

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Алтайский государственный технический университет им. И. И.
Ползунова»
Университетский технологический колледж Кафедра «Информационные
систем в экономике»

Отчет защищен с оценкой.
Преподаватель С. В. Умбетов « » 2025 г.
« »2025г.

Отчёт по лабораторной работе №1

Студент группы 1ИСП22	<u>В.К.Мундусов</u>
Преподаватель ассистент, к. т. н.	<u>С. В. Умбетов</u>

БАРНАУЛ 2025

Лабораторная работа №1

Вариант - 15

Цели и задачи работы:

Создать дешифратор ROT13, используя html, css и js.

Страница должна быть валидной стандарту HTML5

ROT13 — это простой шифр замены букв, который заменяет букву буквой, находящейся через 13 букв после неё в алфавите. Если в строку включены цифры или специальные символы, их следует вернуть в исходном виде. Смещать следует только буквы латинского/английского алфавита.

Задание принял: В.К.Мундусов



Ход работы

```
1  <!DOCTYPE html>
2  <html lang="ru">
3    <head>
4      <title>лаба 1</title>
5      <meta charset="UTF-8" />
6      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7      <link rel="stylesheet" href="styles.css" />
8    </head>
9    <body>
10     <div class="container">
11       <h1>Дешифратор ROT13</h1>
12       <section class="text-section">
13         <div class="input-encode-block">
14           <h2>Зашифрованный текст</h2>
15           <textarea
16             id="input-encode-text"
17             placeholder="Введите зашифрованный текст..."
18           ></textarea>
19         </div>
20
21         <div class="output-decode-block">
22           <h2>Расшифрованный текст:</h2>
23           <p id="output-decode-text"></p>
24         </div>
25       </section>
26     </div>
27
28     <script src="script.js"></script>
29   </body>
30 </html>
```

Рисунок 1 – html код

styles.css > ...

```
1  .container {
2      display: flex;
3      align-items: center;
4      min-height: 90vh;
5      flex-direction: column;
6      justify-content: center;
7  }
8
9  h1 {
10     color: ■ #333;
11     font-family: "Arial", sans-serif;
12     font-size: 30px;
13 }
14
15 h2 {
16     margin-block-end: 0px;
17     margin-left: 10px;
18     color: ■ #333;
19     font-family: "Arial", sans-serif;
20     font-size: 15px;
21 }
22
23 #input-encode-text {
24     width: 45vw;
25     height: 60vh;
26     padding: 8px;
27     margin: 8px;
28     border: none;
29     resize: none;
30     box-shadow: 0px 0px 5px 1px ■ rgb(217 217 217);
31     background-color: □ rgba(251, 252, 254);
32     border-radius: 3%;
```

Рисунок 2 – css код

```
32     border-radius: 3%;
33     color: ■ #333;
34     font-family: monospace;
35     font-size: 15px;
36 }
37
38 #output-decode-text {
39     width: 45vw;
40     height: 57.5vh;
41     padding: 8px;
42     margin: 8px;
43     box-shadow: 0px 0px 5px 1px ■ rgb(179 140 170 / 60%);
44     background-color: □ rgba(251, 238, 248);
45     border-radius: 3%;
46     color: ■ #333;
47     font-family: "Arial", sans-serif;
48     font-size: 15px;
49     white-space: pre-wrap;
50     word-break: break-word;
51     overflow-y: auto;
52     font-family: monospace;
53     color: ■ rgb(129 67 151 / 83%);
54 }
55
56 .text-section {
57     display: flex;
58     flex-direction: row;
59 }
```

Рисунок 3 – CSS код

```
61   textarea:focus {
62   |   outline: none;
63   }
64
65   @media (min-width: 350px) and (max-width: 1000px) {
66   |   .text-section {
67   |       display: flex;
68   |       flex-direction: column;
69   |   }
70
71   |   #input-encode-text {
72   |       width: 80vw;
73   |       height: 30vh;
74   |       font-size: 30px;
75   |   }
76   |   #output-decode-text {
77   |       width: 80vw;
78   |       height: 30vh;
79   |       font-size: 30px;
80   |   }
81   |   h1 {
82   |       font-size: 60px;
83   |   }
84   |   h2 {
85   |       font-size: 30px;
86   |   }
87   }
88
```

