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

Отчет защищен с оценкой_____
Преподаватель_____ *С. В. Умбетов*
«_____»_____2025 г.

Отчёт по лабораторной работе № 1

«Реализация шифра ROT13 на JavaScript»

ЛР 09.03.01.14.002

Студент группы 1ИСП-21 А.А. Кайль
группа и.о., фамилия

Преподаватель *ассистент*, к. *т. н.* *С. В. Умбетов*
должность, ученая степень и.о., фамилия

БАРНАУЛ 2025

Лабораторная работа №1

Цели и задачи работы: необходимо создать html страницу и js код, страница должна быть валидной стандарту HTML5. В коде реализовать шифрование ROT13 для четного варианта и расшифровку для нечетного.

Задание принял: _____ Кайль А.А.

Подпись ФИО

Ход работы

ROT13 («поворот на 13 позиций») — это простой и широко известный шифр замены букв, который используется на онлайн-форумах и в текстовых файлах для сокрытия текста, например предупреждений о спойлерах или оскорбительном контенте. Этот метод заменяет букву на 13-ю по счёту в алфавите. A становится N, B становится O, C становится P и так далее.

ROT13 является обратным к самому себе, то есть для отмены ROT13 применяется тот же алгоритм, поэтому одно и то же действие можно использовать для кодирования и декодирования.

Алгоритм не обеспечивает криптографическую безопасность и часто приводится в качестве канонического примера слабого шифрования.

