Вариант **4**

Цель работы

Изучение основных принципов шифрования информации, знакомство с широко известными алгоритмами шифрования, приобретение навыков их программной реализации

Задание

Реализовать в программе шифрование и дешифрацию файла с использованием квадрата Полибия, обеспечив его случайное заполнение.

Описание

В криптографии квадрат Полибия (англ. **Polybius square**), также известный как шахматная доска Полибия — оригинальный код простой замены, одна из древнейших систем кодирования, предложенная Полибием (греческий историк, полководец, государственный деятель, III век до н. э.). Данный вид кодирования изначально применялся для греческого алфавита, но затем был распространен на другие языки.

Способ шифрования

К каждому сообщению отдельно составляется таблица шифрования с одинаковым (не обязательно) количеством пронумерованных строк и столбцов, параметры которой зависят от ее мощности (количества различных символов в сообщении). Берутся два целых числа, произведение которых ближе всего к количеству букв в языке — получаем нужное число строк и столбцов. Затем вписываем в таблицу все символы сообщения подряд — по одной в каждую клетку.

Пример с русским алфавитом

	1	2	3	4	5	6
1	А	Б	В	Г	Д	Е
2	Ё	Ж	З	И	Й	К
3	Л	М	Н	О	П	Р
4	С	Т	У	Ф	Х	Ц
5	Ч	Ш	Щ	Ъ	Ы	Ь
6	Э	Ю	Я	-	-	-

Сообщение далее преобразуется в координаты соответственно составленному квадрату Полибия

Закодируем фразу **"He was a boy, she was a girl, can I make it anymore obvious?"**

Создание квадрата и наполнение его символами из сообщения

In [5]:

```
import random
characters = []
phrase = ""

# Добавляем каждый символ в массив символов
```

```
with open("phrase.txt") as fileobj:
    for line in fileobj:
       phrase+=(line)
       for ch in line:
           if ch not in characters:
            characters.append(ch)

# Определяем необходимую сторону квадрата
square_side = 0
i = 2
while (square_side == 0):
    if i*i > len(characters):
        square_side = i
    else:
        i += 1


square = {}

# Наполнение квадрата символами
for i in range(1, square_side+1):
    for j in range(1,square_side+1):
        key = str(i) + str(j)
        # Если символы закончились дополняем квадрат пустыми ячейками
        if (len(characters) == 0):
            value = ""
        else:
            value = characters.pop(random.randint(0, len(characters)-1))
        square[value] = key

# Принтим квадрат со значениями
just_a_counter = 1
for k, v in square.items():
    print('[' + v + "]: " + k, end=" ")
    just_a_counter = just_a_counter + 1
    if (just_a_counter == square_side+1):
        just_a_counter=1
        print("")
```

```
[11]: g [12]: , [13]: I [14]: k [15]: ?
[21]: s [22]: h [23]: n [24]: y [25]: v
[31]: r [32]: m [33]: a [34]: i [35]: l
[41]: t [42]: o [43]: e [44]: b [45]: w
[51]:   [52]: c [53]: H [54]: u [55]:
```

Шифрование сообщения

In [3]:

```
coded_msg = ""

# Заменяем каждый символ соответствующим значением из квадрата
list_phrase = list(phrase)
for x in phrase:
    if x in square:
        coded_msg += square.get(x)
    else:
        coded_msg += (x + x)

print(coded_msg)
```

```
241435514354354335251222423554341435514354354335232131134235334352353235414311143521453543522411231143512251521125354 44
```

Дешифрование сообщения

In [4]:

```
# Дешифрование
```

```
# декодирование
decoded_msg = ""
list_phrase = []
step = 2
# Помещаем закодированное сообщение в массив через два символа
for i in range(0, len(coded_msg), 2):
    list_phrase.append(coded_msg[i:step])
    step += 2
key_square_list = list(square.keys())
val_square_list = list(square.values())

 # Достаем символы из квадрата по их координатам
for x in list_phrase:
    if x in val_square_list:
        i = val_square_list.index(x)
        decoded_msg += key_square_list[i]
    else:
        decoded_msg += x[0:1]

print(decoded_msg)
```

```
He was a boy, she was a girl, can I make it anymore obvious?
```

Вывод

В ходе выполнения лабораторной работы был изучен метод шифрование и дешифрования файла с использованием метода квадрата Полибия. Была реализована программа для шифрования и дешифрования текста с использованием данного метода.