Рисунок 4 – css код

JS script.js >  textDecode

```
1  let alphabet = [  
2    "A",  
3    "B",  
4    "C",  
5    "D",  
6    "E",  
7    "F",  
8    "G",  
9    "H",  
10   "I",  
11   "J",  
12   "K",  
13   "L",  
14   "M",  
15   "N",  
16   "O",  
17   "P",  
18   "Q",  
19   "R",  
20   "S",  
21   "T",  
22   "U",  
23   "V",  
24   "W",  
25   "X",  
26   "Y",  
27   "Z",  
28   "a",  
29   "b",  
30   "c",  
31   "d",  
32   "e",
```

Рисунок 5 – js код

JS script.js > textDecode

```
1  let alphabet = [  
31  "d",  
32  "e",  
33  "f",  
34  "g",  
35  "h",  
36  "i",  
37  "j",  
38  "k",  
39  "l",  
40  "m",  
41  "n",  
42  "o",  
43  "p",  
44  "q",  
45  "r",  
46  "s",  
47  "t",  
48  "u",  
49  "v",  
50  "w",  
51  "x",  
52  "y",  
53  "z",  
54 ];  
55  
56 function textDecode(inputEncodeText) {  
57   let decodeResult = "";  
58  
59   // проходимся по введенному тексту  
60   for (let i = 0; i < inputEncodeText.length; i++) {
```

Рисунок 6 – js код


```

60 for (let i = 0; i < inputEncodeText.length; i++) {
61   let symbolOfInputText = inputEncodeText[i];
62   let checkAlphabetSymbol = false;
63
64   // Проверяем наличие символа в массиве с алфавитом
65   for (let j = 0; j < alphabet.length; j++) {
66     if (symbolOfInputText === alphabet[j]) {
67       checkAlphabetSymbol = true;
68       // проверка заглавных
69       let bigSymbol = j < 26;
70
71       // индекс после отступа
72       let newSymbolIndex;
73       if (bigSymbol) {
74         // заглавные
75         newSymbolIndex = (j - 13 + 26) % 26; // минусуем 13 для дешифровки, плюсуем 26 чтобы избежать минуса
76         // берем остаток от деления на 26 чтобы не получить индекс строчной буквы(26-51)
77       } else {
78         // строчные
79         newSymbolIndex = 26 + ((j - 26 - 13 + 26) % 26);
80       }
81
82       decodeResult += alphabet[newSymbolIndex];
83     }
84   }
85
86   // для не букв
87   if (!checkAlphabetSymbol) {
88     decodeResult += symbolOfInputText;
89   }

```

Рисунок 7 – js код

```

86   // для не букв
87   if (!checkAlphabetSymbol) {
88     decodeResult += symbolOfInputText;
89   }
90 }
91
92 return decodeResult;
93 }
94
95 let inputText = document.getElementById("input-encode-text");
96 let outputDecodeText = document.getElementById("output-decode-text");
97
98 inputText.addEventListener("change", function () {
99   //запуск функции при вводе текста в форму
100   // получtv вводимый текст для функции
101   let inputEncodeText = inputText.value;
102
103   // вывод расшифки
104   outputDecodeText.innerHTML = textDecode(inputEncodeText);
105 });
106

