

# **Отчёт по лабораторной работе 1**

**МОЗИИБ**

Папикян Гагик Тигранович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
3.1	Принцип работы симметричных алгоритмов . . . . .	7
3.2	Шифр Цезаря . . . . .	7
3.3	Шифр Атбáш . . . . .	8
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
<b>5</b>	<b>Выводы</b>	<b>12</b>

# List of Figures

4.1	Выполнение лабораторной работы . . . . .	11
-----	--	----

## List of Tables

# 1 Цель работы

Познакомиться с простейшими алгоритмами шифрования данных, посредством реализации шифра Цезаря и Атбаш

## 2 Задание

- 1) Реализовать шифр Цезаря с ключом  $K$
- 2) Реализовать шифр Атбаш

## 3 Теоретическое введение

### 3.1 Принцип работы симметричных алгоритмов

В целом симметричным считается любой шифр, использующий один и тот же секретный ключ для шифрования и расшифровки.

Например, если алгоритм предполагает замену букв числами, то и у отправителя сообщения, и у его получателя должна быть одна и та же таблица соответствия букв и чисел: первый с ее помощью шифрует сообщения, а второй — расшифровывает.

Однако такие простейшие шифры легко взломать — например, зная частотность разных букв в языке, можно соотносить самые часто встречающиеся буквы с самыми многочисленными числами или символами в коде, пока не удастся получить осмысленные слова. С использованием компьютерных технологий такая задача стала занимать настолько мало времени, что использование подобных алгоритмов утратило всякий смысл.

### 3.2 Шифр Цезаря

Шифр Цезаря, также известный как шифр сдвига, код Цезаря или сдвиг Цезаря — один из самых простых и наиболее широко известных методов шифрования.

Шифр Цезаря — это вид шифра подстановки, в котором каждый символ в открытом тексте заменяется символом, находящимся на некотором постоянном числе позиций левее или правее него в алфавите. Например,

в шифре со сдвигом вправо на 3, А была бы заменена на Г, Б станет Д, и так далее.

Шифр назван в честь римского полководца Гая Юлия Цезаря, использовавшего его для секретной переписки со своими генералами.

Шаг шифрования, выполняемый шифром Цезаря, часто включается как часть более сложных схем, таких как шифр Виженера, и всё ещё имеет современное приложение в системе ROT13. Как и все моноалфавитные шифры, шифр Цезаря легко взламывается и не имеет почти никакого применения на практике.

### 3.3 Шифр Атбáш

Атбáш — простой шифр подстановки для алфавитного письма. Правило шифрования состоит в замене  $i$ -й буквы алфавита буквой с номером  $n-i+1$ , где  $n$  — число букв в алфавите. Ниже даны примеры для английского, русского и еврейского алфавитов:



