Отчёт по лабораторной работе №2

Шифры перестановки

Степанов Виктор НПМмд-02-22

Содержание

[Цель работы 1](#_Toc115450833)

[Теоретические сведения 1](#_Toc115450834)

[Шифр маршрутной перестановки 1](#_Toc115450835)

[Шифр Кардано 2](#_Toc115450836)

[Шифр Виженера 2](#_Toc115450837)

[Выполнение работы 2](#_Toc115450838)

[Реализация шифра маршрутной перестановки на языке Python 2](#_Toc115450839)

[Реализация шифра решеткой на языке Python 3](#_Toc115450840)

[Реализация шифра Виженера на языке Python 5](#_Toc115450841)

[Контрольный пример 7](#_Toc115450842)

[Выводы 8](#_Toc115450843)

[Список литературы 9](#_Toc115450844)

# Цель работы

Изучение алгоритмов маршрутной перестановки, решеток и Виженера

# Теоретические сведения

## Шифр маршрутной перестановки

Широкое распространение получили шифры перестановки, использующие некоторую геометрическую фигуру. Преобразования из этого шифра состоят в том, что в фигуру исходный текст вписывается по ходу одного “маршрута”, а затем по ходу другого выписывается с нее. Такой шифр называют маршрутной перестановкой. Например, можно вписывать исходное сообщение в прямоугольную таблицу, выбрав такой маршрут: по горизонтали, начиная с левого верхнего угла поочередно слева направо и справа налево. Выписывать же сообщение будем по другому маршруту: по вертикали, начиная с верхнего правого угла и двигаясь поочередно сверху вниз и снизу вверх.

## Шифр Кардано

Решётка Кардано — инструмент кодирования и декодирования, представляющий собой специальную прямоугольную (в частном случае — квадратную) таблицу-карточку, четверть ячеек которой вырезана.

Таблица накладывается на носитель, и в вырезанные ячейки вписываются буквы, составляющие сообщение. После переворачивания таблицы вдоль вертикальной оси, процесс вписывания букв повторяется. Затем то же самое происходит после переворачивания вдоль горизонтальной и снова вдоль вертикальной осей.

В частном случае квадратной таблицы, для получения новых позиций для вписывания букв, можно поворачивать квадрат на четверть оборота.

Чтобы прочитать закодированное сообщение, необходимо наложить решётку Кардано нужное число раз на закодированный текст и прочитать буквы, расположенные в вырезанных ячейках.

Такой способ шифрования сообщения был предложен математиком Джероламо Кардано в 1550 году, за что и получил своё название.

## Шифр Виженера

Шифр Виженера (фр. Chiffre de Vigenère) — метод полиалфавитного шифрования буквенного текста с использованием ключевого слова.

Этот метод является простой формой многоалфавитной замены. Шифр Виженера изобретался многократно. Впервые этот метод описал Джован Баттиста Беллазо (итал. Giovan Battista Bellaso) в книге La cifra del. Sig. Giovan Battista Bellasо в 1553 году, однако в XIX веке получил имя Блеза Виженера, французского дипломата. Метод прост для понимания и реализации, он является недоступным для простых методов криптоанализа.

В шифре Цезаря каждая буква алфавита сдвигается на несколько строк; например в шифре Цезаря при сдвиге +3, A стало бы D, B стало бы E и так далее. Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова.

# Выполнение работы

## Реализация шифра маршрутной перестановки на языке Python

import sys  
def prints(lists):  
 for i in lists:  
 for j in i:  
 print(j, end = " ")  
 print()  
def marshshifr():  
 text = input("Input").replace(" ", "")  
 n = int(input("Введите число n"))  
 m = int(input("Введите число m"))  
 parol = input("Введите пароль (слово)")  
 lists = [['a' for i in range(0, n)] for j in range(m)]  
 it = 0  
 for i in range(m):  
 for j in range(n):  
 if it < len(text):  
 lists[i][j] = text[it]  
 it += 1  
 lis = list()  
 for i in range(n):  
 lis.append(parol[i])  
 lists.append(lis)  
 prints(lists)  
 result = ""  
 spisok = sorted(lists[len(lists) - 1])  
 for i in spisok:  
 print(i, " = ", lists[len(lists) - 1].index(i))  
 for j in range(len(lists)):  
 if j == len(lists)-1:  
 continue  
 result += lists[j][lists[len(lists)-1].index(i)]  
 print(result)