```

Рисунок 8 – js код

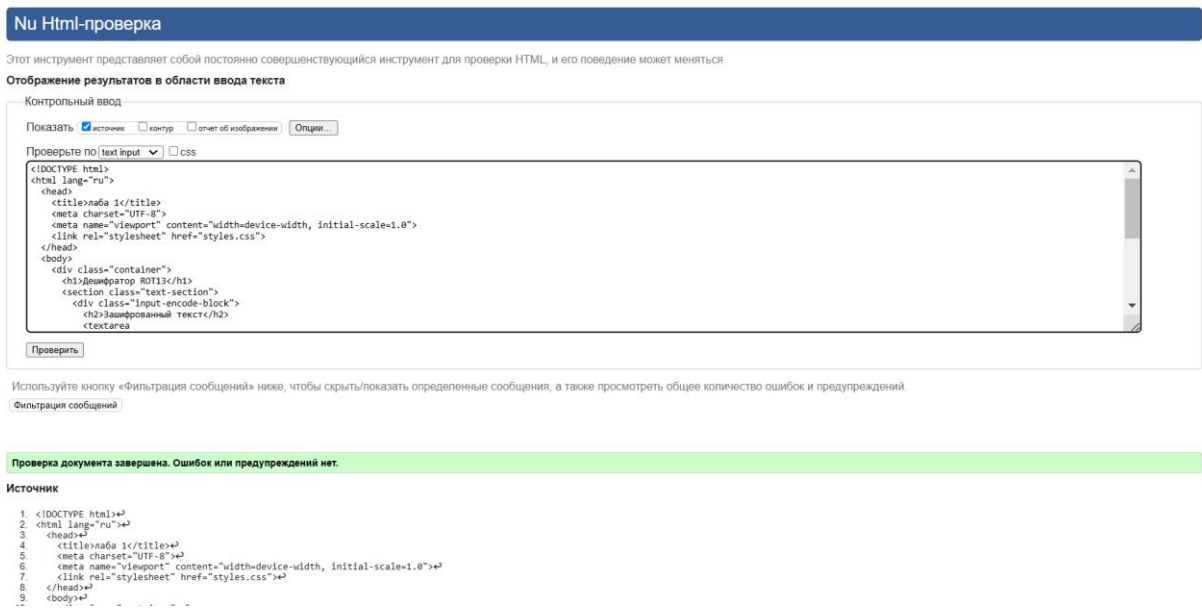


Рисунок 9 – проверка валидности

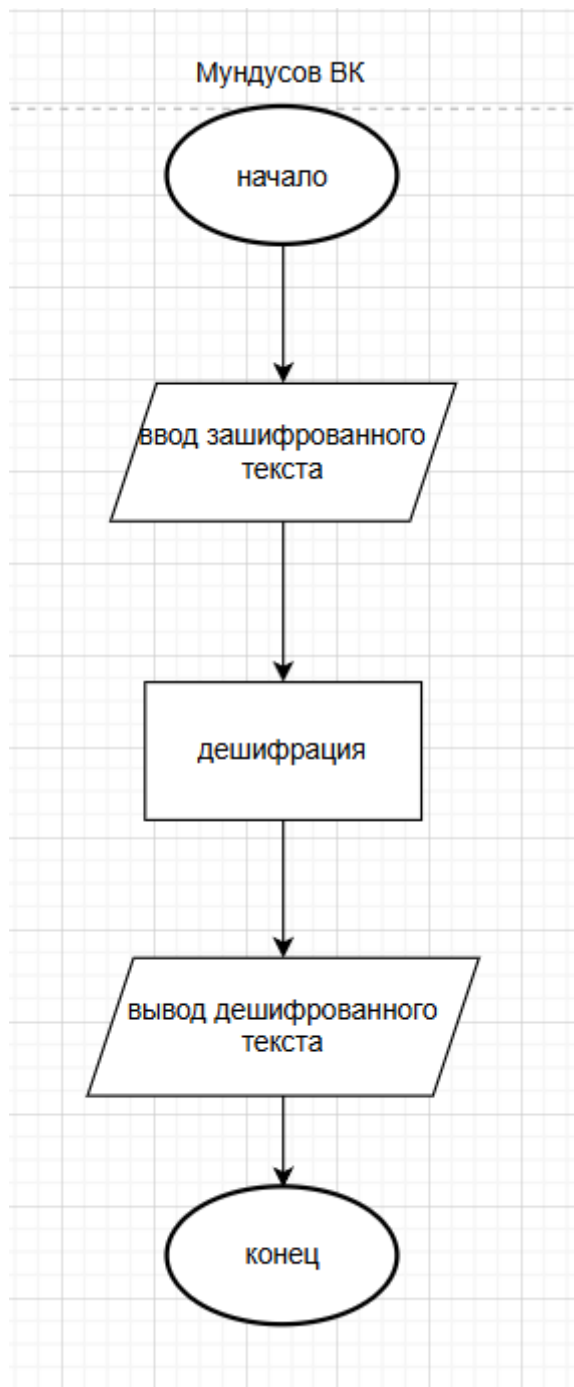


Рисунок 10 – блок-схема(нет)

