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 В: Шифры многозначной замены

Выполнила студентка 3 курса группы 171-341

Решетникова Дарья

Москва 2020 г.

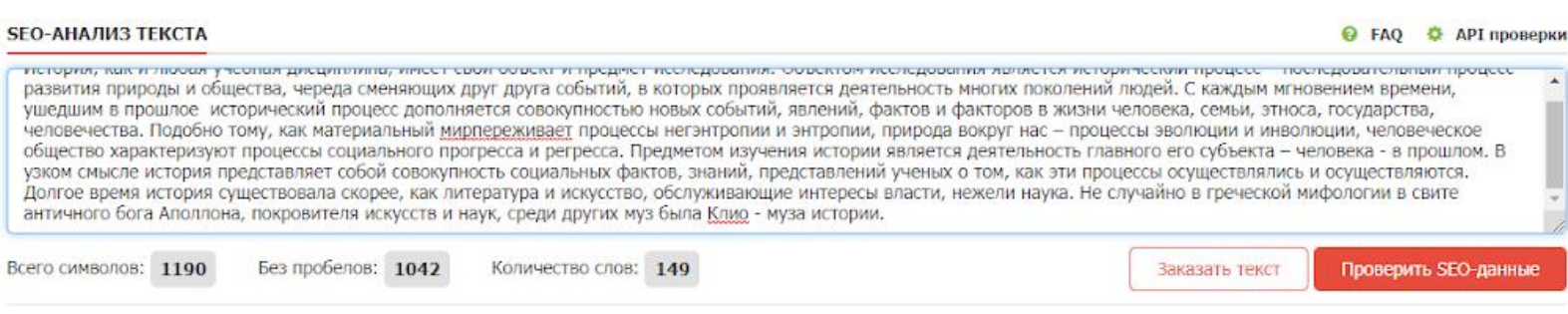
Аннотация

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.



Шифр Тритемия.

1. Описание шифра.

Шифр Тритемиуса — система шифрования, разработанная Иоганном Тритемием. Представляет собой усовершенствованный шифр Цезаря, то есть шифр подстановки. По алгоритму шифрования, каждый символ сообщения смещается на символ, отстающий от данного на некоторый шаг. Здесь шаг смещения делается переменным, то есть зависящим от каких-либо дополнительных факторов.

2. Алгоритм шифра.

