

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
КАФЕДРА «ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ»

Программирование криптографических алгоритмов
Блок А: Шифры однозначной замены

Выполнила студентка 3 курса группы 171-341

Решетникова Дарья

Москва 2020 г.

Аннотация

Язык : Python

Программа: Visual Studio 2017

Пословица: Плод никогда не падает далеко от дерева.

Текст: История, как и любая учебная дисциплина, имеет свой объект и предмет исследования. Объектом исследования является исторический процесс - последовательный процесс развития природы и общества, череда сменяющих друг друга событий, в которых проявляется деятельность многих поколений людей. С каждым мгновением времени, ушедшим в прошлое исторический процесс дополняется совокупностью новых событий, явлений, фактов и факторов в жизни человека, семьи, этноса, государства, человечества. Подобно тому, как материальный мир переживает процессы негэнтропии и энтропии, природа вокруг нас - процессы эволюции и инволюции, человеческое общество характеризуют процессы социального прогресса и регресса. Предметом изучения истории является деятельность главного его субъекта - человека - в прошлом. В узком смысле история представляет собой совокупность социальных фактов, знаний, представлений ученых о том, как эти процессы осуществлялись и осуществляются. Долгое время история существовала скорее, как литература и искусство, обслуживающие интересы власти, нежели наука. Не случайно в греческой мифологии в свите античного бога Аполлона, покровителя искусств и наук, среди других муз была Клио - муза истории.

SEO-АНАЛИЗ ТЕКСТА

FAQAPI проверки

история, как и любая учебная дисциплина, имеет свой объект и предмет исследования. Объектом исследования является исторический процесс - последовательный процесс развития природы и общества, череда сменяющих друг друга событий, в которых проявляется деятельность многих поколений людей. С каждым мгновением времени, ушедшим в прошлое исторический процесс дополняется совокупностью новых событий, явлений, фактов и факторов в жизни человека, семьи, этноса, государства, человечества. Подобно тому, как материальный мир переживает процессы негэнтропии и энтропии, природа вокруг нас – процессы эволюции и инволюции, человеческое общество характеризуют процессы социального прогресса и регресса. Предметом изучения истории является деятельность главного его субъекта – человека - в прошлом. В узком смысле история представляет собой совокупность социальных фактов, знаний, представлений ученых о том, как эти процессы осуществлялись и осуществляются. Долгое время история существовала скорее, как литература и искусство, обслуживающие интересы власти, нежели наука. Не случайно в греческой мифологии в свите античного бога Аполлона, покровителя искусств и наук, среди других муз была Клио - муза истории.

Всего символов: 1190

Без пробелов: 1042

Количество слов: 149

Заказать текст

Проверить SEO-данные

Шифр простой замены АТБАШ

1. Описание шифра.

Атбáш (ивр. אֶתְבָּאֵשׁ) — простой шифр подстановки для алфавитного письма. Правило шифрования состоит в замене i -й буквы алфавита буквой с номером $n - i + 1$, где n — число букв в алфавите.

2. Алгоритм шифра.

$$Y_i = X_{(n-i+1)}$$

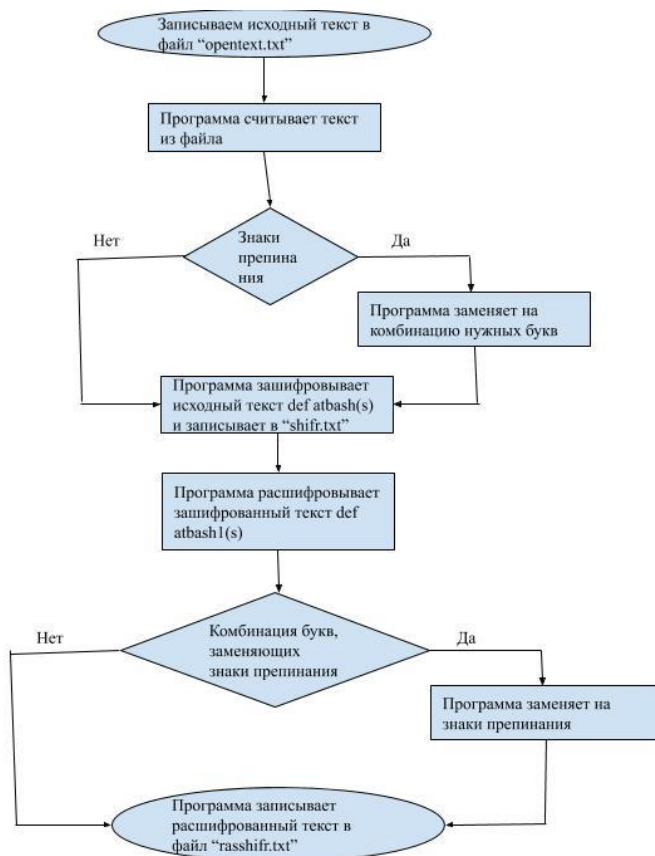
X – исходный (открытый) текст

Y– зашифрованный текст

i – порядковый номер буквы в открытом алфавите, $i=1 \dots n$

n – количество букв в открытом алфавите.

3. Блок-схема программы



4. Код программы

функция для зашифрования

def atbash(s):

abc = 'абвгдежзийклмнопрстуфхцчшщъыьэюя'

s = s.replace('.', 'тчк') # Если в сообщении попадетсся точка, она заменется на тчк

s = s.replace(',', 'зпт') # Если в сообщении попадетсся запятая, она заменяется на зпт

s = s.replace('-', 'тире') # Если в сообщении попадетсся - (тире), оно заменяется на тире

return s.translate(str.maketrans(

abc + abc.upper(), abc[::-1] + abc.upper()[::-1])) # функция шифрования

Метод translate() возвращает копию строки, в которой все символы были переведены

maketrans(intab, outtab) - замена одного алфавита на другой

.upper() - Преобразование строки к верхнему регистру

```
#[::-1] - срез для вывода строки в обратном порядке

# функция для расшифрования
def atbash1(s):
    abc = 'абвгдежзийклмнопрстуфхцшщъыьэюя'
    s = s.replace('них', '.') # это сочетания получается при шифровании тчк
    s = s.replace('шрн', ',') # это сочетания получается при шифровании зпт
    s = s.replace('нчпъ', '-') # это сочетания получается при шифровании тире
    return s.translate(str.maketrans(
        abc + abc.upper(), abc[::-1] + abc.upper()[::-1])) # функция расшифрования

# в этот файл нужно записать исходный текст
f = open(r"opentext.txt", "rt", encoding='utf-8')
text = f.read()

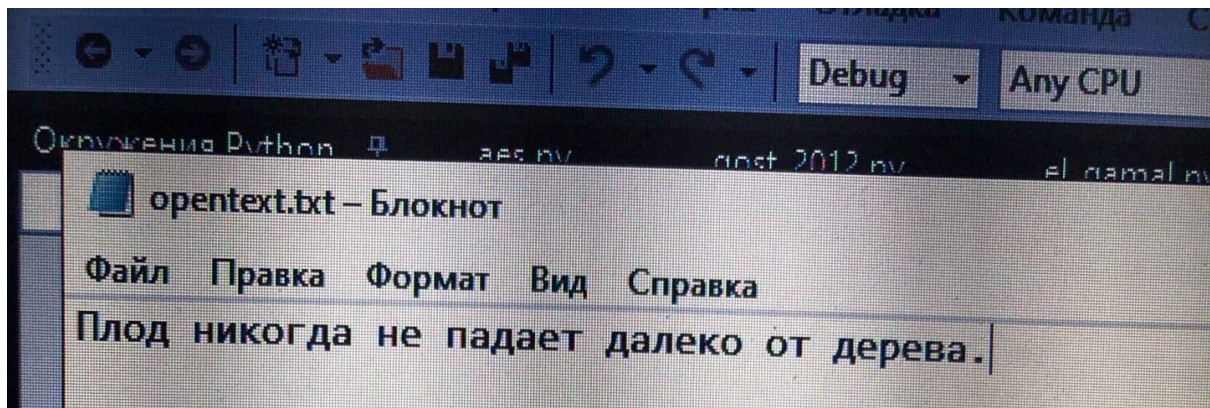
# в этот файл записывается зашифрованный текст
f = open('shifr.txt', 'wt', encoding='utf-8')
sh = atbash(str(text))
f.writelines(sh)
f.close()

# в этот файл записывается расшифрованный текст
file = open('rasshifr.txt', 'wt', encoding='utf-8')
rassh = atbash1(str(sh))
file.writelines(rassh)
file.close()
```