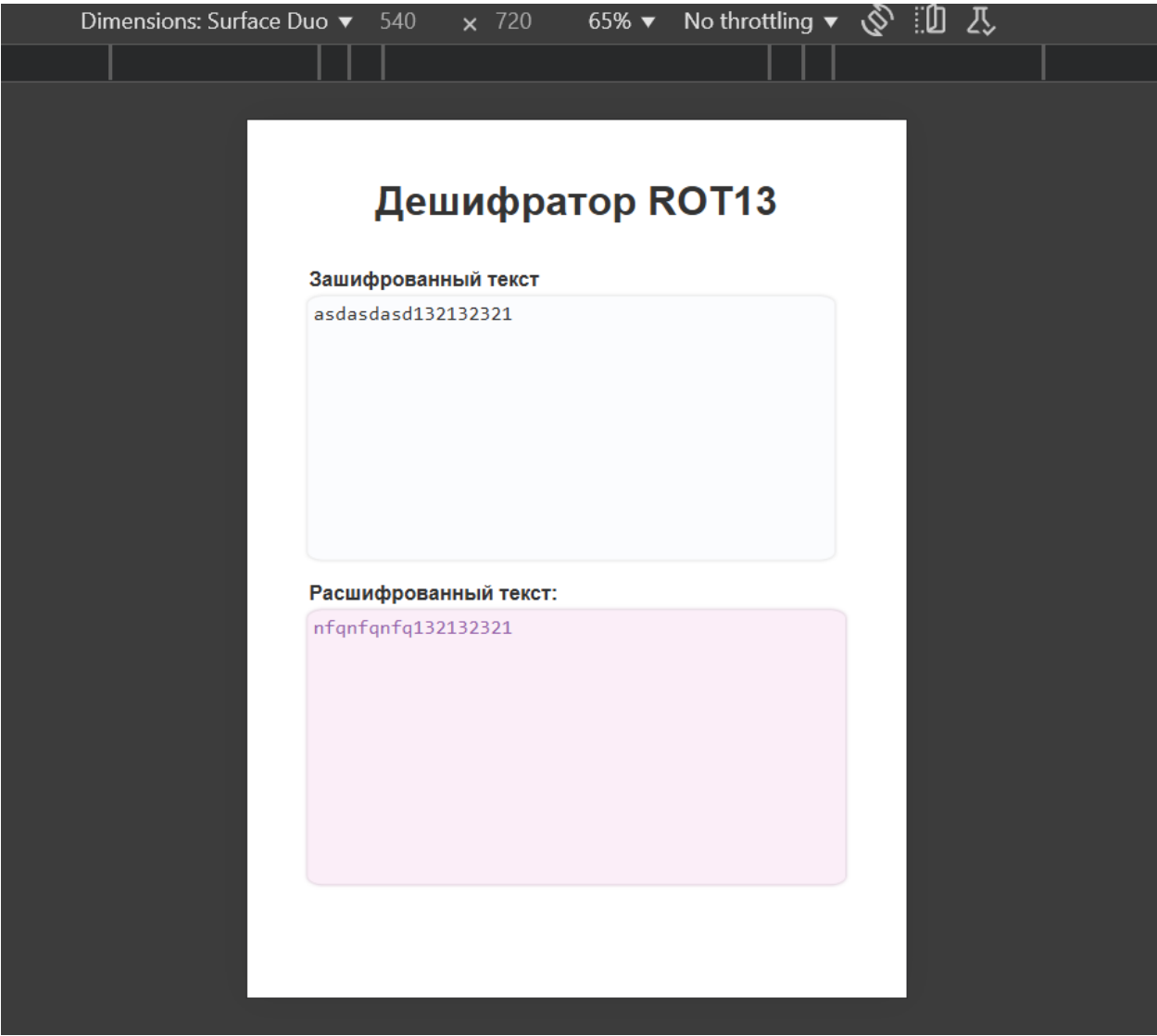


Рисунок 11 – адаптация

Тестирование

Ввод	Вывод
CEVIRG123!@#	PRIVET123!@#
grfg1212	test1212
312#!\$@#134#\$134	312#!\$@#134#\$134
Nmnmnmnm	Azazazaz
Uryyb, jbeyq!	Hello, world!