## 4 Выполнение лабораторной работы

Был написан следующий скрипт на javascript

```
//Caesar Cipher
//UTF
//[a-z] 97-122
//[A-Z] 65-90
const alphabet = "! abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"

function CaesarEncode( msg, k = 1){
    let res = ''
    for(let char of msg){
        let code = alphabet.indexOf(char)
        res += alphabet[ (code+k)%alphabet.length ]
    }
    return res
}

function CaesarDecode( msg, k = 1){
    let res = ''
    for(let char of msg){
        let code = [...alphabet].reverse().indexOf(char)
        res += [...alphabet].reverse()[ (code+k)%alphabet.length ]
    }
}
```

```

    return res
}

//Atbash Cipher
function AtbashEncodeDecode( msg ){
    let res = ''
    for(let char of msg){
        let code = alphabet.indexOf(char)
        res += [...alphabet].reverse()[ code ]
    }
    return res
}

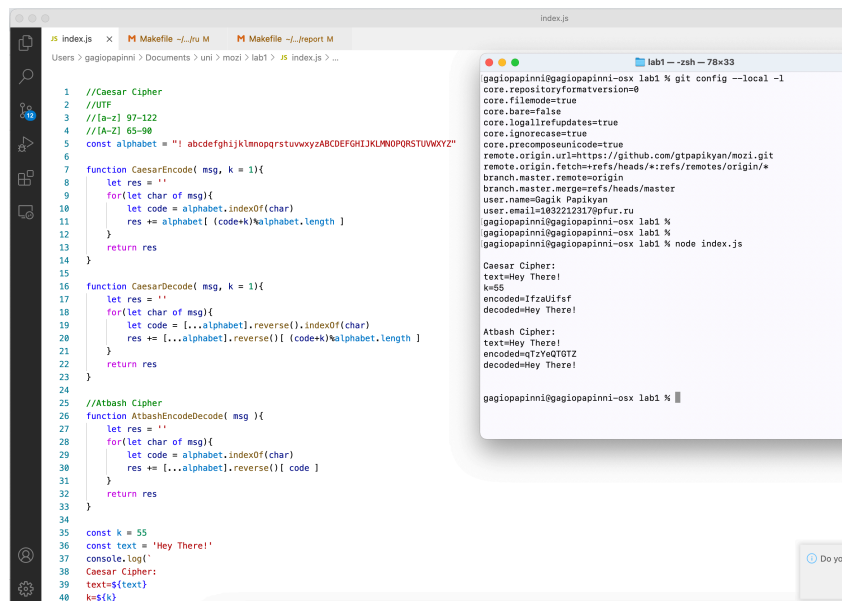
const k = 55
const text = 'Hey There!'
console.log(`
Caesar Cipher:
text=${text}
k=${k}
encoded=${CaesarEncode( text, k )}
decoded=${ CaesarDecode( CaesarEncode( text, k ), k)}

Atbash Cipher:
text=${text}
encoded=${AtbashEncodeDecode( text )}
decoded=${ AtbashEncodeDecode( AtbashEncodeDecode( text ))}

`)

```

Результат исполнения скрипта приведен на рисунке 1 (рис. 4.1)



```
1 //Caesar Cipher
2 //UTF
3 //[-z] 97-122
4 //[-z] 65-90
5 const alphabet = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
6
7 function CaesarEncode( msg, k = 1){
8   let res = ''
9   for(let char of msg){
10     let code = alphabet.indexOf(char)
11     res += alphabet[(code+k)%alphabet.length]
12   }
13   return res
14 }
15
16 function CaesarDecode( msg, k = 1){
17   let res = ''
18   for(let char of msg){
19     let code = [...alphabet].reverse().indexOf(char)
20     res += [...alphabet].reverse()[ (code+k)%alphabet.length ]
21   }
22   return res
23 }
24
25 //Atbash Cipher
26 function AtbashEncodeDecode( msg ){
27   let res = ''
28   for(let char of msg){
29     let code = alphabet.indexOf(char)
30     res += [...alphabet].reverse()[ code ]
31   }
32   return res
33 }
34
35 const k = 55
36 const text = "Hey There!"
37 console.log('
38 Caesar Cipher:
39 text=${text}
40 k=${k}
```

```
gagiopapinni@gagiopapinni-osx lab1 % git config --local -l
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
core.ignorecase=true
core.precomposeunicode=true
remote.origin.url=https://github.com/gtpapikyan/mozi.git
remote.origin.fetch=refs/heads/*:refs/remotes/origin/*
branch.master.remote=origin
branch.master.merge=refs/heads/master
user.name=Gagik Papikyan
user.email=1032212317@pfur.ru
gagiopapinni@gagiopapinni-osx lab1 %
gagiopapinni@gagiopapinni-osx lab1 % node index.js

Caesar Cipher:
text=Hey There!
k=55
encoded=IfzaUlfzf
decoded=Hey There!

Atbash Cipher:
text=Hey There!
encoded=q7zYq7GTGTZ
decoded=Hey There!

gagiopapinni@gagiopapinni-osx lab1 %
```

Figure 4.1: Выполнение лабораторной работы

## 5 Выводы

Был реализован Шифр Цезаря с произвольным ключом и Шифр Атбаша  
Был использован фиксированный алфавит, состоящий из символов “!  
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ”

На рисунке 4.1 в окне терминала было показано, как текст “Hey There!”  
зашифровывается и расшифровывается сначала алгоритмом Цезаря, потом  
Атбаш