## Реализация шифра решеткой на языке Python

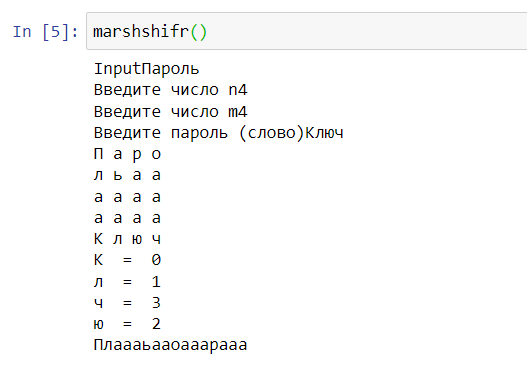
def delete(largelist,inn,k):  
 for i in range(k\*2):  
 for j in range(k\*2):  
 if largelist[i][j] == inn:  
 largelist[i][j] = ' '  
 return  
  
def povorot(matrix):  
 return[list(reversed(col)) for col in zip(\*matrix)]  
  
def reshetka():  
 k = int(input('Введите k'))  
 s = 1  
 lists = [[i for i in range(k)] for i in range(k)]  
 for i in range(k):  
 for j in range(k):  
 lists[i][j] = s  
 s += 1  
 print(lists)  
 lists1 = povorot(lists)  
 lists2 = povorot(lists1)  
 lists3 = povorot(lists2)  
 largelist = [[1 for i in range(2\*k)] for i in range(2\*k)]  
 for i in range(k):  
 for j in range(k):  
 largelist[i][j] = lists[i][j]  
 i1 = 0  
 j1 = 0  
 for i in range(0, k):  
 for j in range(k, k\*2):  
 largelist[i][j] = lists1[i1][j1]  
 j1 += 1  
 j1 = 0  
 i1 += 1  
 i1 = 0  
 j1 = 0  
 for i in range(k, k\*2):  
 for j in range(k, k\*2):  
 largelist[i][j] = lists2[i1][j1]  
 j1 += 1  
 j1 = 0  
 i1 += 1  
 i1 = 0  
 j1 = 0  
 for i in range(k, k\*2):  
 for j in range(0, k):  
 largelist[i][j] = lists3[i1][j1]  
 j1 += 1  
 j1 = 0  
 i1 += 1  
 prints(largelist)  
 text = 'договорподписали'  
 largelist1 = [[' ' for i in range(2\*k)] for i in range(2\*k)]  
 s = 0  
 li = [i for i in range(1, k\*\*2 + 1)]  
 for inn in li:  
 delete(largelist, inn, k)  
 ind = 0  
 for i in range(k\*2):  
 for j in range(k\*2):  
 if largelist[i][j] == largelist1[i][j] and len(text) > 0:  
 largelist1[i][j] = text[0]  
 text = text[1:]  
 largelist = povorot(largelist)  
 for i in range(k\*2):  
 for j in range(k\*2):  
 if largelist[i][j] == largelist1[i][j] and len(text) > 0:  
 largelist1[i][j] = text[0]  
 text = text[1:]  
 if len(text) > 0:  
 largelist = povorot(largelist)  
 for i in range(k\*2):  
 for j in range(k\*2):  
 if largelist[i][j] == largelist1[i][j] and len(text) > 0:  
 largelist1[i][j] = text[0]  
 text = text[1:]  
 if len(text) > 0:  
 largelist = povorot(largelist)  
 for i in range(k\*2):  
 for j in range(k\*2):  
 if largelist[i][j] == largelist1[i][j] and len(text) > 0:  
 largelist1[i][j] = text[0]  
 text = text[1:]  
 prints(largelist1)  
 string = input('Введите пароль')  
 if len(string) > k\*2:  
 string = string[:k\*2]  
 elif len(string) < k\*2:  
 while len(string) != k\*2:  
 string += 'z'  
 largelist1.append(list(string))  
 prints(largelist1)  
 result = ''  
 spisok = sorted(largelist1[len(largelist1) - 1])  
 for i in spisok:  
 print(i, ' = ', largelist1[len(largelist1) - 1].index(i))  
 for j in range(len(largelist1)):  
 if j == len(largelist1) - 1:  
 continue  
 result += largelist1[j][largelist1[len(largelist1) - 1].index(i)]  
 print(result.replace(' ', ''))

