

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

|  |  |
| --- | --- |
| Факультет | Информатика и вычислительная техника |
|  | (наименование факультета) |
| Кафедра | Кибербезопасность информационных систем |
|  | (наименование кафедры) |

**Лабораторная работа №1**

**по дисциплине «Дискретная математика»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент |  | |  |  |
|  | (подпись, дата) | |  | (ФИО, группа) |
| Направление подготовки | | 10.05.01 Компьютерная безопасность | | |
| Профиль | Информатика и вычислительная техника | | | |
| Преподаватель |  | |  | ст. пр. Типаева Э. Р. |
|  | (подпись, дата) | |  |  |

Ростов-на-Дону

2023

**Листинг**

# Функция нахождения НОД

def NOD():

    nodinston=[]

    for i in range(len(fin\_mnoj)):

        in\_fin\_mnoj=fin\_mnoj[i]

        for o in range(2,50):

            if o in in\_fin\_mnoj:

                nodinston.append(o)

    itog=1

    for p in range(2,50):

        if len(fin\_mnoj)==1 or len(fin\_mnoj)==0:

            continue

        else:

            if len(fin\_mnoj)==nodinston.count(p):

                itog=p

    print('Ваш НОД:')

    return itog

# Функция нахождения НОК

def NOK1():

    nokinston=[]

    for i in range(len(fin\_mnoj)):

        in\_fin\_mnoj=fin\_mnoj[i]

        for d in range(len(in\_fin\_mnoj)):

            if in\_fin\_mnoj[d] not in nokinston:

                nokinston.append(in\_fin\_mnoj[d])

            elif nokinston.count(in\_fin\_mnoj[d]) < list(in\_fin\_mnoj).count(in\_fin\_mnoj[d]):

                nokinston.append(in\_fin\_mnoj[d])

    itog=1

    for i in range(len(nokinston)):

        itog\*=nokinston[i]

    print('Ваш НОК:')

    return itog

# Функция разложения на множители

def Packlad(n):

    mnoj = []

    delit = 2

    while n >= delit:

        if n % delit == 0:

            mnoj.append(delit)

            n = n / delit

        else:

            delit = delit + 1

    return mnoj

# Функции для ввода и проверка на 0

num1=[]

num=input('Введите числа через пробел, нельзя вводить ноль: ').split()

for i in range(len(num)):

    if num[i]!='0':

        num1.append(num[i])

    else:

        print('Ваш 0 был удален')

print('Ваши числа: ',num1)

fin\_mnoj=[]

for i in range(len(num1)):

    try:

        numb=abs(int(num[i]))

    except:

        fin\_mnoj.append([float(num[i])])

    else:

        fin\_mnoj.append(Packlad(numb))

print('Множители все: ', fin\_mnoj)

print(NOD())

print(NOK1())

Изображение выглядит как символ, логотип, зарисовка, эмблема

Автоматически созданное описание

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

|  |  |
| --- | --- |
| Факультет | Информатика и вычислительная техника |
|  | (наименование факультета) |
| Кафедра | Кибербезопасность информационных систем |
|  | (наименование кафедры) |

**Лабораторная работа №2**

**по дисциплине «Дискретная математика»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент |  | |  |  |
|  | (подпись, дата) | |  | (ФИО, группа) |
| Направление подготовки | | 10.05.01 Компьютерная безопасность | | |
| Профиль | Информатика и вычислительная техника | | | |
| Преподаватель |  | |  | ст. пр. Типаева Э. Р. |
|  | (подпись, дата) | |  |  |