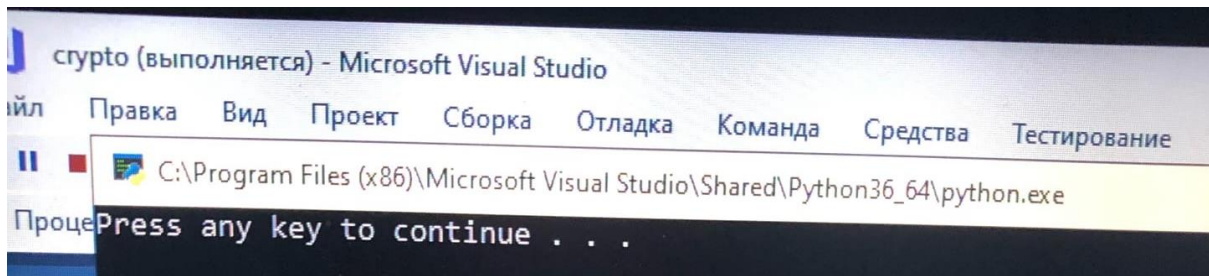
5. Тестирование

Извините за фото вместо скринов, у меня не работает клавиша принтскрин на ноутбуке, а скачивать Visual Studio на стационарный компьютер невозможно из-за отсутствия нормального интернета.

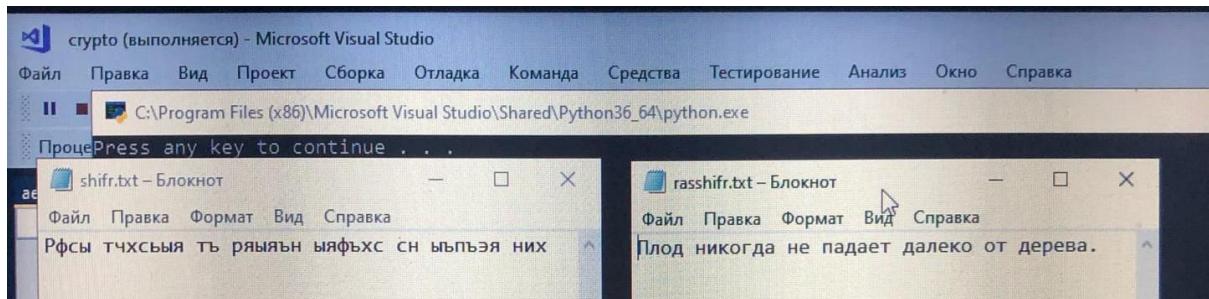
Перед началом работы программы в файл “opentext.txt” записываем исходный текст.



После выполнения программы окно выполнения программы выглядит так:

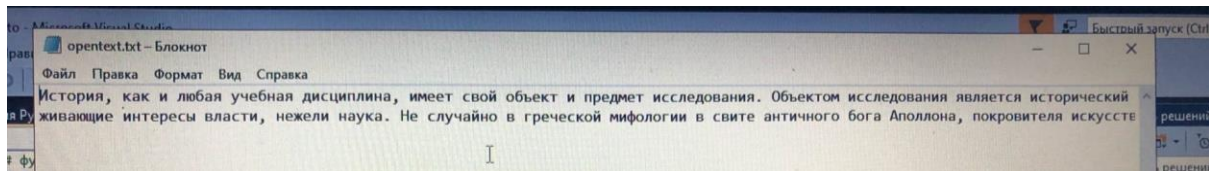


После выполнения программы в файл “shifr.txt” записывается зашифрованный текст, а в файл “rasshifr.txt” расшифрованный текст.

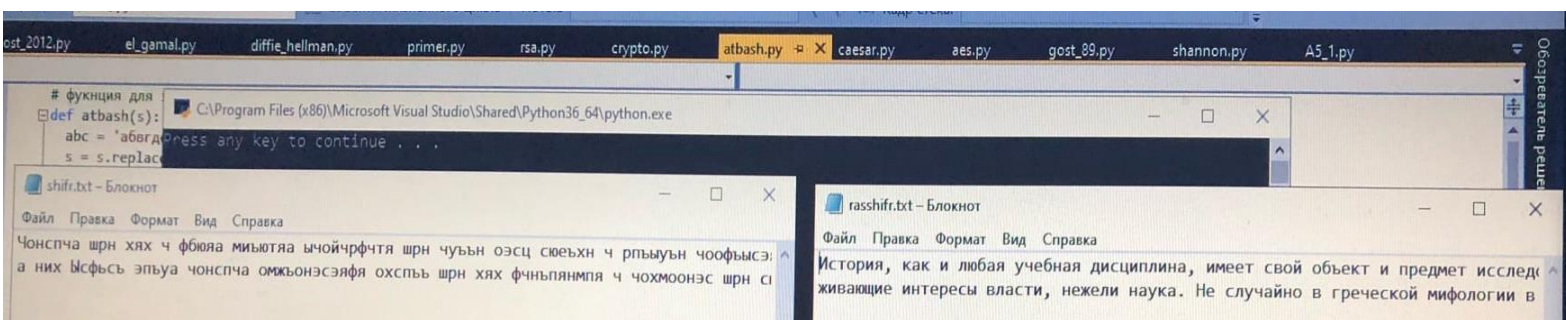


6. Работа с текстом не менее 1000 знаков

Перед началом работы программы в файл “opentext.txt” записываем исходный текст. (Полный исходный текст лежит в аннотации)



После выполнения программы в файл “shifr.txt” записывается зашифрованный текст, а в файл “rasshifr.txt” расшифрованный текст.