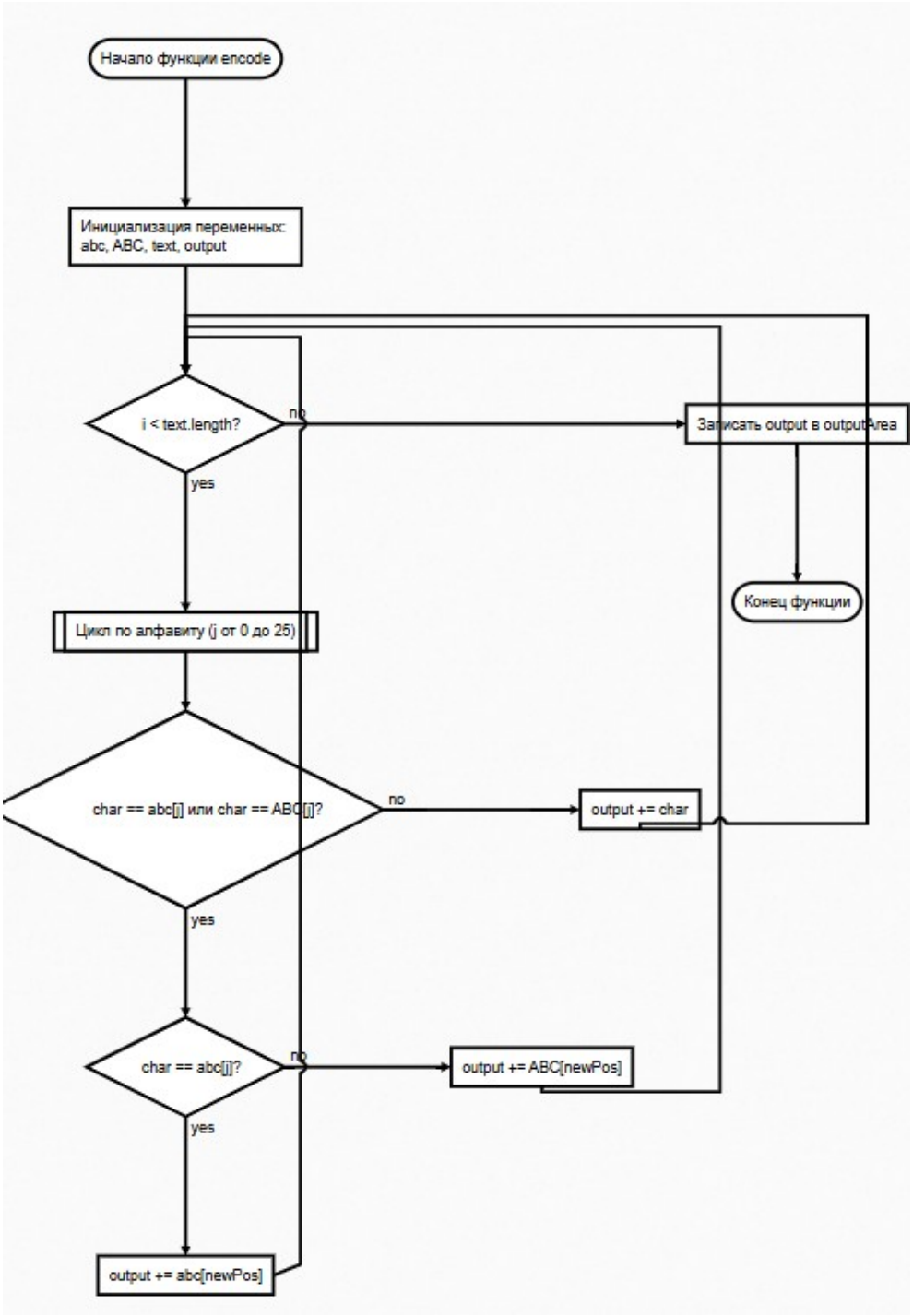

Рисунок 1 — Блок-схема

```
function encode() {
    let abc = 'abcdefghijklmnopqrstuvwxyz';
    let ABC = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
    let text = document.getElementById('inputArea').value;
    let output = '';
```

Сначала создадим основную функцию encode(), abc – строчные буквы, ABC – заглавные буквы, text —для шифрования из текстового поля с id="inputArea". и output чтобы записать сюда зашифрованный текст.

```
 for (let i = 0; i < text.length; i++) {
        let char = text[i];
        let found = false;
```

Это цикл, который проходит по каждому символу введённого текста. char – текущий символ (например, h в слове hello) found – флаг, который показывает была ли буква найдена в алфавите.

```
      for (let j = 0; j < abc.length; j++) {
           if (char = = = abc[j] || char = = = ABC[j]) {
```

for ((let j = 0; j < abc.length; j++) вложенный цикл проверяет есть ли текущий символ (char) в алфавите (abc или ABC).

if (char === abc[j] || char === ABC[j]) если символ совпадает с буквой в алфавите (строчной или заглавной) выполняется шифрование.

```
                let shift = 13;
                let newPos = (j + shift) % abc.length;
```

newPos = (j + shift) % abc.length вычисляет позицию буквы после сдвига. Если j + shift больше 26 позиция следует в начало.

Буква n (индекс 13): (13 + 13) % 26 = 0 — > a

```
 if (char === abc[j]) {
                    output += abc[newPos];
                } else {
                    output += ABC[newPos];
                }
                found = true;
                break;
```

Если строчный то берём новую букву из abc[newPos], если заглавный ABC[newPos]

Для цифр знаков препинаний и тд.

```
 if (!found) {
            output += char;
        }
```