Ростов-на-Дону

2023

**Листинг**

from prettytable import PrettyTable

# Функция построения таблицы истинности

def Table(a):

    table=PrettyTable()

    table.field\_names=['x','y','z','f']

    table.add\_row(['0','0','0',a[0]])

    table.add\_row(['0','0','1',a[1]])

    table.add\_row(['0','1','0',a[2]])

    table.add\_row(['0','1','1',a[3]])

    table.add\_row(['1','0','0',a[4]])

    table.add\_row(['1','0','1',a[5]])

    table.add\_row(['1','1','0',a[6]])

    table.add\_row(['1','1','1',a[7]])

    print('Таблица истинности:')

    return table

# Функция метода треугольником

def Triangle(a):

    print(' '.join(a).center(20))

    post=[a[0]]

    bylfun1=a

    for l in range(len(a)-1):

        bylfunft1=bylfun1

        lenbfft=len(bylfunft1)-1

        bylfunft=''

        for i in range(lenbfft):

            bylfunft+=logic(bylfunft1[i],bylfunft1[i+1])

        post.append(bylfunft[0])

        print(' '.join(bylfunft).center(20))

        bylfun1=bylfunft

    print(Jegalkin(post))

    return post

# Функция преобразования Жегалкина

def Jegalkin(a):

    table\_ist={0:'1',1:'z',2:'y',3:'yz',4:'x',5:'xz',6:'xy',7:'xyz'}

    prejeg=[]

    for i in range(len(a)):

        if a[i]=='1':

            prejeg.append(table\_ist[i])

    prejeg='⨁ '.join(prejeg)

    print('')

    print('Полином Жегалкина:')

    return prejeg

# Функция логики(Исключающее или)

def logic(a,b):

        if (a=='0' and b=='0') or (a=='1' and b=='1'):

            return '0'

        else:

            return '1'

# Функция метода БПФ

def Foure(a):

    print(' '.join(a).center(17))

    bylfun1=a

    bylfunft=''

    for i in range(0,len(a),2):

        bylfunft+=bylfun1[i]

        bylfunft+=logic(bylfun1[i],bylfun1[i+1])

    print(' '.join(bylfunft).center(17))

    bylfun1=bylfunft[0]+bylfunft[1]

    for i in range(0,2):

        bylfun1+=logic(bylfunft[i],bylfunft[i+2])

    bylfun1+=bylfunft[4]+bylfunft[5]

    for i in range(4,6):

        bylfun1+=logic(bylfunft[i],bylfunft[i+2])

    print(' '.join(bylfun1).center(17))

    print('-'\*17)

    bylfunft=bylfun1[0:4]

    for i in range(4):

        bylfunft+=logic(bylfun1[i],bylfun1[i+4])

    print(' '.join(bylfunft).center(17))

    print(Jegalkin(bylfunft))

    return bylfunft

#Функция ввода и проверки

bylfun=input('Введите б.ф. для 3-ех переменных (слитно): ')

u=0

while u!=1:

    if len(bylfun)==8 and bylfun.isdigit()==True:

        for i2 in range(len(bylfun)):

            if '0' in bylfun[i2] or '1' in bylfun[i2]:

                continue

            else: bylfun=input('Введите цифры б.ф. для 3-ех переменных(8 цифр состоящие из 0 и 1): ')

        u=1

    else:

        bylfun=input('Введите цифры б.ф. для 3-ех переменных(8 цифр): ')

print('Всё правильно! Ваша функция:')

print(bylfun)

print("")

print(Table(bylfun))

print("")

print("Метод треугольника:")

post = Triangle(bylfun)

print("")

print("Обратный треугольник:")

Triangle(post)

print("")

print('Метод БПФ:')

post = Foure(bylfun)

print("")

print('Метод БПФ, но обратно:')

Foure(post)

print("")

Изображение выглядит как символ, логотип, зарисовка, эмблема

Автоматически созданное описание

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

|  |  |
| --- | --- |
| Факультет | Информатика и вычислительная техника |
|  | (наименование факультета) |
| Кафедра | Кибербезопасность информационных систем |
|  | (наименование кафедры) |

**Лабораторная работа №3**

**по дисциплине «Дискретная математика»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент |  | |  |  |
|  | (подпись, дата) | |  | (ФИО, группа) |
| Направление подготовки | | 10.05.01 Компьютерная безопасность | | |
| Профиль | Информатика и вычислительная техника | | | |
| Преподаватель |  | |  | ст. пр. Типаева Э. Р. |
|  | (подпись, дата) | |  |  |

Ростов-на-Дону

2023

**Лабораторная работа №3**

**Цель работы:** Изучение принципа шифрования с использованием метода многомерной перестановки, на основе маршрутных перестановок

**Задание для лабораторной работы:**

1. Изучить программу Шифратор «Маршрутные перестановки»
2. Закодировать с помощью программы: ФИО, Кафедру
3. Раскодировать полученный текст

**Ход работы:**

1. Изучил программу Шифратор «Маршрутные перестановки».