Полный зашифрованный текст:

Чонспча шрн хях ч фбюя миьютя ычойчрфчтя шрн чуьн оэсц сюехн ч рпыуьн
чоофысыятча них Сюехнсу чоофысыятча аэфаьноа чонспчиьохчч рпсйьоо нчпъ
рсофысыянфгтдц рпсйьоо пшэчнча рпчпсид ч сюжонэя шрн ипъыя оуьтабжчк
ыпмь ыпмья осюднчч шрн э хенспдк рпсаэфаьноа ыьаньфгтсонг утсьчк рсхсфьтчч
фбыьц них О хящыду уьтсэьтчуэпъуьтч шрн мзыьзчу э рпсзфсь чонспчиьохчч
рпсйьоо ысрсфтаьноа осэсхмртсонгб тсэдк осюднчч шрн аэфьтчч шрн ляхнсэ ч
ляхнспсэ э щштч иьфсэьхя шрн оьугч шрн внтсоя шрн ьсомыяпонэя шрн

$$Y_i = X_{i+3} \bmod n$$

X – исходный (открытый) текст

Y– зашифрованный текст

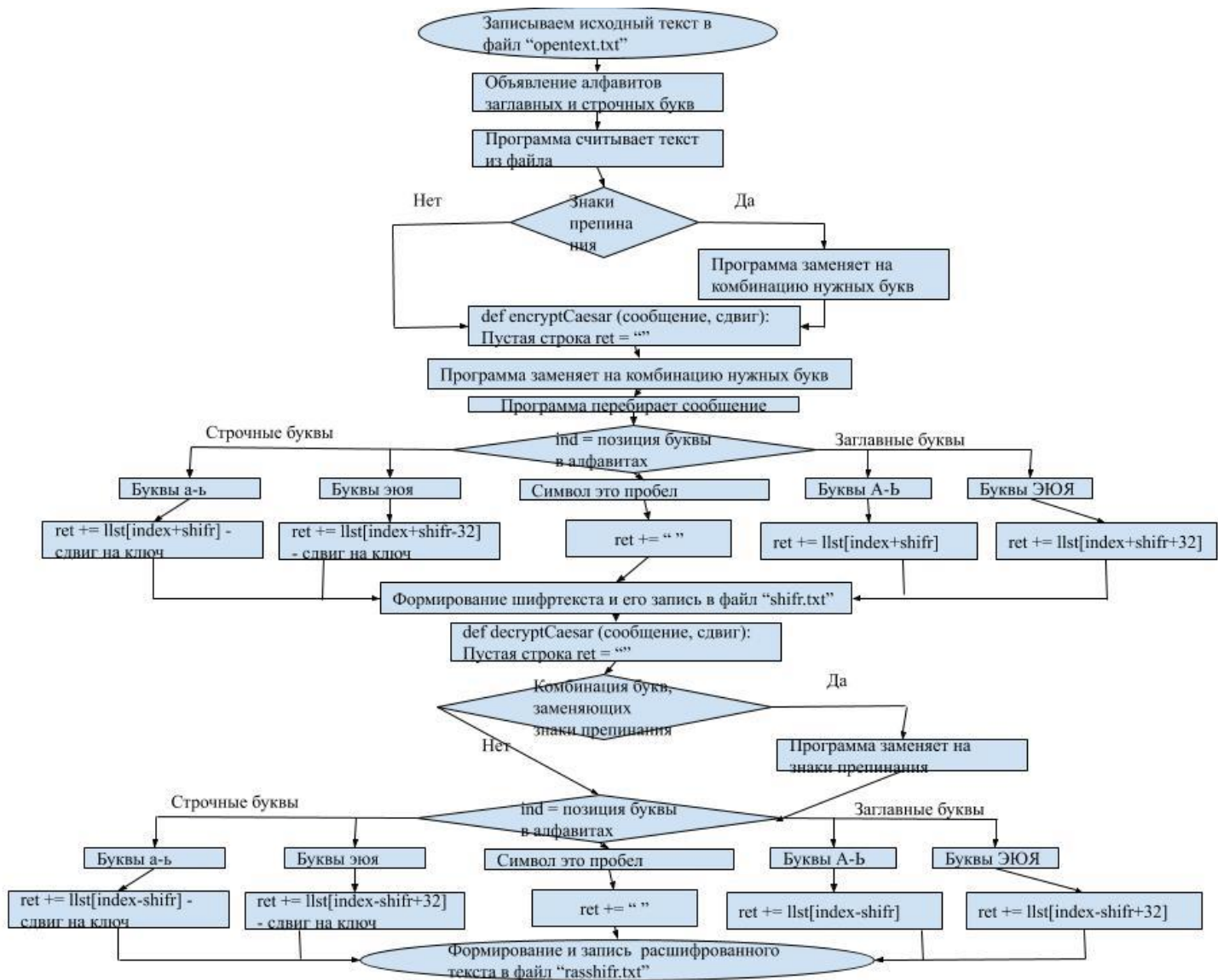
i – порядковый номер буквы открытого текста в алфавите, $i=(1 \dots n)$

n – количество букв в выбранном алфавите (мощность алфавита).

X	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
Y	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т
i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

X	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
Y	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В
i	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

3. Блок-схема программы



4. Код программы

```

llst =
['a','б','в','г','д','е','ж','з','и','й','к','л','м','н','о','п','р','с','т','у','ф','х','ц','ч','ш','щ','ъ','ы','ь','э','ю','я']
# задаем алфавит строчных букв
blst =
['А','Б','В','Г','Д','Е','Ж','З','И','Й','К','Л','М','Н','О','П','Р','С','Т','У','Ф','Х','Ц','Ч','Ш','Щ','Ъ','Ы','Ь','Э','Ю','Я']
# задаем алфавит заглавных букв
# функция для зашифрования
def encryptCaesar(msg, shift=3):
    msg = msg.replace('.', ' тчк') # Если в сообщении попадетсся точка, она заменется на тчк
    msg = msg.replace(',', ' зпт') # Если в сообщении попадетсся запятая, она заменется на зпт
    msg = msg.replace('-', ' тире') # Если в сообщении попадетсся - (тире), оно заменется на тире
    ret = "" # задаем пустую строку
    for x in msg: # цикл преобразует каждый элемент в тексте