Если символ не был найден в алфавите то выводится без изменений

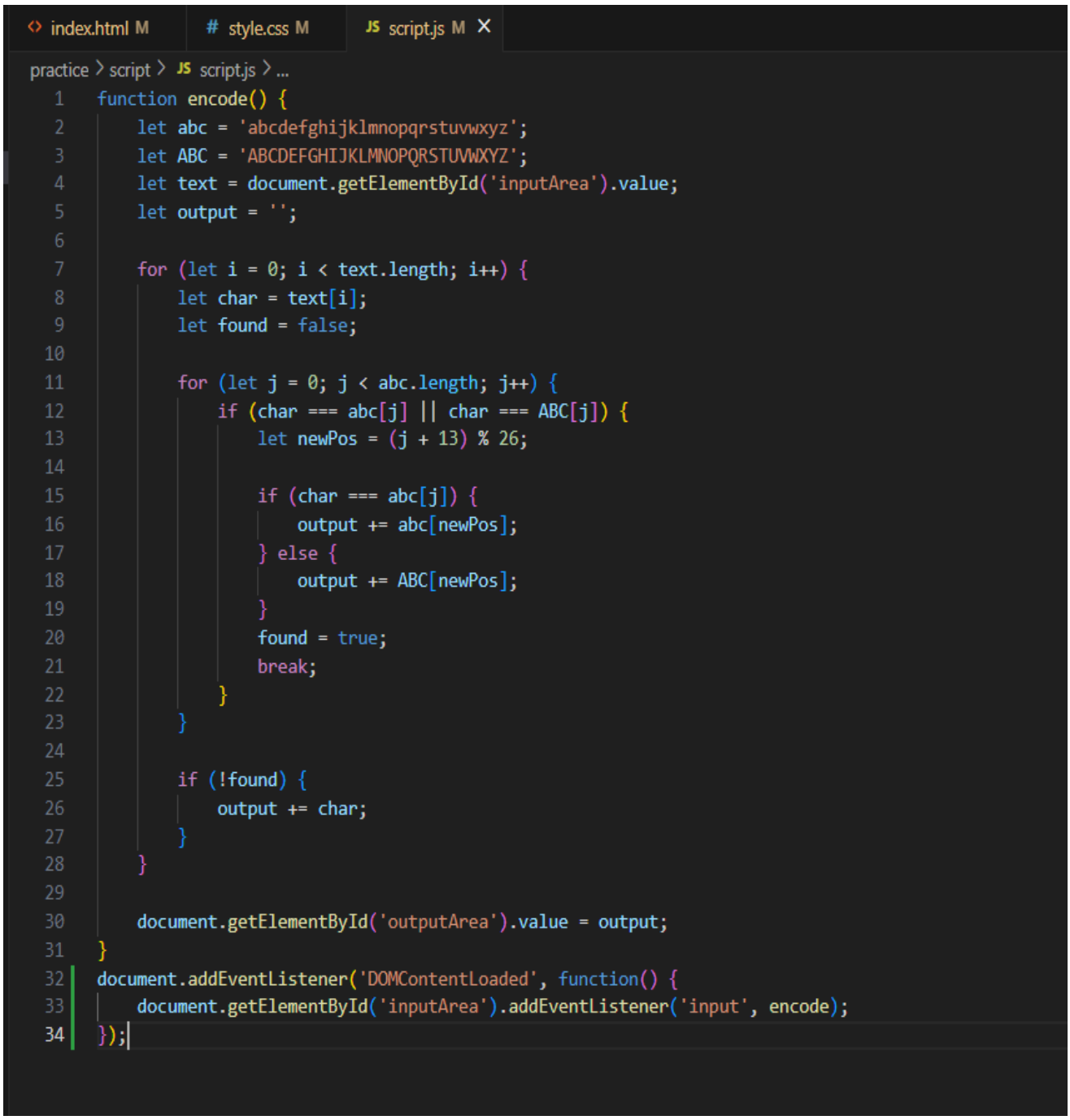

Рисунок 2 — Код JS

Тестирование

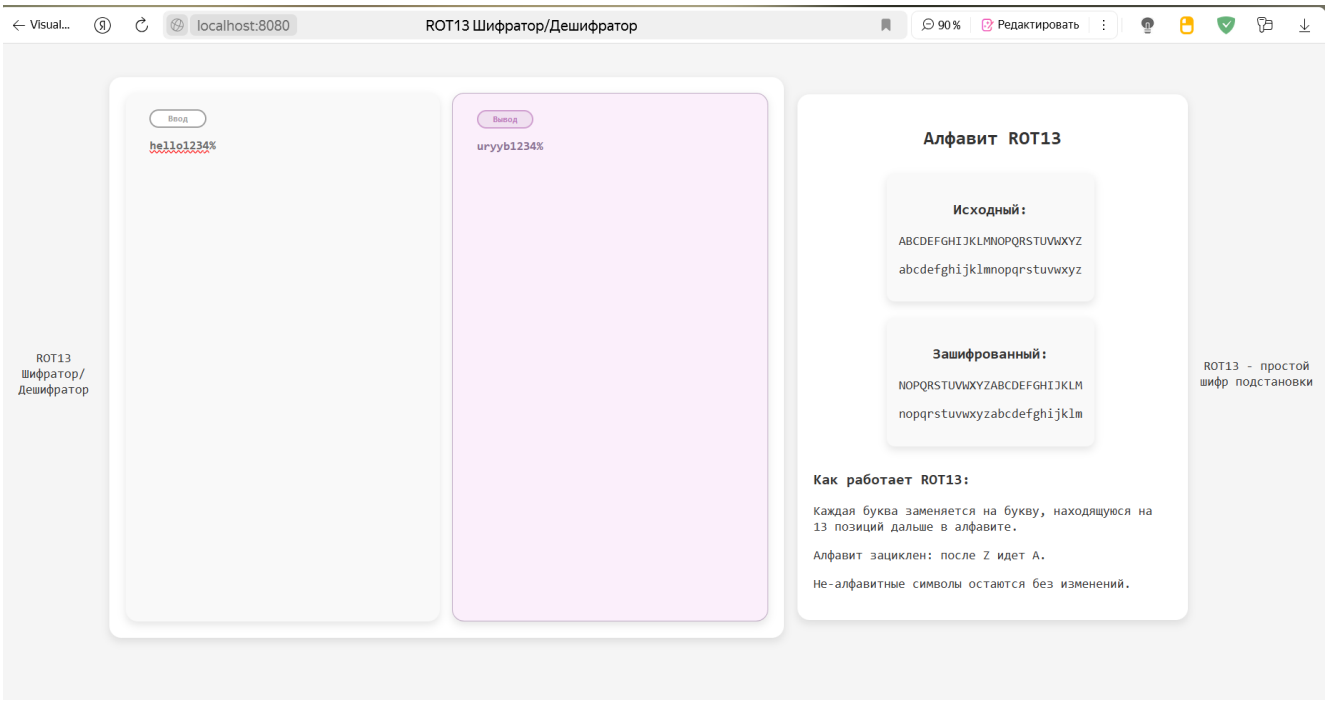

Рисунок 3 — Тестирование шифровка

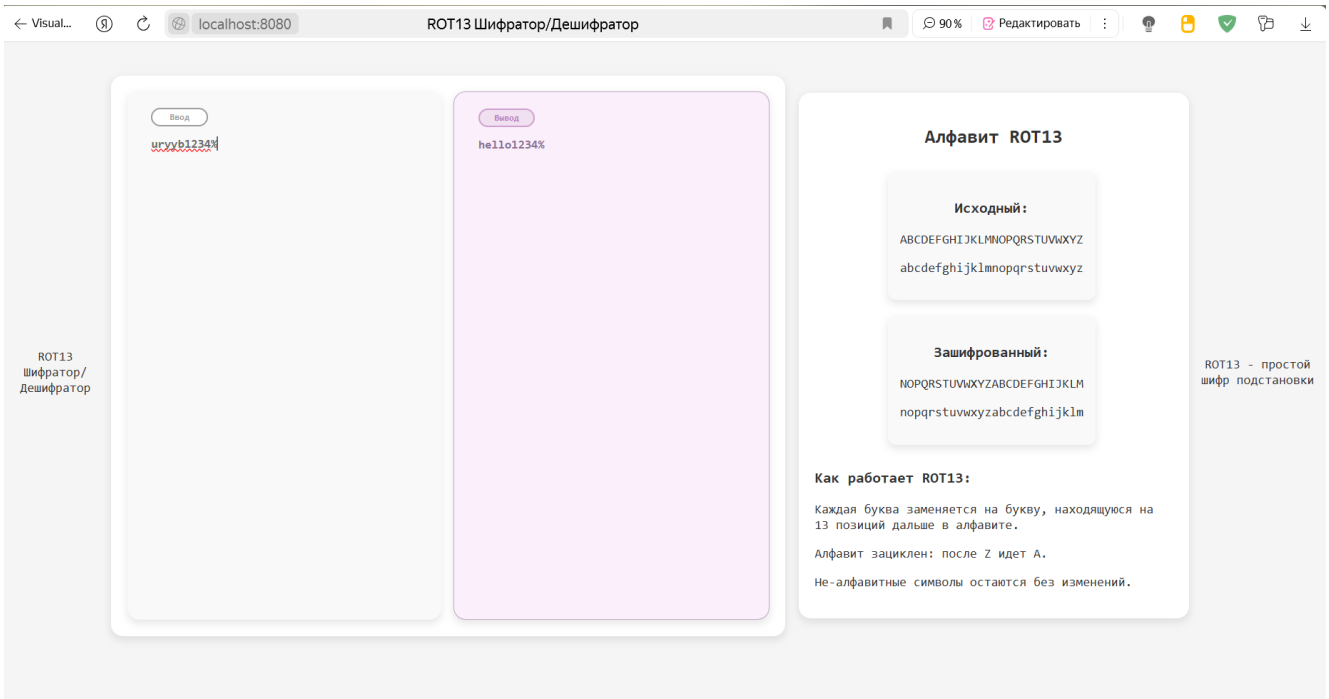

Рисунок 4 — Тестирование дешифровка

Проверка

Таблица 1: Проверка шифровки

Символ	**Тип**	**Позиция (j)**	**newPos = (j+13)%26**	**Измененный символ**	**Результат**
h	строчная	7	(7+13)%26 = 20	u (abc[20])	u
e	строчная	4	(4+13)%26 = 17	r (abc[17])	ur
l	строчная	11	(11+13)%26 = 24	y (abc[24])	ury
l	строчная	11	(11+13)%26 = 24	y (abc[24])	uryy
o	строчная	14	(14+13)%26 = 1	b (abc[1])	uryyb
1	цифра	-	-	1 (без изменений)	uryyb1
%	символ	-	-	% (без изменений)	uryyb1234%

Таблица 2: Проверка дешифровки

Символ	**Тип**	**Позиция (j)**	**newPos = (j+13)%26**	**Изначальный символ**	**Результат**
u	строчная	20	(20+13)%26 = 7	h (abc[7])	h
r	строчная	17	(17+13)%26 = 4	e (abc[4])	he