Дешифратор ROT13

Зашифрованный текст:

CEVIRG123!@#

Расшифрованный текст:

PRIVET123!@#

Рисунок 12 – тест 1

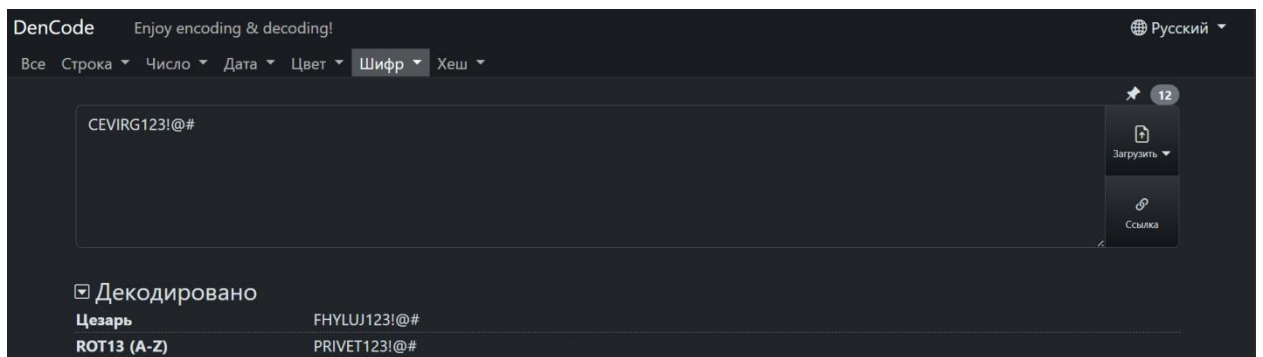


Рисунок 13 – тест 1

Дешифратор ROT13

Зашифрованный текст:

grfg1212

Расшифрованный текст:

test1212

Рисунок 14 – тест 2

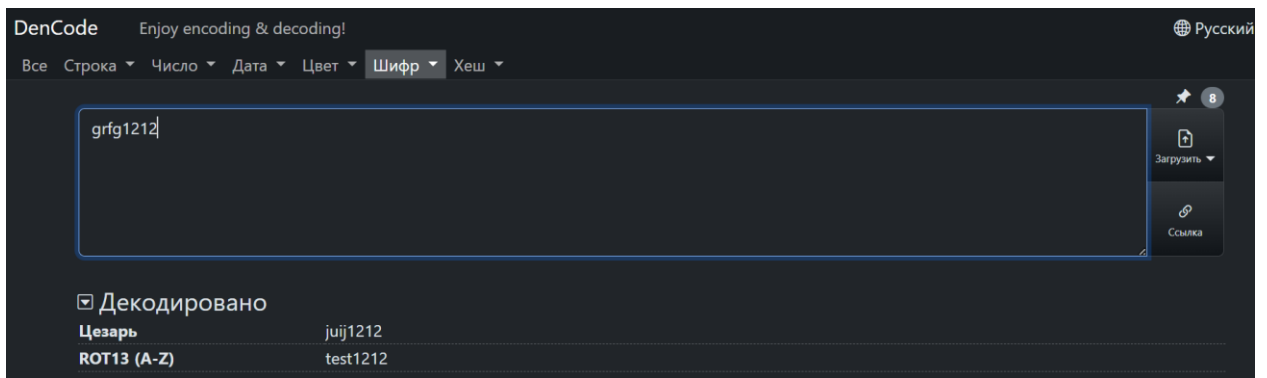


Рисунок 15 – тест 2

Дешифратор ROT13

Зашифрованный текст:

312#!\$@#134#\$134

Расшифрованный текст:

312#!\$@#134#\$134

Рисунок 16 – тест 3

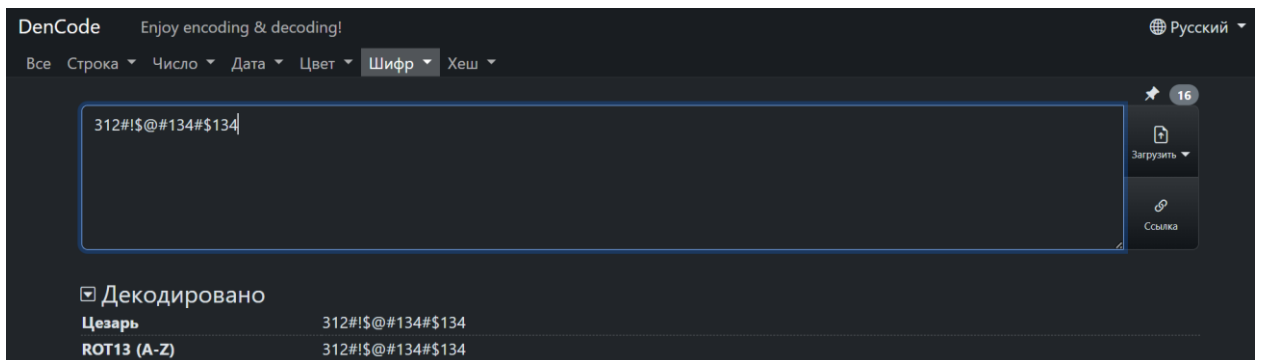


Рисунок 17 – тест 3

Дешифратор ROT13

Зашифрованный текст:

