

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»
Кафедра информатики

Отчет по лабораторной работе №1
Шифр Цезаря.

Выполнил:
Брычиков Д.Д.
Проверил:
Чернявский Ю. А.

Минск 2018

Введение

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 4 А была бы заменена на Г, Б станет Д, и так далее.

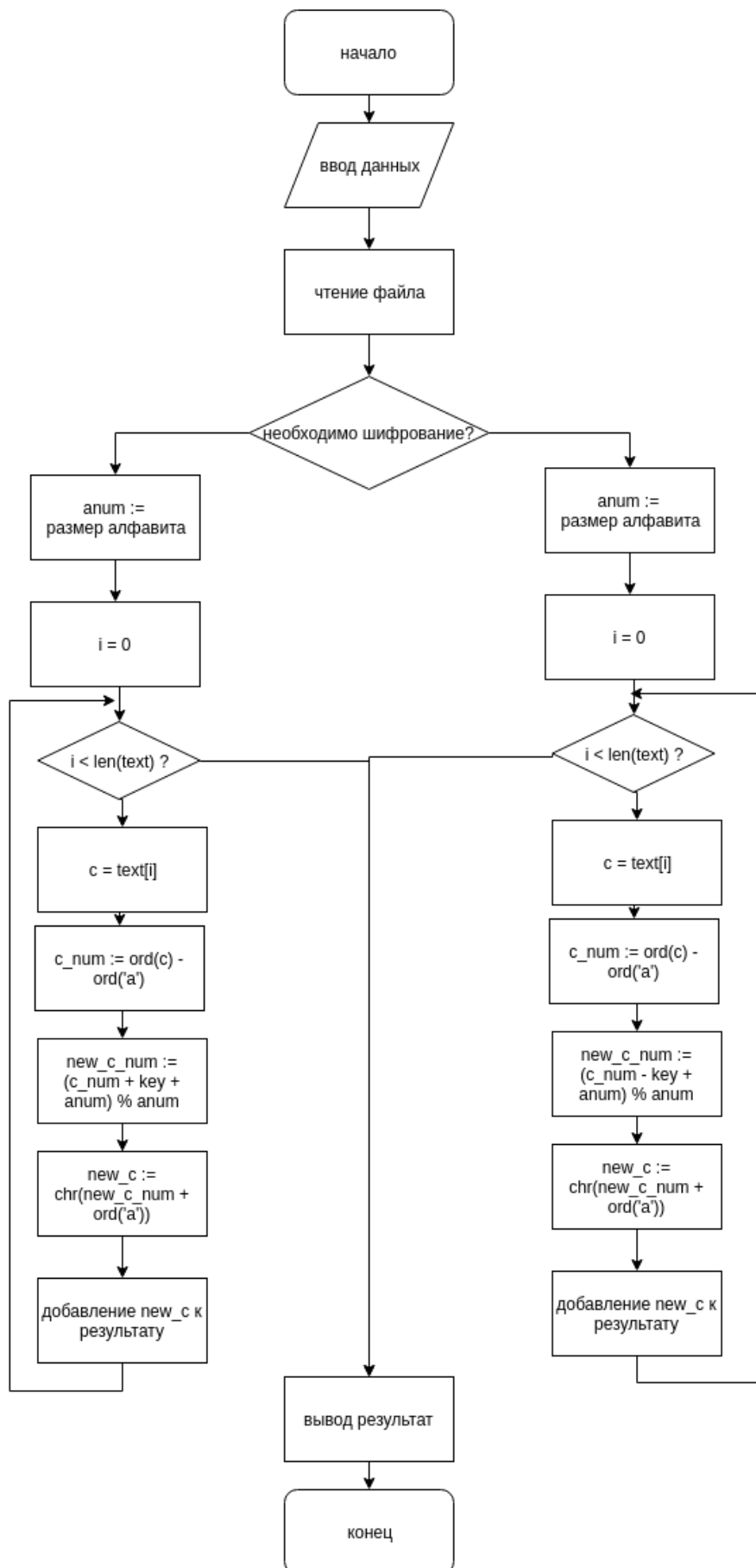
Шифр Виженера

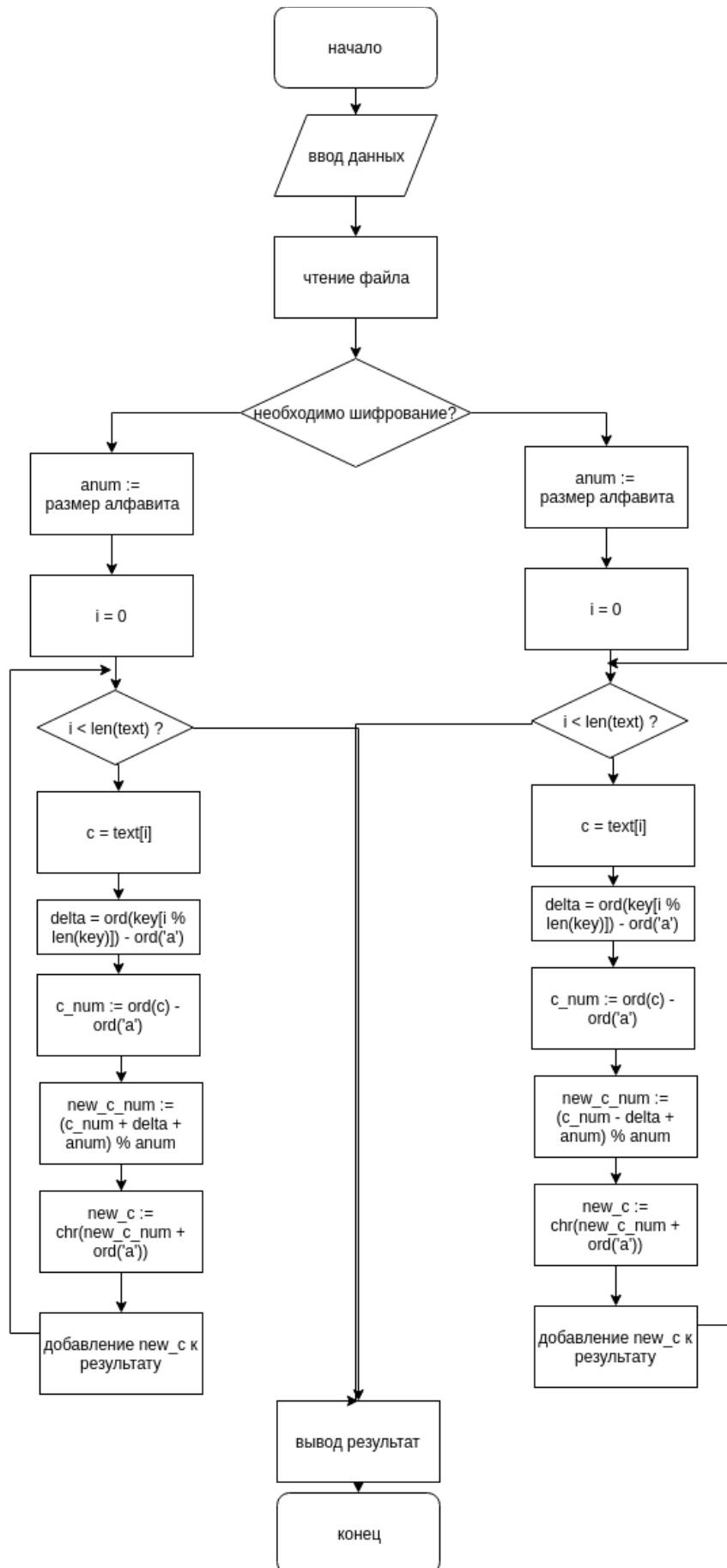
Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая *tabula recta* или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, 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.