$$Y_j = X_{i+j-1} \bmod n$$

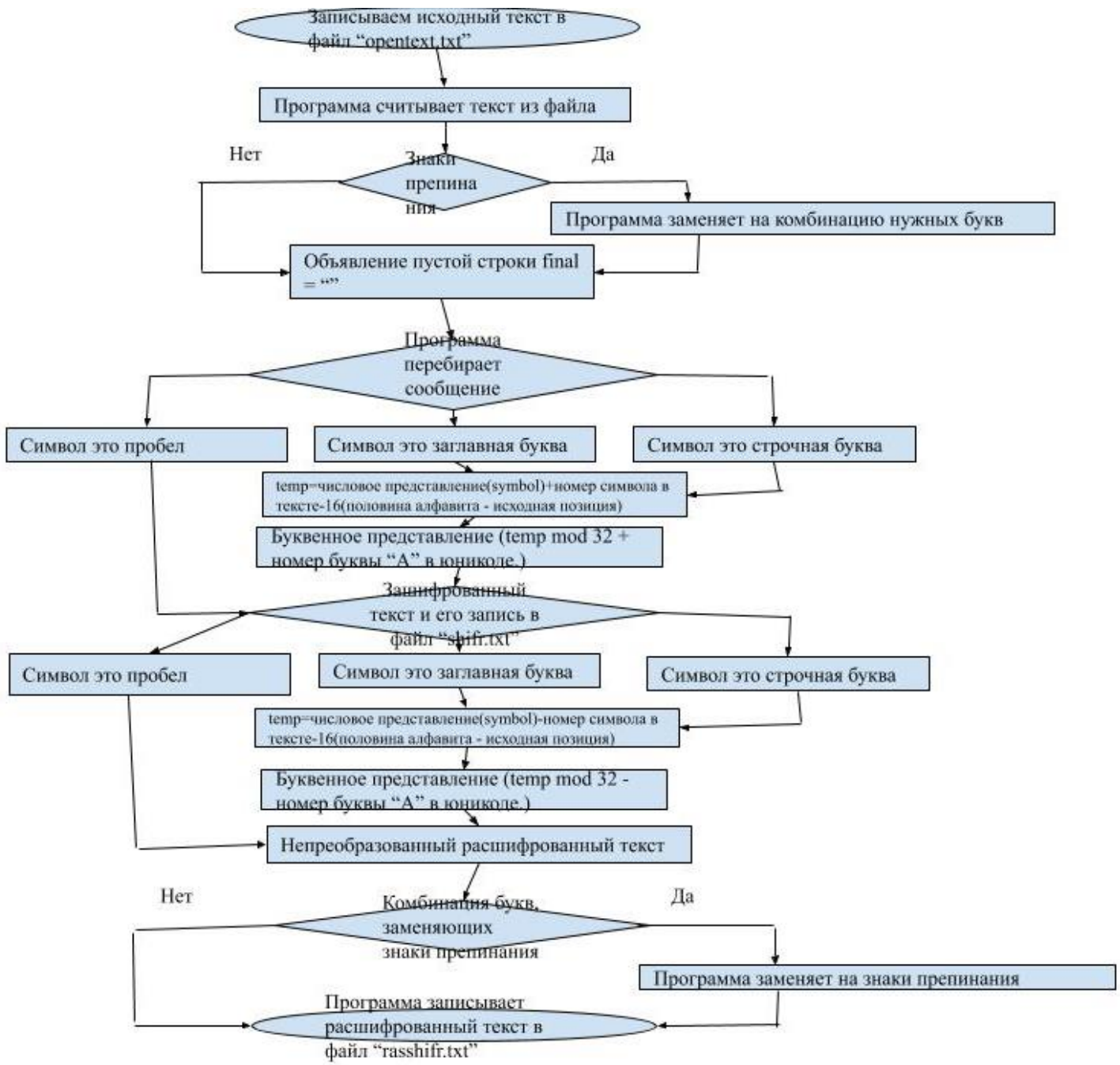
X – исходный (открытый) текст
Y– зашифрованный текст
i – порядковый номер буквы в алфавите таблицы, i=1…n
j – порядковый номер буквы в тексте, j=1…k
k – количество букв в тексте
n – количество букв в выбранном алфавите (мощность алфавита).

ТАБЛИЦА ТРИТЕМИЯ

A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	X	Y	Z	W
B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	X	Y	Z	W	A
C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	X	Y	Z	W	A	B
D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	X	Y	Z	W	A	B	C
E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	X	Y	Z	W	A	B	C	D
F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	X	Y	Z	W	A	B	C	D	E
G	H	I	K	L	M	N	O	P	Q	R	S	T	U	X	Y	Z	W	A	B	C	D	E	F
H	I	K	L	M	N	O	P	Q	R	S	T	U	X	Y	Z	W	A	B	C	D	E	F	G
I	K	L	M	N	O	P	Q	R	S	T	U	X	Y	Z	W	A	B	C	D	E	F	G	H
K	L	M	N	O	P	Q	R	S	T	U	X	Y	Z	W	A	B	C	D	E	F	G	H	I
L	M	N	O	P	Q	R	S	T	U	X	Y	Z	W	A	B	C	D	E	F	G	H	I	K
M	N	O	P	Q	R	S	T	U	X	Y	Z	W	A	B	C	D	E	F	G	H	I	K	L
N	O	P	Q	R	S	T	U	X	Y	Z	W	A	B	C	D	E	F	G	H	I	K	L	M
O	P	Q	R	S	T	U	X	Y	Z	W	A	B	C	D	E	F	G	H	I	K	L	M	N
P	Q	R	S	T	U	X	Y	Z	W	A	B	C	D	E	F	G	H	I	K	L	M	N	O
Q	R	S	T	U	X	Y	Z	W	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P
R	S	T	U	X	Y	Z	W	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q
S	T	U	X	Y	Z	W	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R
T	U	X	Y	Z	W	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S
U	X	Y	Z	W	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T
X	Y	Z	W	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U
Y	Z	W	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	X
Z	W	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	X	Y
W	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	X	Y	Z