y	строчная	24	(24+13)%26 = 11	l (abc[11])	hel
y	строчная	24	(24+13)%26 = 11	l (abc[11])	hell
b	строчная	1	(1+13)%26 = 14	o (abc[14])	hello
1	цифра	-	-	1 (без изменений)	hello1
%	символ	-	-	% (без изменений)	hello1234%

Код

HTML

index.html M | style.css M

practice > index.html > html > body

```html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>ROT13 Шифратор/Дешифратор</title>
    <link rel="stylesheet" href="./style/style.css">
    <script src="./script/script.js"></script>
</head>
<body>
    <header>
        <p>ROT13 Шифратор/Дешифратор</p>
    </header>

    <main>
        <div class="centralBlock">
            <div class="inputPanel">
                <p>Ввод</p>
                <textarea id="inputArea" placeholder="Введите текст для шифрования/дешифрования..."></textarea>
            </div>

            <div class="outputPanel">
                <p>Вывод</p>
                <textarea id="outputArea" readonly></textarea>
            </div>
        </div>
    </main>
    <div class="alphabet-info">
        <h2>Алфавит ROT13</h2>
        <div class="alphabet-pair">
            <div class="alphabet-original">
                <h3>Исходный:</h3>
                <p>ABCDEFGHIJKLMNOPQRSTUVWXYZ</p>
                <p>abcdefghijklmnopqrstuvwxyz</p>
            </div>
            <div class="alphabet-rot13">
                <h3>Зашифрованный:</h3>
                <p>NOPQRSTUVWXYZABCDEFGHIJKLM</p>
                <p>nopqrstuvwxyzabcdefghijklm</p>
            </div>
        </div>
        <div class="operation-info">
            <h3>Как работает ROT13:</h3>
            <p>Каждая буква заменяется на букву, находящуюся на 13 позиций дальше в алфавите.</p>
            <p>Алфавит зациклен: после Z идет A.</p>
            <p>Не-алфавитные символы остаются без изменений.</p>
        </div>
    </div>

    <footer>
        <p>ROT13 - простой шифр подстановки</p>
    </footer>


</body>
</html>
```