Постановка задачи

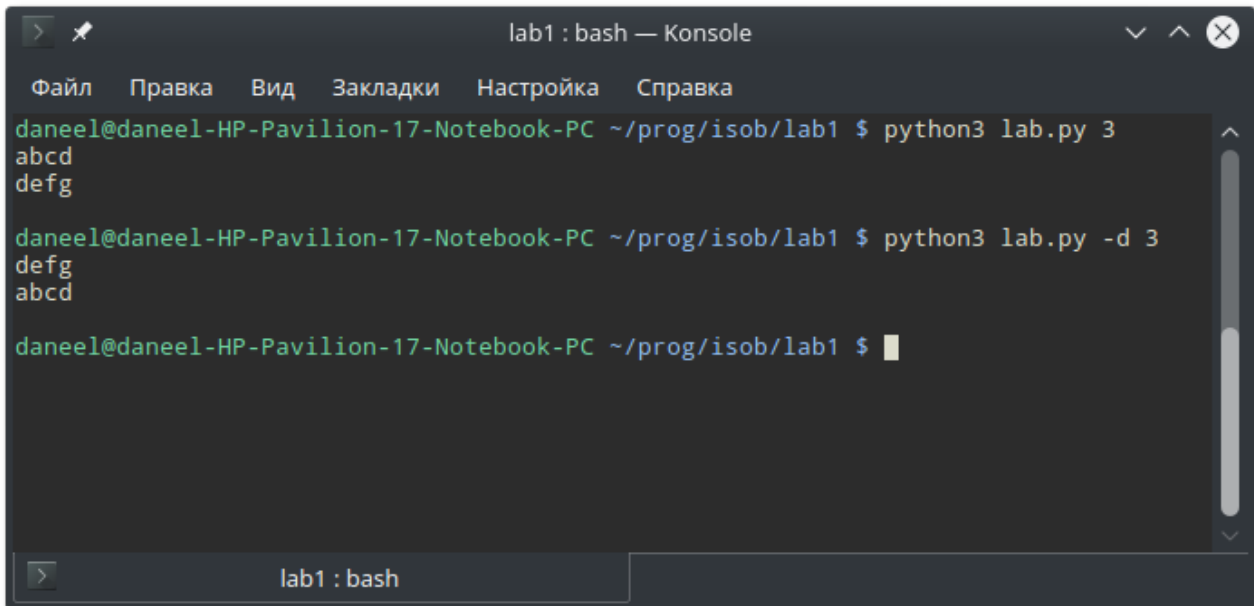
Реализовать программные средства шифрования и дешифрования текстовых файлов при помощи Шифра Цезаря, (шифра сдвига, кода Цезаря) и шифра Виженера.

Схема алгоритма





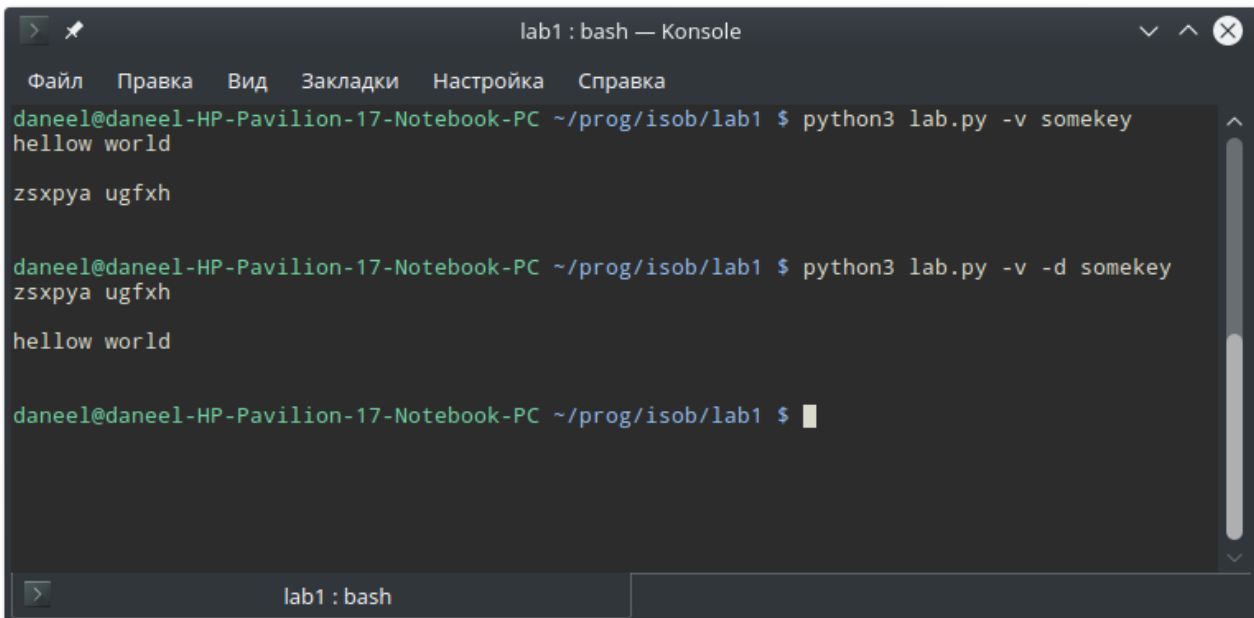
Демонстрация работы программы



```
lab1 : bash — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка
daneel@daneel-HP-Pavilion-17-Notebook-PC ~/prog/isob/lab1 $ python3 lab.py 3
abcd
defg

daneel@daneel-HP-Pavilion-17-Notebook-PC ~/prog/isob/lab1 $ python3 lab.py -d 3
defg
abcd

daneel@daneel-HP-Pavilion-17-Notebook-PC ~/prog/isob/lab1 $
```



```
lab1 : bash — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка
daneel@daneel-HP-Pavilion-17-Notebook-PC ~/prog/isob/lab1 $ python3 lab.py -v somekey
hellow world

zsxpya ugfxh

daneel@daneel-HP-Pavilion-17-Notebook-PC ~/prog/isob/lab1 $ python3 lab.py -v -d somekey
zsxpya ugfxh

hellow world

daneel@daneel-HP-Pavilion-17-Notebook-PC ~/prog/isob/lab1 $
```

