET" action="/markup/generate">
57             <h2>
58                 <p>Введите поля</p>
59                 <h2>
60                     <input name="fields" spellcheck="false"
61                         autocomplete="off" class="colortextInput">
62                     <p>Введите адрес</p>
63                     <input name="address" spellcheck="false"
64                         autocomplete="off" class="colortextInput">
65                     <br>
66                     <br>
67                     <input type="submit" value="Получить разметку"
68                         class="colortextSubmit">
69                 </h2>
69             </h2>
69         </form>
69     </div>

```

```

68 <p></p>
70 <p></p>
72
73
74 </body>
75
76 </html>

```

Листинг 8 — Код программы. TASK_2. Страница с заданием 4

```

1
2 <!DOCTYPE html>
3 <html>
4
5 <head>
6     <meta charset="UTF-8">
7     <link rel="stylesheet" type="text/css" href="../task4.css">
8     <title>Массив чисел на отрезке от A до B</title>
9     <style>
10         .colortextInput {
11             background: -webkit-gradient(linear , 0 0, 0 100%,
12                 from(#D0ECF4), to(#D0ECF4), color-stop(0.5, rgb(0, 0,
13                 0)));
14             filter:
15                 progid:DXImageTransform.Microsoft.gradient(startColorstr='
16                 #00BBD6',
17                 endColorstr=' #EBFFFF ');
18             padding: 5px 10px;
19             font-size: 20px;
20             text-align: center;
21             color: rgb(13, 202, 38);
22             -webkit-border-radius: 5px;
23             border-radius: 5px;
24             border: 1px solid #666;
25         }
26
27         .colortextSubmit {
28             background: -webkit-gradient(linear , 0 0, 0 100%,
29                 from(#D0ECF4), to(#D0ECF4), color-stop(0.5, rgb(0, 0,
30                 0)));

```

```

24         filter :
                progid : DXImageTransform . Microsoft . gradient ( startColorstr = '#00BBD6',
                endColorstr = '#EBFFFF' );
25         padding : 5px 40px;
26         font-size : 20px;
27         color : rgb(13, 202, 38);
28         -webkit-border-radius : 5px;
29         border-radius : 5px;
30         border : 1px solid #666;
31     }
32
33     body {
34         font-family : Geneva , Arial , Helvetica , sans-serif;
35         text-align : center;
36         padding : 50px;
37         /* Цвет фона */
38         background : black;
39     }
40
41     h1 {
42         color : rgb(0, 0, 0);
43     }
44
45     h2 {
46         color : rgb(13, 202, 38);
47     }
48 </style>
49 </head>
50
51 <body>
52     <div id="header">
53         <!-- <h1 class="header-title">Массив чисел</h1> -->
54     </div>
55
56     <div id="main">
57         <form method="GET" action="/interval/res">
58             <h2>
59                 <p>Введите A</p>
60                 <h2>
61                     <input name="a" spellcheck="false"
62                         autocomplete="off" class="colortextInput">
63                     <p>Введите B</p>
64                     <input name="b" spellcheck="false"
65                         autocomplete="off" class="colortextInput">
66                     <p>Введите C</p>
67                     <input name="c" spellcheck="false"
68                         autocomplete="off" class="colortextInput">

```

```

66         <br>
67         <br>
68         <input type="submit" value="Получить числа"
        class="colortextSubmit">
69     </form>
70 </div>
71 <p></p>
73 <p></p>
75
76
77 </body>
78
79 </html>

```

Вывод:

- Мы запустили собственные сервера;
- Мы изучили и реализовали хранение данных на стороне сервера;
- Реализовали генерацию HTML страниц;
- Изучили и реализовали взаимодействие с пользователем.

Пример работы:



Рисунок 0.1 — Пример работы программы

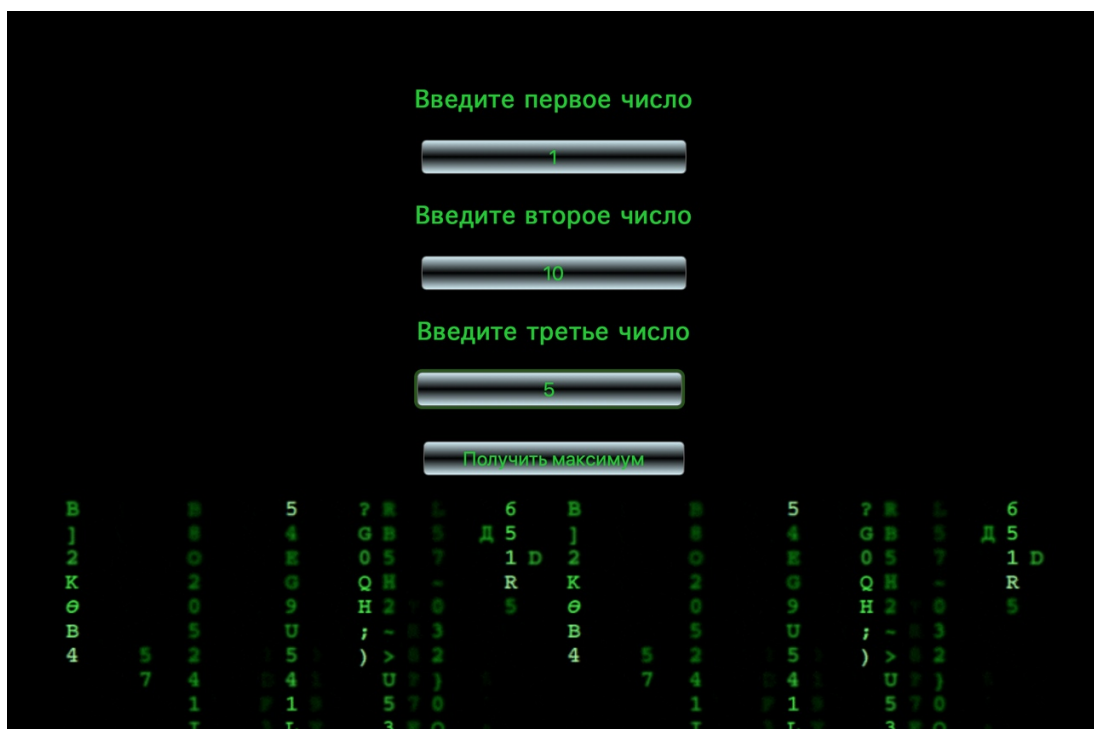


Рисунок 0.2 — Пример работы программы

```
{ "result": 10 }
```

Рисунок 0.3 — Пример работы программы