Это первый **многоалфавитный периодический шиф**

3. Блок-схема программы



4. Код программы

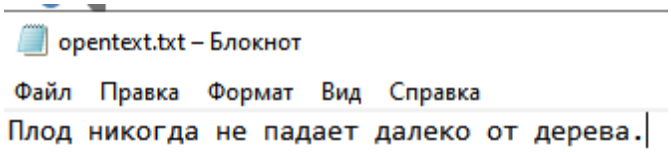
```
#Функция шифрования
def encryptTritem(message):
    message = message.replace('.', ' тчк') # Если в сообщении попадется точка, она заменется на тчк
    message = message.replace(',', ' зпт') # Если в сообщении попадется запятая, она заменется на зпт
    message = message.replace('-', 'тире') # Если в сообщении попадется - (тире), оно заменется на тире
    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
```

```
    final = "" # задаем строку
    index = -1 # индекс -1, тк при первом шаге мы прибавляем 1 в любом случае
    for symbol in message: # перебираем каждый символ сообщения
        index = index+1 # прибавляем 1 при каждом шаге
        if symbol == " ": # если символ - пробел (такие же условия будут при ! ? ?! / и тд)
            index = index-1 # не считаем его как индекс, чтобы последующие символы не сдвигались
            final += " " # прибавляем пробел к зашифрованному сообщению
        elif symbol in blst: # если символ заглавная буква
            temp = ord(symbol) + index - 16 # присваеваем переменной temp=числовое представление(symbol)+номер символа в тексте-16(половина алфавита - исходная позиция)
            final += chr(temp%32 + ord('A')) # возвращаем символ, который соответствует числовому представлению (делим переменную temp по модулю 32 + числовое значение первого символа алфавита заглавных букв) (% - деление по модулю - остаток от деления)
        else:
            temp = ord(symbol) + index - 16
            final += chr(temp%32 + ord('a'))
    return final

#Функция расшифрования
def decryptTritem(message):
    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
    final = ""
    index = -1
    for symbol in message:
        index = index+1
        if symbol == " ":
            index = index-1
            final += " "
        elif symbol in blst:
            temp = ord(symbol) - index - 16 # вычитаем позицию числа в тексте
            final += chr(temp%32 + ord('A'))
        else:
            temp = ord(symbol) - index - 16
            final += chr(temp%32 + ord('a'))
    final = final.replace(' тчк', '.') # Если в сообщении попадется точка, она заменется на тчк
    final = final.replace(' зпт', ',') # Если в сообщении попадется запятая, она заменется на зпт
```