Рисунок 5 — Код html

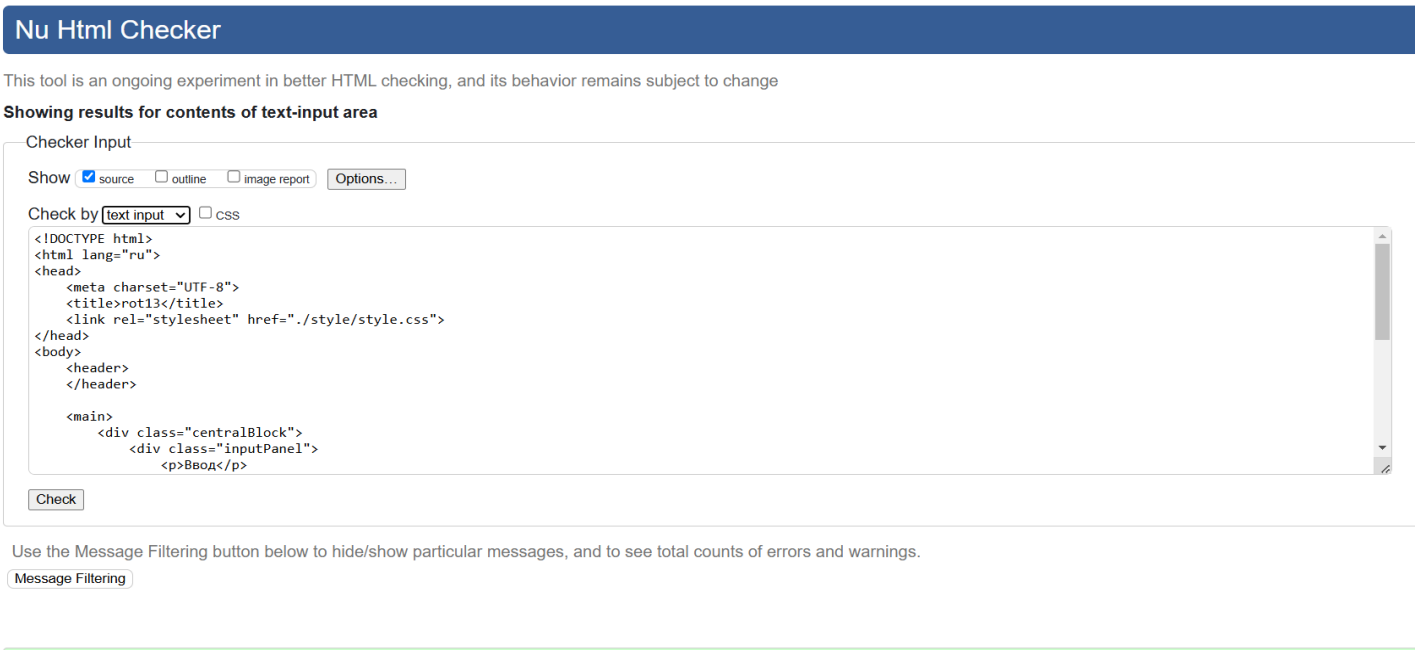

Рисунок 6 — Проверка html на валидность

CSS

Рисунок 7 — Стили css

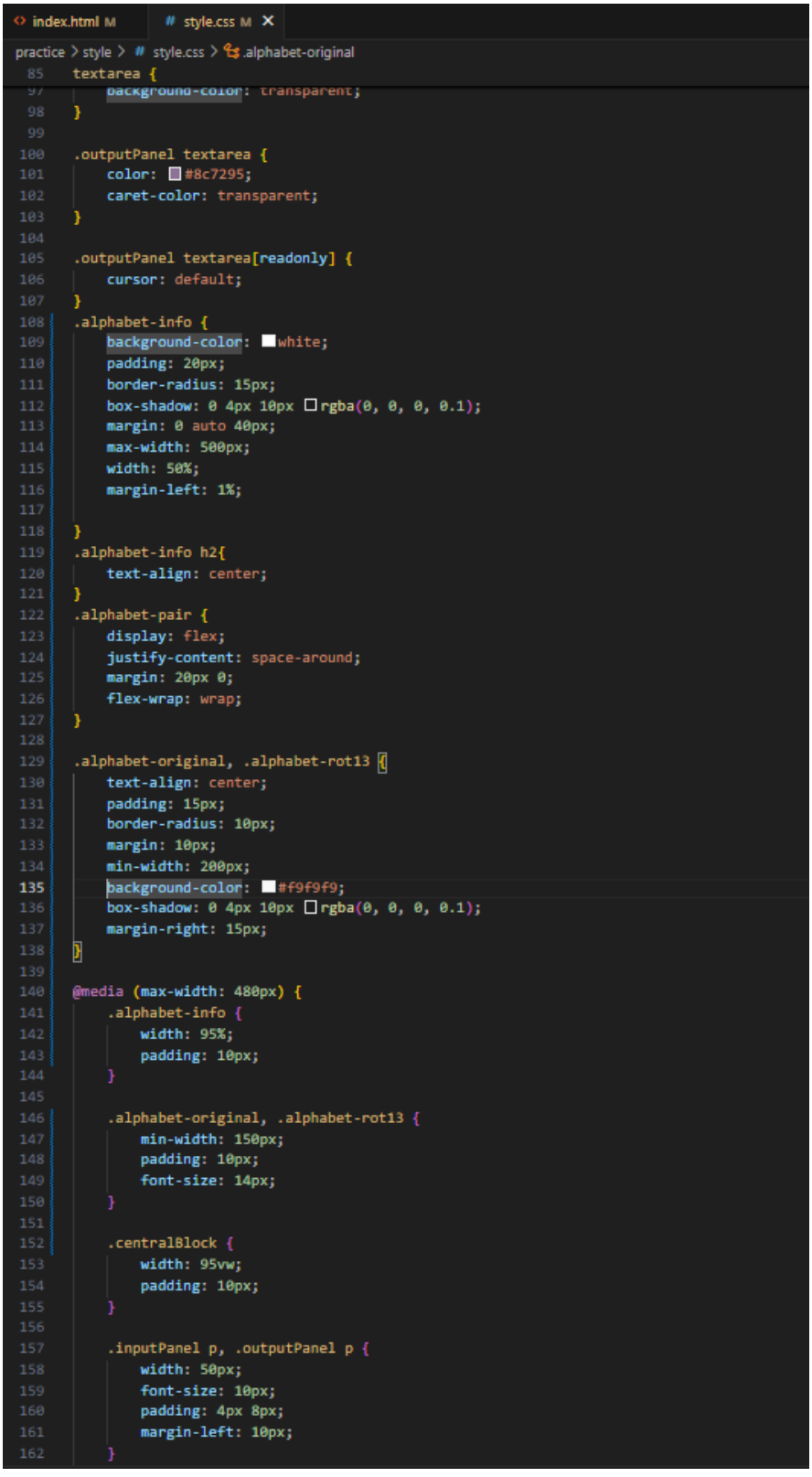

Рисунок 8 — Стили css @media

