

# **Отчёт по лабораторной работе №1**

**Шифр простой замены**

Исмит Шаманта НФИМд-22

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Теоретические сведения</b>	<b>5</b>
2.1	Шифр Цезаря . . . . .	5
2.2	Шифр Атбаш . . . . .	6
<b>3</b>	<b>Выполнение работы</b>	<b>7</b>
3.1	Реализация шифра Цезаря на языке Python . . . . .	7
3.2	Реализация шифра Атбаш на языке Python . . . . .	9
3.3	Контрольный пример . . . . .	10
<b>4</b>	<b>Выводы</b>	<b>11</b>
	<b>Список литературы</b>	<b>12</b>

# List of Figures

3.1 Работа алгоритмов . . . . . 10

# 1 Цель работы

Изучение алгоритмов шифрования Цезаря и Атбаш

## 2 Теоретические сведения

### 2.1 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 3 А была бы заменена на Г, Б станет Д, и так далее.

Шифр назван в честь римского императора 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 все ещё имеет современное приложение в системе ROT13. Как и все моноалфавитные шифры, шифр Цезаря легко взламывается и не имеет практически никакого применения на практике.

Если сопоставить каждому символу алфавита его порядковый номер (нумеруя с 0), то шифрование и дешифрование можно выразить формулами модульной арифметики:

$$y = (x + k) \bmod n$$

$$x = (y - k + n) \bmod n$$

где  $x$  — символ открытого текста,  $y$  — символ шифрованного текста  $n$  — мощность алфавита  $k$  — ключ.

С точки зрения математики шифр Цезаря является частным случаем аффинного шифра.

## 2.2 Шифр Атбаш

Атбаш — простой шифр подстановки, изначально придуманный для иврита. Правило шифрования состоит в замене  $i$ -й буквы алфавита буквой с номером  $n - i + 1$ , где  $n$  — число букв в алфавите.

## 3 Выполнение работы

### 3.1 Реализация шифра Цезаря на языке Python

Блок шифрования

```
# функция шифрования по алгоритму цезаря
def tsesar():
    # для работы необходим алфавит, его мы и объявили
    # алфавит можно расширить и до русских букв
    letters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ'
    # это шаг в шифровке. Его можно и даже нужно менять.
    # Типо на сколько шагов сделать ход по алфавиту.
    step = 5
    # строка для ввода текста. Вводить надо исключительно те символы
    # что есть в алфавите, который мы сверху написали,
    # иначе будут ошибки.
    text = input("Цезарь - шифрование :)")
    # переменная для записи результата
    result = ''
    # сам процесс шифрования начинается уже тут
    for i in text:
        ind = letters.find(i)
        # Вычисляем места символов в списке
        newind = ind + step
```

```

        # Сдвигаем символы на указанный в переменной step шаг
        if i in letters:
            result += letters[newind]
            # Задаем значения в итог
        else:
            result += i
    print(result)

```

### Блок дешифровки

```

# процесс дешифровки уже должен быть ясен
# вместо добавления шага, надо, наоборот же, вычитать,
# чтоб из зашифр сообщения получить открытый текст
# по сути код такой же, лишь маленькое отличие: вместо + -
def tsesar_deshifr():
    letters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ'
    smeshenie = 5
    text = input("Цезарь - дешифровка")
    result = ''

    for i in text:
        ind = letters.find(i)
        newind = ind - smeshenie
        if i in letters:
            result += letters[newind]
        else:
            result += i
    print(result)

```



## 3.2 Реализация шифра Атбаш на языке Python

### Блок шифрования

```
# шифр атбаша заключается тупо в том, что меняются буквы
# из обычного алфавита на буквы из алфавита-наоборот
# вместо А идет Z и тп
def atbash():
    # задаем алфавит
    letters = [chr(x) for x in range(65, 91)]
    # алфавит-наоборот
    letters_r = [x for x in letters]
    letters_r.reverse()

    text = input("Атбаш - шифрование")
    result = ""
    # тут для перебираются буквы из исходного текста
    for i in text:
        # перебираются индексы и значения из letters
        for j, l in enumerate(letters):
            if i == l: # если буквы i и l равны, то
                result += letters_r[j]
        # ставим в результат букву из реверсированного списка с индексом j
    print(result)
```

### Блок дешифровки

```
# функция дешифровки практически такая же
# тут просто местами мы поменяли списки чтоб наоборот дешифровать сообщения
def atbash_desh():
    letters = [chr(x) for x in range(65, 91)]
```

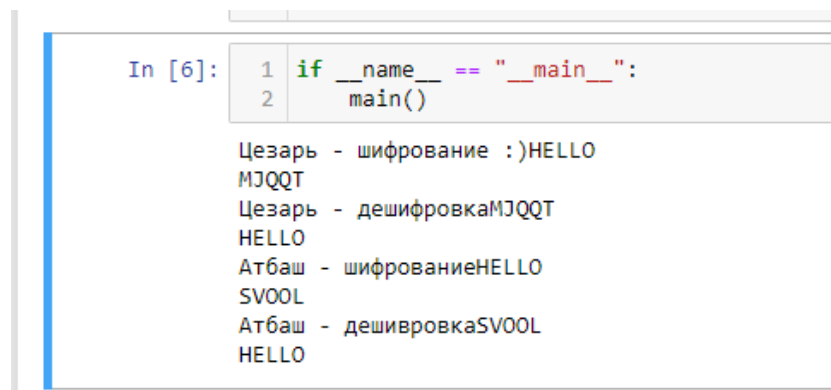
```

letters_r = [x for x in letters]
letters_r.reverse()

text = input("Атбаш - дешифровка")
result = ""
for i in text:
    for j, l in enumerate(letters_r):
        if i == l:
            result += letters[j]
print(result)

```

### 3.3 Контрольный пример



```

In [6]: 1 if __name__ == "__main__":
        2     main()

Цезарь - шифрование :)HELLO
MJQQT
Цезарь - дешифровкаMJQQT
HELLO
Атбаш - шифрованиеHELLO
SV00L
Атбаш - дешифровкаSV00L
HELLO

```

Figure 3.1: Работа алгоритмов

## **4 Выводы**

Изучили алгоритмы шифрования Цезаря и Атбаш.

# Список литературы

1. Шифр Цезаря
2. Шифр Атбаш