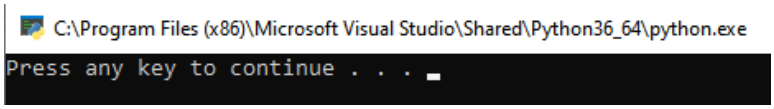
```
    final = final.replace('тире', '-') # Если в сообщении попадется - (тире), оно заменется на тире
    return final

# 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 = encryptTritem(str(text))
f.writelines(sh)
f.close()

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

5. Тестирование

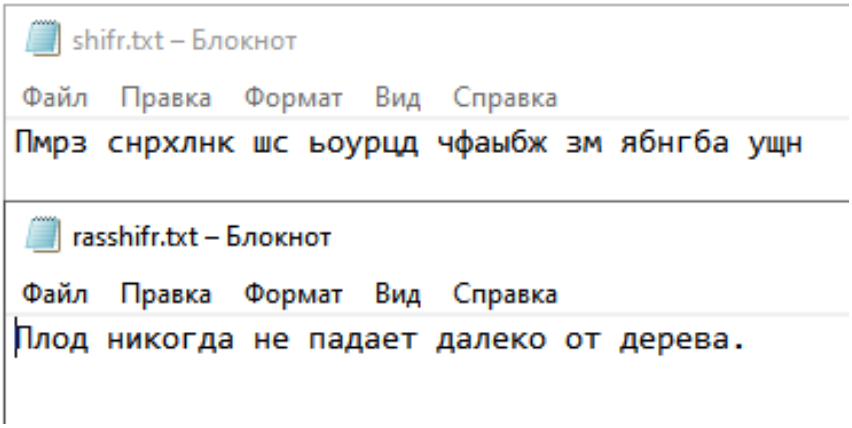
Перед началом работы программы в файл “opentext.txt” записываем исходный текст.



Так выглядит окно выполнения программы.

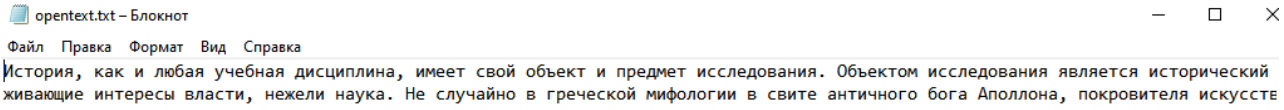


После выполнения программы в файл “shifr.txt” записывается зашифрованный текст, а в файл “rasshifr.txt” расшифрованный текст.

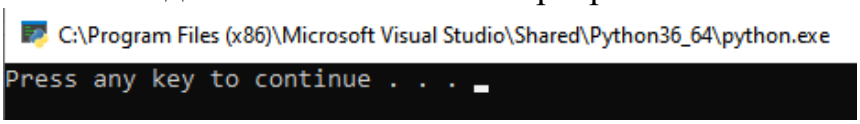


6. Работа с текстом не менее 1000 знаков

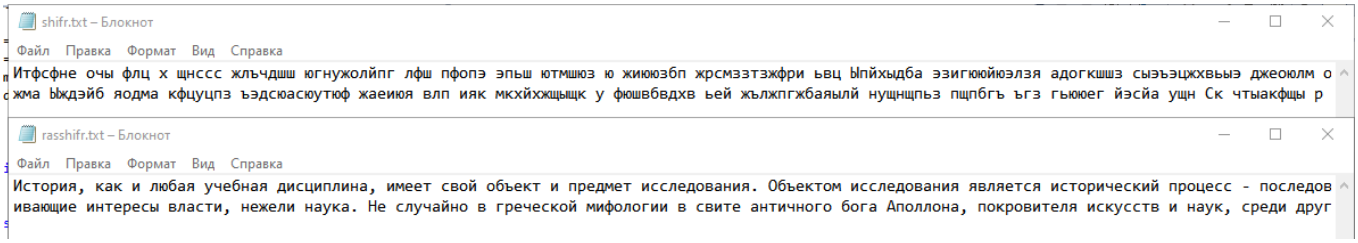
Перед началом работы программы в файл “opentext.txt” записываем исходный текст. (Полный исходный текст лежит в аннотации)



Так выглядит окно выполнения программы.



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



Полный зашифрованный текст:

Итфсфне очы флц х щнссс жлъчдшш югнужолйпг лфш пфопэ эпьш ютмшюз ю жиююзбп жрсмззтзжфри ьвц Ыпйхыдба эзигююйюэлзя адогкшшз сыэъэцжхвьыэ джеоюлм оеод ппуоййфйиыпциъйш ябайщжз зшаьгоеэ орйтсиа о хйвпьюпо цяг йшдъъч йеяиыычзх етцз йцълй ыщниачщ шбе ц ядйжйхр лнмювмбицце лниьрчйыэбго ябгщян ииекигмик нбикп щяу Ы хмуткь эхавчыдаюж эмвкднй йтц шюммбтч о ьюэиьаш ьжиеибсанзжи псрщйцч лцшшщщмуббр гбцгакзжимощь могэш хузвъсу тыя нсыцяыэ ьей мщднкя ж уалфсфуи й