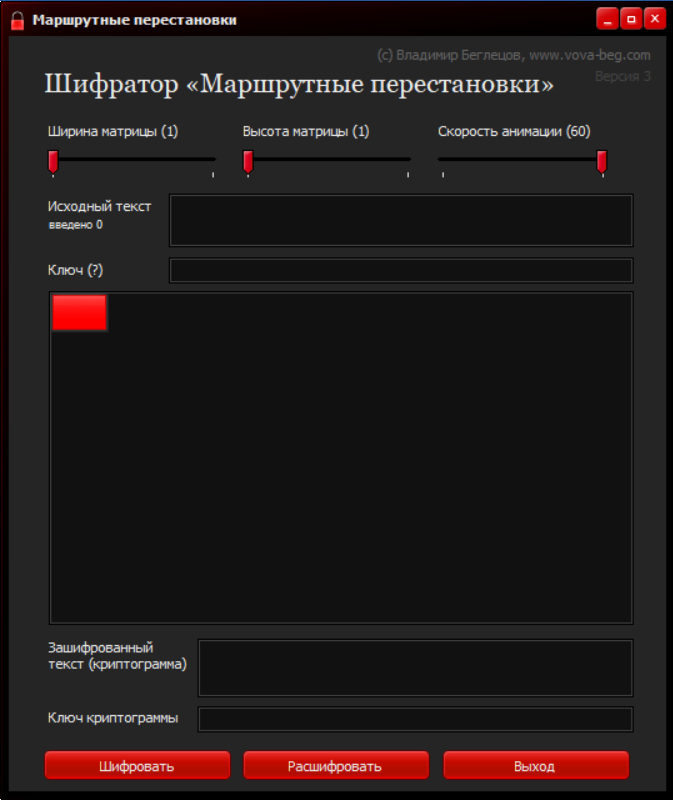


Рис. 1 – Окно программы шифратора «Маршрутные перестановки»

В окне «Исходный текст» записывается текст, который требуется закодировать. В строке «Ключ» записывается ключ, которым будет шифроваться текст и по количеству символов строиться матрица для шифровки. После шифровки в окне «Зашифрованный текст» будет содержаться уже закодированный текст, или можно самому написать текст, который нужно раскодировать. В строке «Ключ криптограммы» надо записать ключ для расшифровки полученного или записанного сообщения, соблюдая регистр букв.

1. С помощью программы кодирую ФИО и кафедру. Чтобы текст правильно зашифровался следует сначала задумать ключ и посчитать количество знаков в нем. Затем составить текст, в котором количество знаков будет кратно квадрату количеству знаков в ключе. Иначе во вовремя кодирования к концу текста добавятся случайные буквы. Также размером ключа задается квадратная матрица. Исходным текстом будет являться: КовалецИльяАндреевичКафКБиС, длиной 27 знаков, а ключом – ВКБ. Так как в моем случае длина ключа равна 3, то сам текст кратен и матрица будет размером 33.

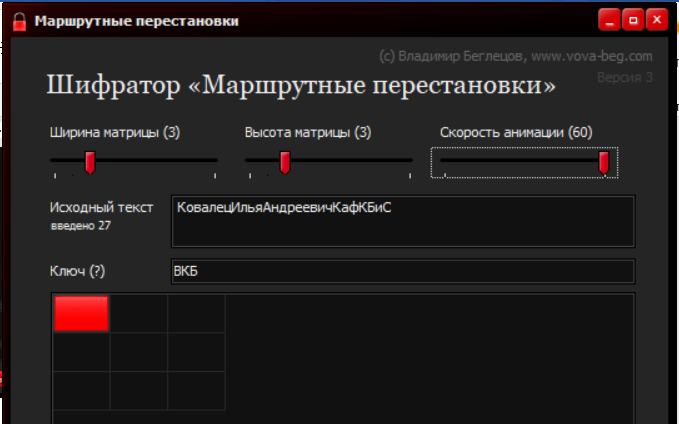


Рис. 2 – В поля введен текст и построена матрица

После нажатия кнопки «Шифровать» начнется процесс шифрования, скоростью которого можно управлять ползунком «Скорость анимации».