## Реализация шифра Виженера на языке Python

def form\_dict():  
 d = {}  
 iter = 0  
 for i in range(0, 127):  
 d[iter] = chr(i)  
 iter = iter + 1  
 return d  
  
def encode\_val(word):  
 list\_code = []  
 lent = len(word)  
 d = form\_dict()  
 for w in range(lent):  
 for value in d:  
 if word[w] == d[value]:  
 list\_code.append(value)  
 return list\_code  
  
def comp(value, key):  
 len\_key = len(key)  
 dic = {}  
 iter = 0  
 full = 0  
 for i in value:  
 dic[full] = [i, key[iter]]  
 full = full + 1  
 iter = iter + 1  
 if (iter >= len\_key):  
 iter = 0  
 return dic  
  
def full\_encode(value, key):  
 dic = comp(value, key)  
 print('Compare full encode', dic)  
 lis = []  
 d = form\_dict()  
 for v in dic:  
 go = (dic[v][0]+dic[v][1]) % len(d)  
 lis.append(go)  
 return lis  
  
def decode\_val(list\_in):  
 list\_code = []  
 lent = len(list\_in)  
 d = form\_dict()  
 for i in range(lent):  
 for value in d:  
 if list\_in[i] == value:  
 list\_code.append(d[value])  
 return list\_code  
  
def full\_decode(value, key):  
 dic = comp(value, key)  
 print('Deshifr = ', dic)  
 d = form\_dict()  
 lis = []  
 for v in dic:  
 go = (dic[v][0]-dic[v][1]+len(d)) % len(d)  
 lis.append(go)  
 return lis  
  
def vijiner():  
 word = "Hello world"  
 key = "key"  
 sys.stdout.write(word)  
 sys.stdout.write(key)  
 key\_encoded = encode\_val(key)  
 value\_encoded = encode\_val(word)  
 sys.stdout.write(str(key\_encoded))  
 sys.stdout.write(str(value\_encoded))  
 shifr = full\_encode(value\_encoded, key\_encoded)  
 print('Шифр = ', ''.join(decode\_val(shifr)))  
 decoded = full\_decode(shifr, key\_encoded)  
 print('Decode list = ', decoded)  
 decode\_word\_list = decode\_val(decoded)  
 print('Word = ', ''.join(decode\_word\_list))

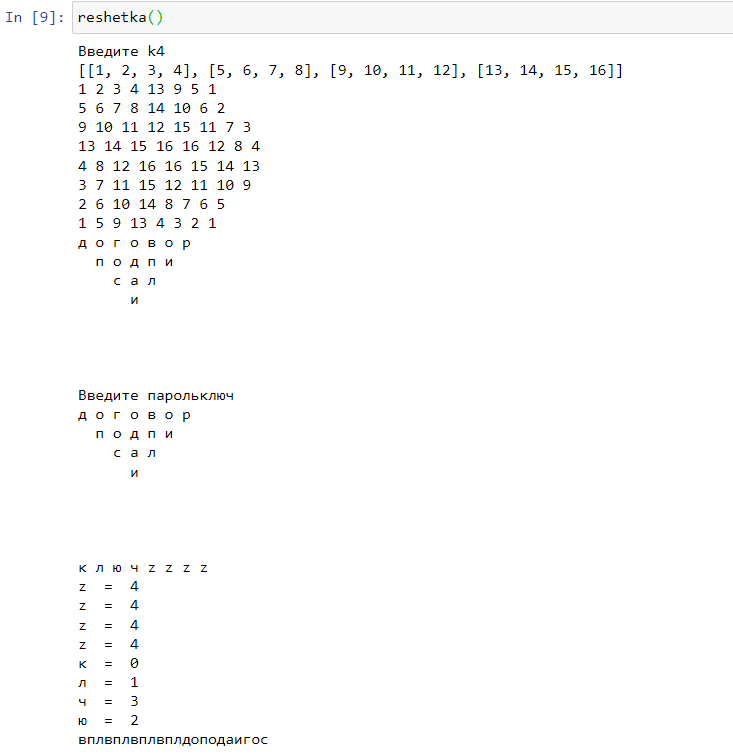
## Контрольный пример

Работа алгоритма маршрутной перестановки (рис. 1)



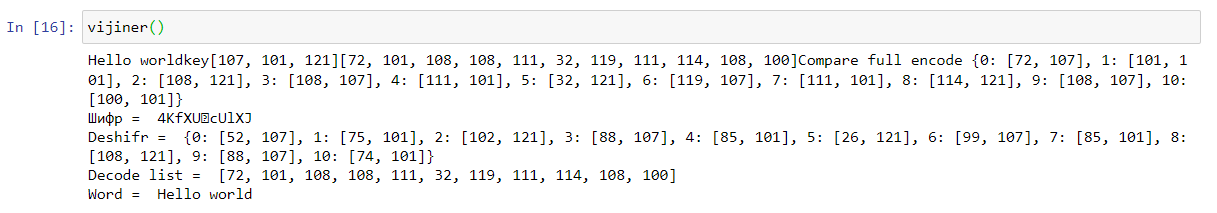
Пример 1

Работа алгоритма решетки (рис. 2)



Пример 2

Работа алгоритма Виженера (рис. 3)



Пример 3

# Выводы

Изучил алгоритмы шифрования с помощью перестановок.