Исходный код

```
def encrypt(s, key):
    anum = ord('z') - ord('a') + 1
    res = []
    for c in s:
        if c > 'z' or c < 'a':
            res.append(c)
            continue

        c_num = ord(c) - ord('a')
        new_c_num = (c_num + key + anum) % anum
        new_c = chr(new_c_num + ord('a'))
        res.append(new_c)
    return ''.join(res)
```

```
def decrypt(s, key):
    anum = ord('z') - ord('a') + 1
    res = []
    for c in s:
        if c > 'z' or c < 'a':
            res.append(c)
            continue

        c_num = ord(c) - ord('a')
        new_c_num = (c_num - key + anum) % anum
        new_c = chr(new_c_num + ord('a'))
        res.append(new_c)
    return ''.join(res)
```

```
def encrypt(s, key):
    anum = ord('z') - ord('a') + 1
    res = []
    i = 0
    for c in s:
        if c > 'z' or c < 'a':
            res.append(c)
            continue

        c_num = ord(c) - ord('a')
        delta = ord(key[i % len(key)]) - ord('a')
        new_c_num = (c_num + delta + anum) % anum
        new_c = chr(new_c_num + ord('a'))
        res.append(new_c)
        i += 1
    return ''.join(res)
```

```
def decrypt(s, key):
    anum = ord('z') - ord('a') + 1
    res = []
    i = 0
```

```

for c in s:
    if c > 'z' or c < 'a':
        res.append(c)
        continue

    c_num = ord(c) - ord('a')
    delta = ord(key[i % len(key)]) - ord('a')
    new_c_num = (c_num - delta + anum) % anum
    new_c = chr(new_c_num + ord('a'))
    res.append(new_c)
    i += 1
return ''.join(res)

```

```

import cesar
import visin
import argparse
import sys

```

```

if __name__ == '__main__':
    ap = argparse.ArgumentParser()
    ap.add_argument('-v', action='store_const', const=True)
    ap.add_argument('-d', action='store_const', const=True)
    ap.add_argument('key')
    args = ap.parse_args()

    if args.v:
        module = visin
        key = args.key
    else:
        module = cesar
        key = int(args.key)

    if args.d:
        fun = module.dencrypt
    else:
        fun = module.encrypt

    inp = sys.stdin.read()
    out = fun(inp, key)
    print(out)

```