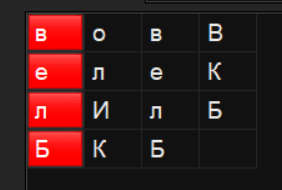


Рис. 3 – Процесс шифрования

Сначала построчно записывается каждая буква пока не закончится массив и по краям массива выписывается ключ. При таком тексте в массив поместилось КовалецИл, т.е. 9 букв. Далее шифратор меняет столбцы местами в соответствии с алфавитным порядком в ключе. Затем шифратор меняет строки местами в соответствии с алфавитным порядком в ключе. В заключительном этапе выписывается буквенная последовательность по столбцам сверху вниз. Этот процесс будет повторяться то количество раз пока длина текста не будет вся закодирована. В моем случае 3 раза.

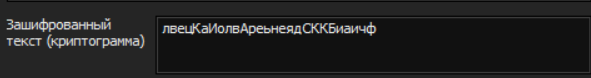


Рис. 4 – Полностью закодированный текст

1. Раскодировал полученный текст. В поле «Ключ криптограммы» записываю ключ, соблюдая регистр букв. Записываю полученный текст. После нажатия кнопки «Расшифровать» начнется процесс расшифровки.

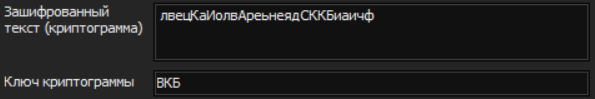


Рис. 5 – Готовая к расшифровке программа

Сначала записывается каждая буква в столбцы сверху вниз пока не закончится массив и по краям массива выписывается ключ в алфавитном порядке. Далее шифратор меняет строки местами в соответствии с правильным (изначальным) порядком в ключе.



Рис. 6 – Процесс расшифровки

Затем шифратор меняет столбцы местами в соответствии с правильным (изначальным) порядком в ключе. В заключительном этапе выписывается буквенная последовательность построчно слева направо начиная сверху. Этот процесс будет повторяться то количество раз пока длина закодированного текста не будет вся закодирована.



Рис. 7 – Расшифрованный текст

Изображение выглядит как символ, логотип, зарисовка, эмблема

Автоматически созданное описание

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

|  |  |
| --- | --- |
| Факультет | Информатика и вычислительная техника |
|  | (наименование факультета) |
| Кафедра | Кибербезопасность информационных систем |
|  | (наименование кафедры) |

**Лабораторная работа №4**

**по дисциплине «Дискретная математика»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент |  | |  |  |
|  | (подпись, дата) | |  | (ФИО, группа) |
| Направление подготовки | | 10.05.01 Компьютерная безопасность | | |
| Профиль | Информатика и вычислительная техника | | | |
| Преподаватель |  | |  | ст. пр. Типаева Э. Р. |
|  | (подпись, дата) | |  |  |

Ростов-на-Дону

2023

**Лабораторная работа №4**

**Цель работы:** Изучение принципа шифрования с использованием метода сдвигов регистра, на основе бинарных побитных сложений

**Задание для лабораторной работы:**

1. Изучить программу для шифрования методом сдвигов регистра
2. Закодировать с помощью программы численную последовательность
3. Раскодировать полученный текст

**Ход работы:**

1. Изучил программу для шифрования методом сдвигов регистра.