Nmmmmmm

Расшифрованный текст:

Azazazaz

Рисунок 18 – тест 4

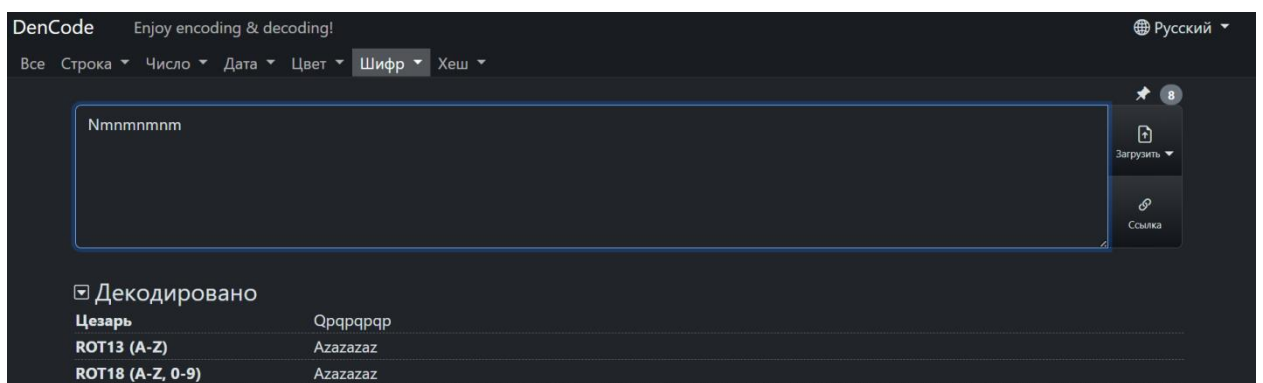


Рисунок 19 – тест 4

Дешифратор ROT13

Зашифрованный текст:

Uryyb, jbeyq!

Расшифрованный текст:

Hello, world!

Рисунок 20 – тест 5

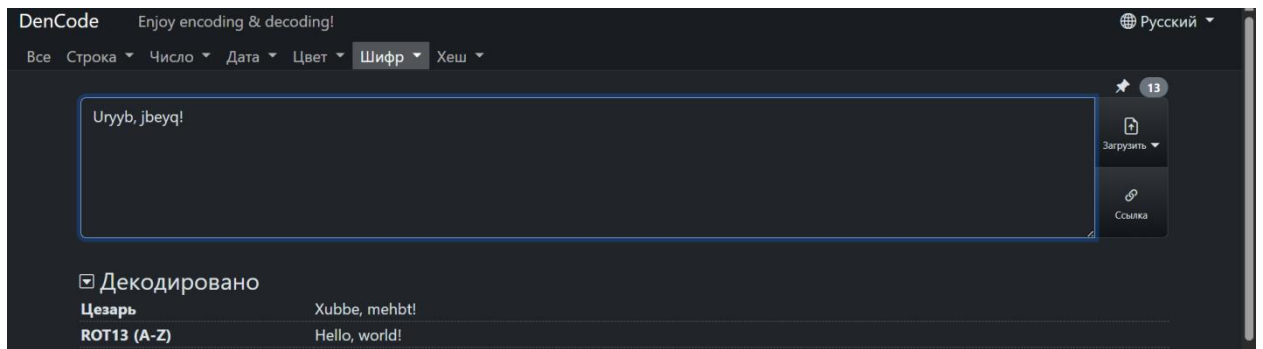


Рисунок 21 – тест 5

Вывод

Данное задание освежило мои знания о html, css и js. Я вспомнил как работать с массивами и математическими операциями js, верстать на html, стилизовать на css и работать с медиазапросами.

Гит:

https://github.com/MundusovVK/PRACT_UMBETOV_MUNDUSOV_VK_1ISP-22

Сайт:

https://mundusovvk.github.io/PRACT_UMBETOV_MUNDUSOV_VK_1ISP-22/