```



```

    if x in llst and ord(x) < 1101: # если элемент строчная буква
        ind = llst.index(x) # переменная ind = позиция буквы в алфавите (а-0, б-1, ...)
        ret += llst[ind+shift] # в переменную ret записываем букву со сдвигом, при
# последующих выполнениях цикла к ret будут
    elif x in llst and ord(x) >= 1101: # если буквы эюя
        ind = llst.index(x) # переменная ind = позиция буквы в алфавите (а-0, б-1, ...)
        ret += llst[ind+shift-32]
        # приписываться следующие зашифрованные буквы
    elif x in blst and ord(x) < 1069: # если элемент заглавная буква
        ind = blst.index(x)
        ret += blst[ind+shift]
    elif x in blst and ord(x) >= 1069: # если буквы эюя
        ind = blst.index(x) # переменная ind = позиция буквы в алфавите (а-0, б-1, ...)
        ret += blst[ind+shift-32]
    else: # если элемент пробел
        ret += x
    return ret
# функция для расшифрования
def decryptCaesar(msg, shift=3):
    msg = msg.replace('хън', '.') # это сочетания получается при шифровании тчк
    msg = msg.replace('ктх', ',') # это сочетания получается при шифровании зпт
    msg = msg.replace('хлуи', '-') # это сочетания получается при шифровании тире
    ret = ""
    for x in msg:
        if x in llst and ord(x) >= 1075:
            ind = llst.index(x)
            ret += llst[ind-shift]
        elif x in llst and ord(x) < 1075: # если буквы эюя
            ind = llst.index(x) # переменная ind = позиция буквы в алфавите (а-1, б-2, ...)
            ret += llst[ind-shift+32]
        elif x in blst and ord(x) >= 1043:
            ind = blst.index(x)
            ret += blst[ind-shift]
        elif x in blst and ord(x) < 1043: # если буквы эюя
            ind = blst.index(x)
            ret += blst[ind-shift+32]
        else:
            ret += x
    return ret

# в этот файл нужно записать исходный текст
f = open(r"opentext.txt", "rt", encoding='utf-8')
text = f.read()

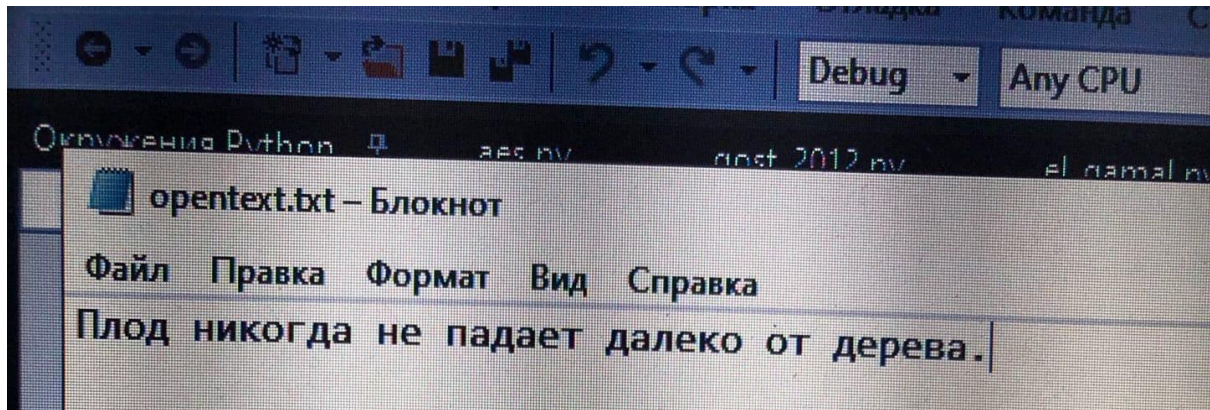
# в этот файл записывается зашифрованный текст
f = open('shifr.txt', 'wt', encoding='utf-8')
sh = encryptCaesar(str(text))
f.writelines(sh)
f.close()

```

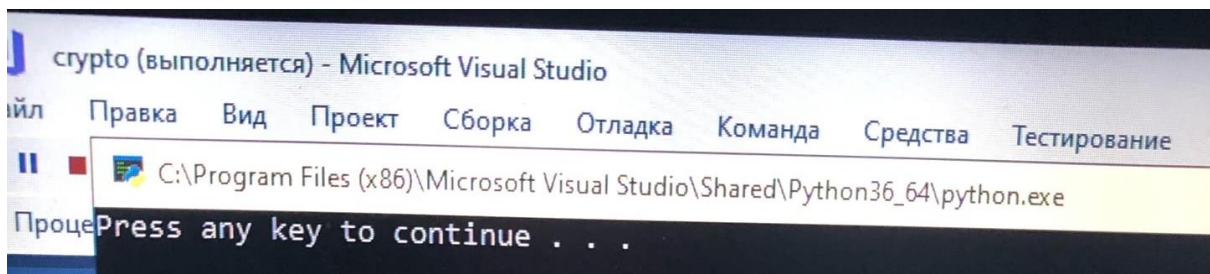
```
# в этот файл записывается расшифрованный текст
file = open('rasshifr.txt', 'wt', encoding='utf-8')
rassh = decryptCaesar(str(sh))
file.writelines(rassh)
file.close()
```

5. Тестирование

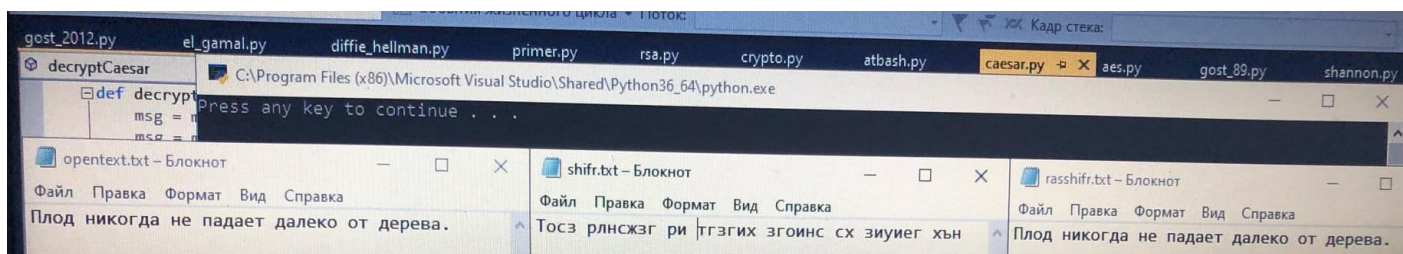
Перед началом работы программы в файл “opentext.txt” записываем исходный текст.



После выполнения программы окно выполнения программы выглядит так:

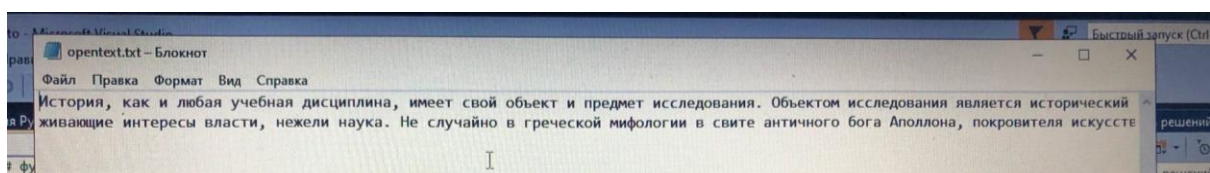


После выполнения программы в файл “shifr.txt” записывается зашифрованный текст, а в файл “rasshifr.txt” расшифрованный текст.