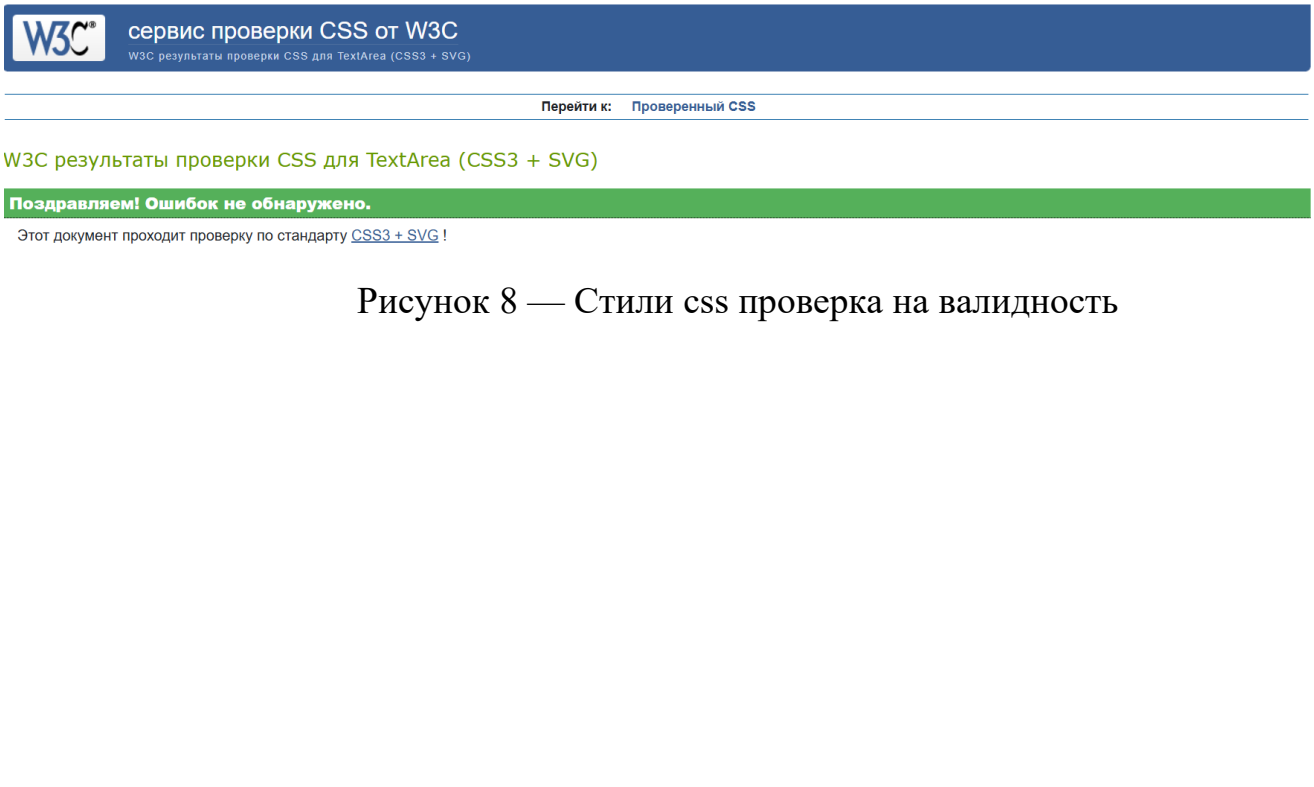

Рисунок 8 — Стили css проверка на валидность

Вывод

В ходе данной лабораторной работы была создана функция, которая реализует алгоритм rot13 на JavaScript, который шифрует и дешифрует введенные данные. Проблемы: обработка регистра букв(раздельные алфавиты строчные и заглавные), использование % 26 для обратного сдвига.

https://github.com/him1k0ta/practice.git