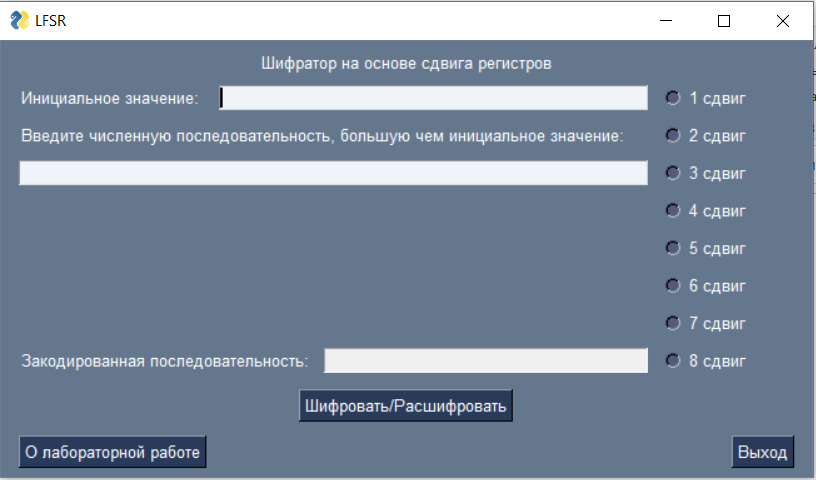


Рис. 1 – Окно программы шифрования

В поле «Инициальное значение» по заданию надо ввести бинарной системе счисления номер своего варианта, либо если не учитывать задание любое бинарное число, у которого длина больше 4, иначе слева от числа будет добавлены 0. В поле для численной последовательности, надо вписать любую числовую последовательность, у которой длина должна быть строго больше, чем длина инициального значения, иначе будет выведена ошибка. В флажках 1-8 сдвиг надо выбрать с каким сдвигом будет шифроваться числовая последовательность. В поле «Закодированная последовательность» после нажатия «Шифровать/Расшифровать» выведется закодированный текст. По нажатию кнопки «О лабораторной работе» выведется сообщение о том, что надо сделать в этой работе. После нажатия кнопки «Выход» программа завершится.

1. Закодировал с помощью программы численную последовательность. Для этого ввожу свой номер в бинарной системе счисления в поле и любую последовательность во второе поле. По таблице смотрю, что мои сдвиги это 1, 5, 8. Выбираю сдвиг и нажимаю «Шифровать/Расшифровать».

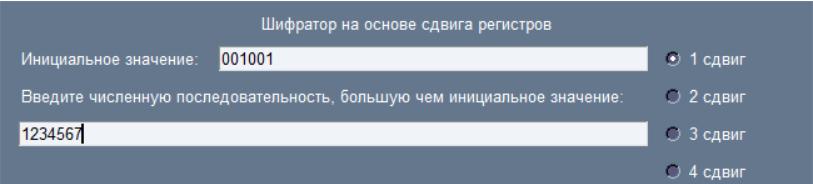


Рис. 2 – Данные введены в поля

Начинается процесс шифровки. Записывается инициальное значение как «iz», сдвиг как ключ записывается как «bk» и числовая последовательность как «bt». Происходит создание нового инициального значения (НИЗ) и вспомогательное значение для подсчета индекса для НИЗ в побитном сложении. Далее цикл будет выполняться пока длина НИЗ не равна длине текста, для добавления длины для НИЗ. Вводится вспомогательное значение «а» для подсчета НИЗ. В цикле for идет пересчет «count» и «с», который берем в числовой последовательности. Происходит побитное бинарное сложение («с» умноженное с значением НИЗ по индексу «schetchik» + «count»).

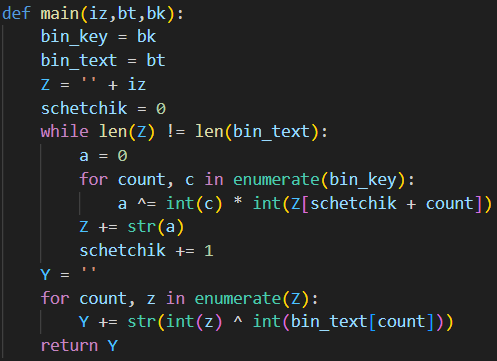


Рис. 3 – Код шифратора

После прохождения всего цикла значение «а» добавляется в конец НИЗ. После n прохода «schetchik» увеличивается на n чтобы сместить ключ на n элементов при переборе в цикле. После того как длина НИЗ достигла длины числовой последовательности, вводится переменная «Y» для итогового значения. В цикле for идет пересчет «count» и «z», который берем из НИЗ. Далее в переменной «Y» происходит сложение к итоговому значению побитного сложения из «z» и значения из числовой последовательности по индексу «count». И в заключении происходит вывод переменной «Y».