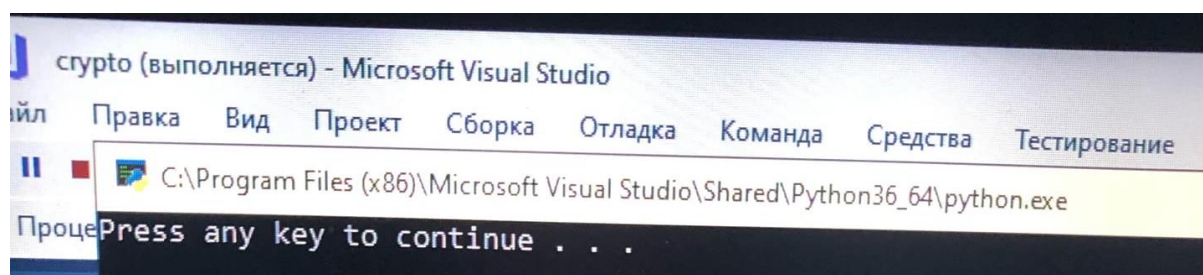


6. Работа с текстом не менее 1000 знаков

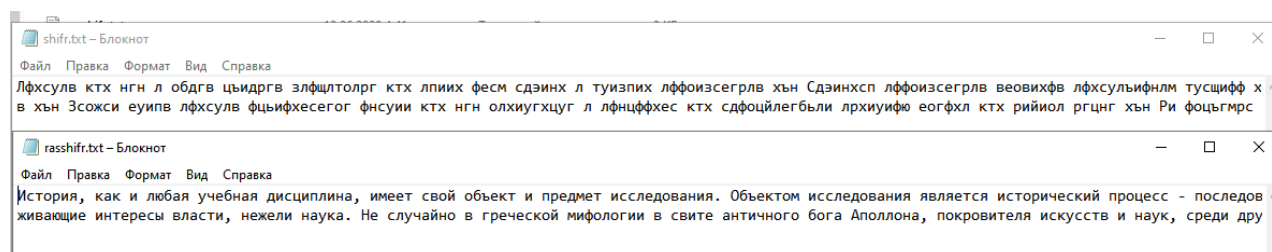
Перед началом работы программы в файл “opentext.txt” записываем исходный текст.
(Полный исходный текст лежит в аннотации)



После выполнения программы окно выполнения программы выглядит так:



После выполнения программы в файл “shifr.txt” записывается зашифрованный текст, а в файл “rasshifr.txt” расшифрованный текст.



Полный зашифрованный текст:

Лфхсулв ктх нгн л обдгв цыидргв злфщлтолрг ктх лпиих фесм сдэинх л туизпих
лффоизсегрлв хън Сдэинхсп лффоизсегрлв веовихфв лфхсулѳиѳнлм тусщифф хлуи
тсфоизсегхиоярюм тусщифф угкелхлв тулусзю л сдыифхег ктх ѳиуизг фпирвбьлш зуцж
зуцжг фсдюхлм ктх е нсхсуюш тусвеовихфв зивхиоярсфхя прсжлш тснсоирлм обзим
хън Ф нгѳюп пжрсеирлип еуипирл ктх цыизылп е тусыюси лфхсулѳиѳнлм тусщифф
зтсоровихфв фсеснцтрсфхяб рсеюш фсдюхлм ктх веоирлм ктх чгнхсе л чгнхсусе е
ѳлкрл ѳиосеинг ктх фиѳял ктх ахрсфг ктх жсфцзгугфхег ктх ѳиосеиѳифхег хън Тсзсдрс
хспц ктх нгн пгхиулгоярюм плу тиуййлегих тусщиффю рижархустлл л архустлл ктх
тулусзг еснущж ргф хлуи тусщиффю аесобщлл л лресобщлл ктх ѳиосеиѳифнси
сдыифхес шгугнхиулкцбх тусщиффю фсщлгоярсжс тусжуиффг л уижуиффг хън
Туизпихсп лкцѳирлв лфхсулл веовихфв зивхиоярсфхя жогерсжс ижс фцдэинхг хлуи
ѳиосеинг хлуи е тусыюсп хън Е цкнсп фпюѳои лфхсулв туизфхгеових фсдсм
фсеснцтрсфхя фсщлгоярюш чгнхсе ктх кргрлм ктх туизфхгеоирлм цѳирюш с хсп ктх
нгн ахл тусщиффю сфцѳифхеоволѳя л сфцѳифхеовбхфв хън Зсожси еуипв лфхсулв
фцѳифхесегог фнсуии ктх нгн олхиугхцуг л лфнцфхес ктх сдѳоѳйлегбьли лрхиуиѳю
еогфхл ктх рийиол ргцнг хън Ри фоцѳгмрс е жуиѳифнсм плчсосжлл е фелхи грхлърсжс
дежг Гтсоосрг ктх тснуселхиов лфнцфхе л ргцн ктх фуизл зуцжлш пцк дуюг Нолс
хлуи пцкг лфхсулл хън

Полный расшифрованный текст:

История, как и любая учебная дисциплина, имеет свой объект и предмет исследования. Объектом исследования является исторический процесс - последовательный процесс развития природы и общества, череда сменяющихся друг друга событий, в которых проявляется деятельность многих поколений людей. С каждым мгновением времени, ушедшим в прошлое исторический процесс дополняется совокупностью новых событий, явлений, фактов и факторов в жизни человека, семьи, этноса, государства, человечества. Подобно тому, как материальный мир переживает процессы негэнтропии

и энтропии, природа вокруг нас - процессы эволюции и инволюции, человеческое общество характеризуют процессы социального прогресса и регресса. Предметом изучения истории является деятельность главного его субъекта - человека - в прошлом. В узком смысле история представляет собой совокупность социальных фактов, знаний, представлений ученых о том, как эти процессы осуществлялись и осуществляются. Долгое время история существовала скорее, как литература и искусство, обслуживающие интересы власти, нежели наука. Не случайно в греческой мифологии в свите античного бога Аполлона, покровителя искусств и наук, среди других муз была Клио - муза истории.

7. Исполняемый файл

Вся работа происходит в файлах: “opentext.txt”, “shifr.txt”, “rasshifr.txt”.

Квадрат Полибия

1. Описание шифра.

Шифр замены, называемый Квадратом Полибия, был изобретен во 2 веке до нашей эры, он использовался для предупреждения об опасности с помощью двух факелов с охранных постов.

Текст алфавита помещается в таблицу, где каждой букве соответствуют два числа – номер строки i (количество факелов в левой руке) и номер столбца j (количество факелов в правой руке).

2. Алгоритм шифра.

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

Шифрование производится по формуле:

$$Y_{ij} = ij$$

Где:

Y – исходный (открытый) текст

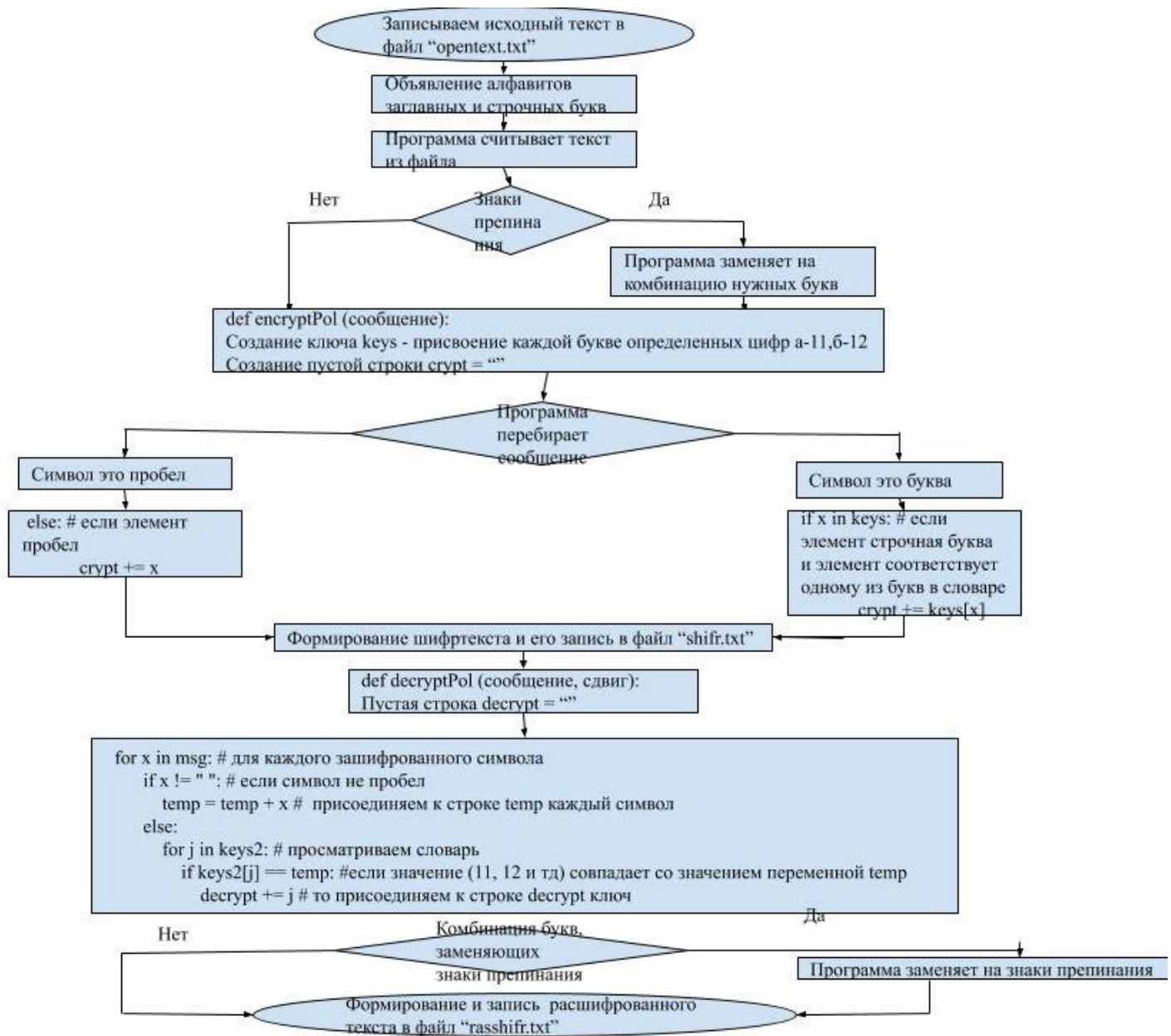
ij – зашифрованный текст

i – номер строки

j – номер столбца

Например, букве «И» соответствует число 23, где 2 – номер строки, 3 – номер столбца.

3. Блок-схема программы



4. Код программы

```

llst =
['a','б','в','г','д','е','ж','з','и','й','к','л','м','н','о','п','р','с','т','у','ф','х','ц','ч','ш','щ','б','ы','ь','э','ю','я']
# задаем алфавит строчных букв
blst =
['А','Б','В','Г','Д','Е','Ж','З','И','Й','К','Л','М','Н','О','П','Р','С','Т','У','Ф','Х','Ц','Ч','Ш','Щ','Ъ','Ы',
'Ь','Э','Ю','Я'] # задаем алфавит заглавных букв
# алфавиты
# Шифрование
def encryptPol(msg):
    msg = msg.replace('.', 'тчк') # Если в сообщении попадетсся точка, она заменется на тчк
    msg = msg.replace(',', 'зпт') # Если в сообщении попадетсся запятая, она заменется на
зпт
    msg = msg.replace('-', 'тире') # Если в сообщении попадетсся - (тире), оно заменется на
тире

```



```
msg = msg.replace(' ', 'прбл') # Если в сообщении попадетсся - (тире), оно заменется на
тире
```

```
keys = { # словарь (ключ: значение)
    'а':11, 'б':12, 'в':13, 'г':14, 'д':15, 'е':16,
    'ж':21, 'з':22, 'и':23, 'й':24, 'к':25, 'л':26,
    'м':31, 'н':32, 'о':33, 'п':34, 'р':35, 'с':36,
    'т':41, 'у':42, 'ф':43, 'х':44, 'ц':45, 'ч':46,
    'ш':51, 'щ':52, 'ъ':53, 'ы':54, 'ь':55, 'э':56,
    'ю':61, 'я':62,
}
keys2 = {
    'А':11, 'Б':12, 'В':13, 'Г':14, 'Д':15, 'Е':16,
    'Ж':21, 'З':22, 'И':23, 'Й':24, 'К':25, 'Л':26,
    'М':31, 'Н':32, 'О':33, 'П':34, 'Р':35, 'С':36,
    'Т':41, 'У':42, 'Ф':43, 'Х':44, 'Ц':45, 'Ч':46,
    'Ш':51, 'Щ':52, 'Ъ':53, 'Ы':54, 'Ь':55, 'Э':56,
    'Ю':61, 'Я':62,
}
crypt = "" # задаем строку
for x in msg:
    if x in keys and x in llst: # если элемент строчная буква и элемент соответствует
одному из букв в словаре
        crypt += keys[x] # к строке присоединятся значение его буквы (если буква А - 11
и тд)
        crypt += " " # к строке присоединятся пробел
    elif x in keys2 and x in blst: # если элемент заглавная буква
        crypt += keys2[x]
        crypt += " "
    else: # если элемент пробел
        crypt += x
crypt1 = crypt.replace('34 35 12 26', '')
# в этот файл записывается зашифрованный текст
f = open('shifr.txt', 'wt', encoding='utf-8')
f.writelines(crypt1)
f.close()
return crypt
```

```
# Расшифрование
```

```
def decryptPol(msg):
```

```
    keys = {
        'а':11, 'б':12, 'в':13, 'г':14, 'д':15, 'е':16,
        'ж':21, 'з':22, 'и':23, 'й':24, 'к':25, 'л':26,
        'м':31, 'н':32, 'о':33, 'п':34, 'р':35, 'с':36,
        'т':41, 'у':42, 'ф':43, 'х':44, 'ц':45, 'ч':46,
        'ш':51, 'щ':52, 'ъ':53, 'ы':54, 'ь':55, 'э':56,
        'ю':61, 'я':62,
    }
    keys2 = {
        'А':11, 'Б':12, 'В':13, 'Г':14, 'Д':15, 'Е':16,
        'Ж':21, 'З':22, 'И':23, 'Й':24, 'К':25, 'Л':26,
```

```

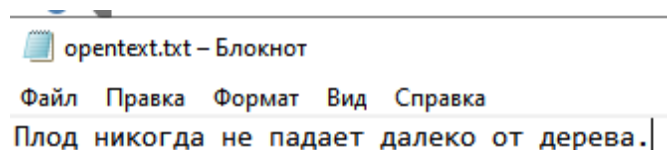
        'М':31, 'Н':32, 'О':33, 'П':34,      'Р':35, 'С':36,
        'Т':41, 'У':42, 'Ф':43, 'Х':44, 'Ц':45, 'Ч':46,
        'Ш':51, 'Щ':52, 'Ъ':53, 'Ы':54, 'Ь':55, 'Э':56,
        'Ю':61, 'Я':62,
    }
    temp = "" # задаем строку
    decrypt = "" # задаем строку
    for x in msg: # для каждого зашифрованного символа
        if x != " ": # если символ не пробел
            temp = temp + x # присоединяем к строке temp каждый символ
            #if x == " ": temp += " "
        else:
            for j in keys2: # просматриваем словарь
                if keys2[j] == temp: #если значение (11, 12 и тд) совпадает со значением
переменной temp
                    decrypt += j # то присоединяем к строке decrypt ключ
            temp = ""
        decrypt = decrypt.replace('ТЧК', '.') # Если в сообщении попадется точка, она
заменится на тчк
        decrypt = decrypt.replace('ЗПТ', ',') # Если в сообщении попадется запятая, она
заменится на зпт
        decrypt = decrypt.replace('ТИРЕ', '-') # Если в сообщении попадется - (тире), оно
заменится на тире
        decrypt = decrypt.replace('ПРБЛ', ' ') # Если в сообщении попадется - (тире), оно
заменится на тире
    return decrypt

# в этот файл нужно записать исходный текст
f = open(r"opentext.txt", "rt", encoding='utf-8')
text = f.read()
# в этот файл записывается зашифрованный текст
f = open('shifr.txt', 'wt', encoding='utf-8')
sh = encryptPol(str(text))
f.close()
# в этот файл записывается расшифрованный текст
file = open('rasshifr.txt', 'wt', encoding='utf-8')
rassh = decryptPol(str(sh))
file.writelines(rassh)
file.close()

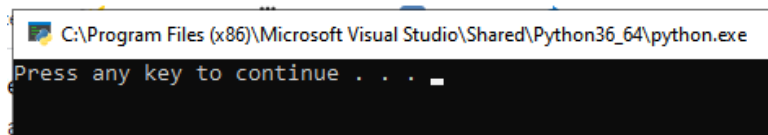
```