Рис. 4 – Вывод результата при 1 сдвиге



Рис. 5 – Вывод результата при 5 сдвиге



Рис. 6 – Вывод результата при 8 сдвиге

1. Раскодировал полученную последовательность также как закодировал, через побитное сложение еще раз.

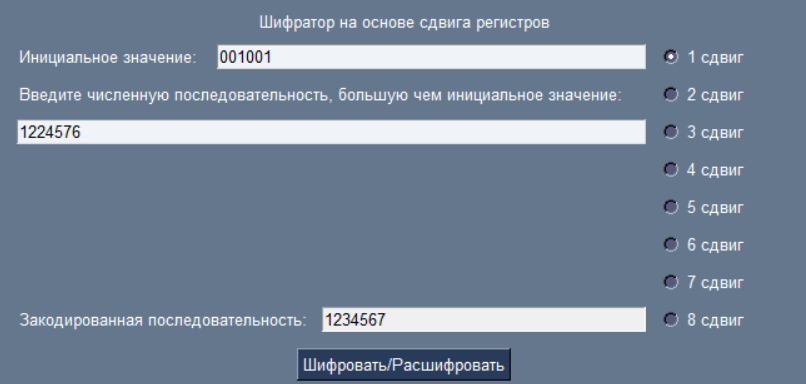


Рис. 7 – Расшифровка при 1 сдвиге

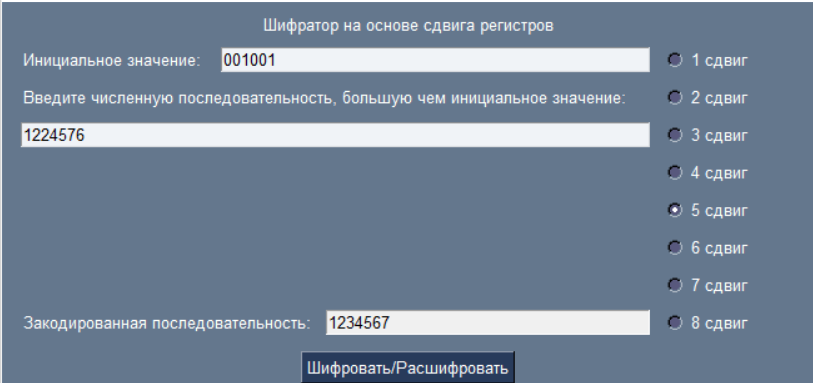


Рис. 8 – Расшифровка при 5 сдвиге

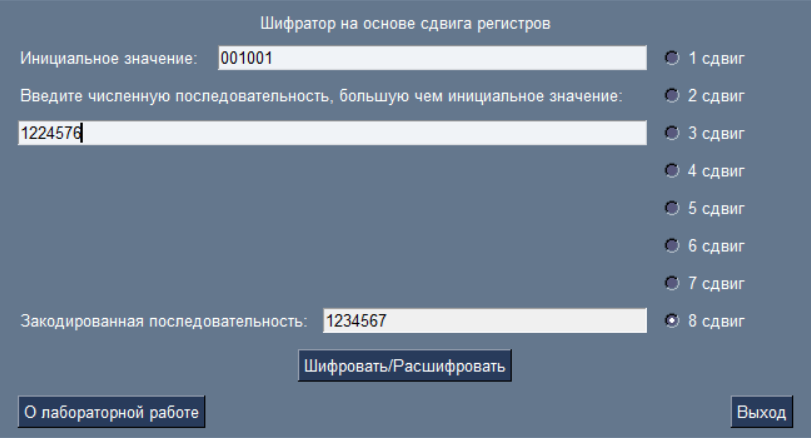


Рис. 9 – Расшифровка при 8 сдвиге