5. Тестирование

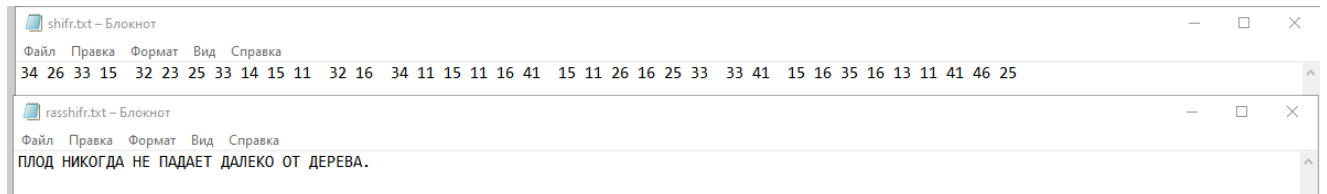
Перед началом работы программы в файл “opentext.txt” записываем исходный текст.



После выполнения программы окно выполнения программы выглядит так:

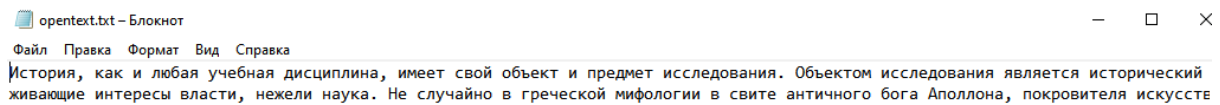


После выполнения программы в файл “shifr.txt” записывается зашифрованный текст, а в файл “rasshifr.txt” расшифрованный текст.

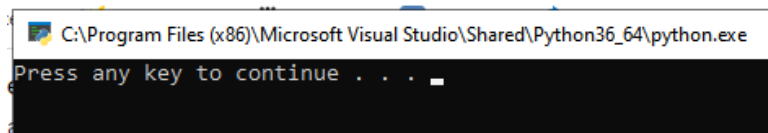


6. Работа с текстом не менее 1000 знаков

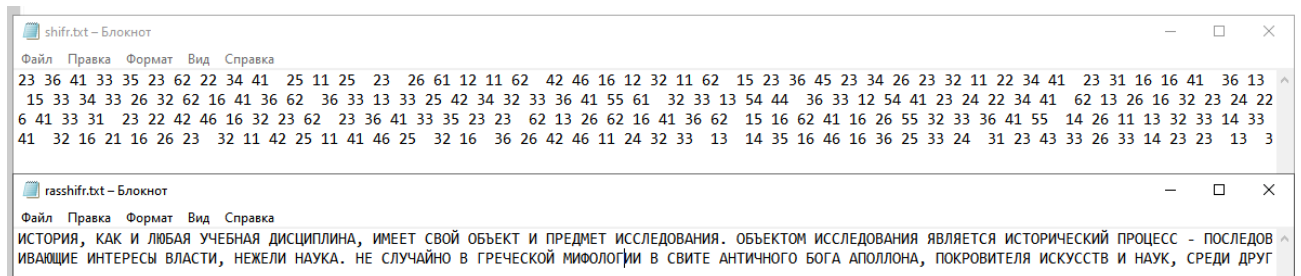
Перед началом работы программы в файл “opentext.txt” записываем исходный текст. (Полный исходный текст лежит в аннотации)



После выполнения программы окно выполнения программы выглядит так:



После выполнения программы в файл “shifr.txt” записывается зашифрованный текст, а в файл “rasshifr.txt” расшифрованный текст.



Полный зашифрованный текст:

23 36 41 33 35 23 62 22 34 41 25 11 25 23 26 61 12 11 62 42 46 16 12 32 11 62 15 23 36
 45 23 34 26 23 32 11 22 34 41 23 31 16 16 41 36 13 33 24 33 12 53 16 25 41 23 34 35 16
 15 31 16 41 23 36 36 26 16 15 33 13 11 32 23 62 41 46 25 33 12 53 16 25 41 33 31 23 36
 36 26 16 15 33 13 11 32 23 62 62 13 26 62 16 41 36 62 23 36 41 33 35 23 46 16 36 25 23
 24 34 35 33 45 16 36 36 41 23 35 16 34 33 36 26 16 15 33 13 11 41 16 26 55 32 54 24 34
 35 33 45 16 36 36 35 11 22 13 23 41 23 62 34 35 23 35 33 15 54 23 33 12 52 16 36 41 13
 11 22 34 41 46 16 35 16 15 11 36 31 16 32 62 61 52 23 44 15 35 42 14 15 35 42 14 11 36
 33 12 54 41 23 24 22 34 41 13 25 33 41 33 35 54 44 34 35 33 62 13 26 62 16 41 36 62 15
 16 62 41 16 26 55 32 33 36 41 55 31 32 33 14 23 44 34 33 25 33 26 16 32 23 24 26 61 15
 16 24 41 46 25 36 25 11 21 15 54 31 31 14 32 33 13 16 32 23 16 31 13 35 16 31 16 32 23

22 34 41 42 51 16 15 51 23 31 13 34 35 33 51 26 33 16 23 36 41 33 35 23 46 16 36 25 23
24 34 35 33 45 16 36 36 15 33 34 33 26 32 62 16 41 36 62 36 33 13 33 25 42 34 32 33 36
41 55 61 32 33 13 54 44 36 33 12 54 41 23 24 22 34 41 62 13 26 16 32 23 24 22 34 41 43
11 25 41 33 13 23 43 11 25 41 33 35 33 13 13 21 23 22 32 23 46 16 26 33 13 16 25 11 22
34 41 36 16 31 55 23 22 34 41 56 41 32 33 36 11 22 34 41 14 33 36 42 15 11 35 36 41 13
11 22 34 41 46 16 26 33 13 16 46 16 36 41 13 11 41 46 25 34 33 15 33 12 32 33 41 33 31
42 22 34 41 25 11 25 31 11 41 16 35 23 11 26 55 32 54 24 31 23 35 34 16 35 16 21 23 13
11 16 41 34 35 33 45 16 36 36 54 32 16 14 56 32 41 35 33 34 23 23 23 56 32 41 35 33 34
23 23 22 34 41 34 35 23 35 33 15 11 13 33 25 35 42 14 32 11 36 41 23 35 16 34 35 33 45
16 36 36 54 56 13 33 26 61 45 23 23 23 23 32 13 33 26 61 45 23 23 22 34 41 46 16 26 33
13 16 46 16 36 25 33 16 33 12 52 16 36 41 13 33 44 11 35 11 25 41 16 35 23 22 42 61 41
34 35 33 45 16 36 36 54 36 33 45 23 11 26 55 32 33 14 33 34 35 33 14 35 16 36 36 11 23
35 16 14 35 16 36 36 11 41 46 25 34 35 16 15 31 16 41 33 31 23 22 42 46 16 32 23 62 23
36 41 33 35 23 23 62 13 26 62 16 41 36 62 15 16 62 41 16 26 55 32 33 36 41 55 14 26 11
13 32 33 14 33 16 14 33 36 42 12 53 16 25 41 11 41 23 35 16 46 16 26 33 13 16 25 11 41
23 35 16 13 34 35 33 51 26 33 31 41 46 25 13 42 22 25 33 31 36 31 54 36 26 16 23 36 41
33 35 23 62 34 35 16 15 36 41 11 13 26 62 16 41 36 33 12 33 24 36 33 13 33 25 42 34 32
33 36 41 55 36 33 45 23 11 26 55 32 54 44 43 11 25 41 33 13 22 34 41 22 32 11 32 23 24
22 34 41 34 35 16 15 36 41 11 13 26 16 32 23 24 42 46 16 32 54 44 33 41 33 31 22 34 41
25 11 25 56 41 23 34 35 33 45 16 36 36 54 33 36 42 52 16 36 41 13 26 62 26 23 36 55 23
33 36 42 52 16 36 41 13 26 62 61 41 36 62 41 46 25 15 33 26 14 33 16 13 35 16 31 62 23
36 41 33 35 23 62 36 42 52 16 36 41 13 33 13 11 26 11 36 25 33 35 16 16 22 34 41 25 11
25 26 23 41 16 35 11 41 42 35 11 23 23 36 25 42 36 36 41 13 33 22 34 41 33 12 36 26 42
21 23 13 11 61 52 23 16 23 32 41 16 35 16 36 54 13 26 11 36 41 23 22 34 41 32 16 21 16
26 23 32 11 42 25 11 41 46 25 32 16 36 26 42 46 11 24 32 33 13 14 35 16 46 16 36 25 33
24 31 23 43 33 26 33 14 23 23 13 36 13 23 41 16 11 32 41 23 46 32 33 14 33 12 33 14 11
11 34 33 26 26 33 32 11 22 34 41 34 33 25 35 33 13 23 41 16 26 62 23 36 25 42 36 36 41
13 23 32 11 42 25 22 34 41 36 35 16 15 23 15 35 42 14 23 44 31 42 22 12 54 26 11 25
26 23 33 41 23 35 16 31 42 22 11 23 36 41 33 35 23 23 41 46 25

Полный расшифрованный текст:

ИСТОРИЯ, КАК И ЛЮБАЯ УЧЕБНАЯ ДИСЦИПЛИНА, ИМЕЕТ СВОЙ ОБЪЕКТ И ПРЕДМЕТ ИССЛЕДОВАНИЯ. ОБЪЕКТОМ ИССЛЕДОВАНИЯ ЯВЛЯЕТСЯ ИСТОРИЧЕСКИЙ ПРОЦЕСС - ПОСЛЕДОВАТЕЛЬНЫЙ ПРОЦЕСС РАЗВИТИЯ ПРИРОДЫ И ОБЩЕСТВА, ЧЕРЕДА СМЕНЯЮЩИХ ДРУГ ДРУГА СОБЫТИЙ, В КОТОРЫХ ПРОЯВЛЯЕТСЯ ДЕЯТЕЛЬНОСТЬ МНОГИХ ПОКОЛЕНИЙ ЛЮДЕЙ. С КАЖДЫМ МГНОВЕНИЕМ ВРЕМЕНИ, УШЕДШИМ В ПРОШЛОЕ ИСТОРИЧЕСКИЙ ПРОЦЕСС ДОПОЛНЯЕТСЯ СОВОКУПНОСТЬЮ НОВЫХ СОБЫТИЙ, ЯВЛЕНИЙ, ФАКТОВ И ФАКТОРОВ В ЖИЗНИ ЧЕЛОВЕКА, СЕМЬИ, ЭТНОСА, ГОСУДАРСТВА, ЧЕЛОВЕЧЕСТВА. ПОДОБНО ТОМУ, КАК МАТЕРИАЛЬНЫЙ МИР ПЕРЕЖИВАЕТ ПРОЦЕССЫ НЕГЭНТРОПИИ И ЭНТРОПИИ, ПРИРОДА ВОКРУГ НАС - ПРОЦЕССЫ ЭВОЛЮЦИИ И ИНВОЛЮЦИИ, ЧЕЛОВЕЧЕСКОЕ ОБЩЕСТВО ХАРАКТЕРИЗУЮТ ПРОЦЕССЫ СОЦИАЛЬНОГО ПРОГРЕССА И РЕГРЕССА. ПРЕДМЕТОМ ИЗУЧЕНИЯ ИСТОРИИ ЯВЛЯЕТСЯ ДЕЯТЕЛЬНОСТЬ ГЛАВНОГО ЕГО СУБЪЕКТА - ЧЕЛОВЕКА - В ПРОШЛОМ. В УЗКОМ СМЫСЛЕ ИСТОРИЯ ПРЕДСТАВЛЯЕТ СОБОЙ СОВОКУПНОСТЬ СОЦИАЛЬНЫХ ФАКТОВ, ЗНАНИЙ, ПРЕДСТАВЛЕНИЙ

УЧЕНЫХ О ТОМ, КАК ЭТИ ПРОЦЕССЫ ОСУЩЕСТВЛЯЛИСЬ И ОСУЩЕСТВЛЯЮТСЯ. ДОЛГОЕ ВРЕМЯ ИСТОРИЯ СУЩЕСТВОВАЛА СКОРЕЕ, КАК ЛИТЕРАТУРА И ИСКУССТВО, ОБСЛУЖИВАЮЩИЕ ИНТЕРЕСЫ ВЛАСТИ, НЕЖЕЛИ НАУКА. НЕ СЛУЧАЙНО В ГРЕЧЕСКОЙ МИФОЛОГИИ В СВИТЕ АНТИЧНОГО БОГА АПОЛЛОНА, ПОКРОВИТЕЛЯ ИСКУССТВ И НАУК, СРЕДИ ДРУГИХ МУЗ БЫЛА КЛИО - МУЗА ИСТОРИИ.

7. Исполняемый файл

Вся работа происходит в файлах: “opentext.txt”, “shifr.txt”, “rasshifr.txt”.