оссшф дуъюучэф ьей йюжчд днс эупсхе нцъ мшьясоябгфу ыди оэдиэбфгртгв хып Ххмчлшъ яьыг шбе юха гшлялдэйыньл пмх хмшоруонуб ябайщжзт еюэшйпонпйк л бтшчцшту уьа юащвбшх шевйню йэп сисз тфуьмщъе иоыщнжщъ ы ьвшегчргд днс чжнсжкэмщушр

ъозфбгфб йхжчвлялддсэт ртсъкчшг ъшбфнщлэяхб гедъиюлмь е одгсзфхе шют Шъррщубюэ ъъзмыдаш вмолози адогкшшз нпкютщлэягер шбчъжиюк вбн сфгэйпшз ъсър гтщэтцьу жэжь ъ икйфимл тшм Е чмрхф ъцжэшу чбгагьф езээлньяйюеу усеуп шцлшхяьыэбго двлючгхзцс сюйтпд куч нфицтф уьа юаццджхшвэжвд пфгмыц р хтс нцъ укх йяц юаяишежс еймуанпакямкфа н фшывпьюпщооггт жма Ыждэйб яодма кфцуцпз ъэдсюасюутюф жаеиюя влп ияк мкхйхжщыщк у фюшвбвдхв ьей жължпгжбаяылй нущнщпьз пщпбгъ ъгз гьююег йэсйа ущн Ск чтыакфщы р тацйшеяда дбойзлбзи г уемчл зхытвщысэ сяху Фддвгззы гмр оолтсжншмуи тьцаяаву ъ афиа юзл ллббж грфелщ сщо йдхл Цшцэ вщвш аиэч акмймеж счл

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”.

Шифр Белазо.

1. Описание шифра.

Джованни Батиста Белазо в 1553 году (брошюра «Шифр синьора Белазо») предложил использовать для многоалфавитного шифра буквенный, легко запоминаемый ключ, который он назвал паролем.

2. Алгоритм шифра.

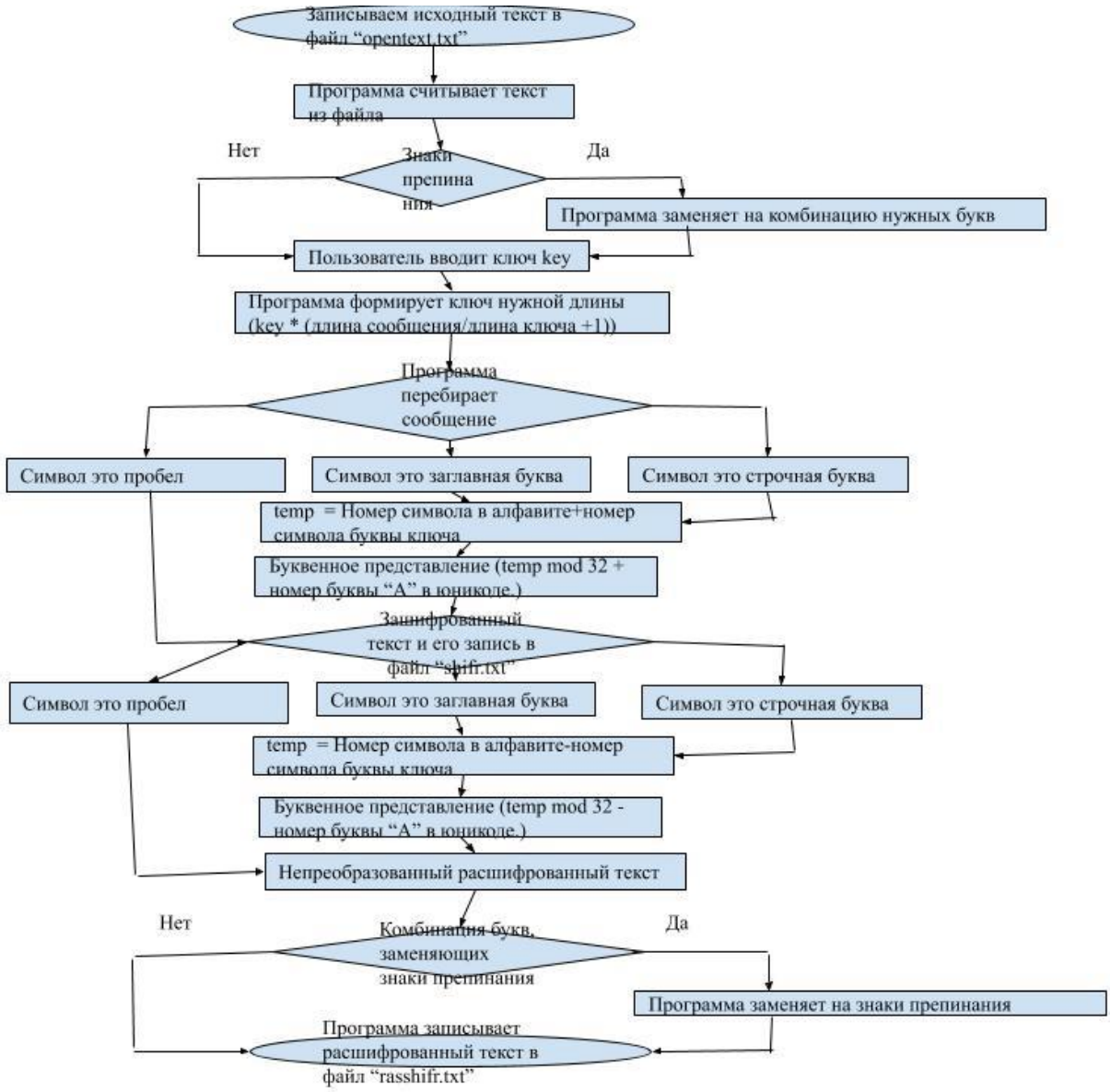
Шифрование осуществляется с помощью пароля-ключа, состоящего из M символов. Из полной таблицы Тритемия выделяется матрица $T_Ш$ размерностью [(M+1) x R]. Она включает первую строку и строки, первые элементы которых совпадают с символами ключа. Если в качестве ключа выбрано слово <ЗОНД>, то матрица шифрования содержит пять строк :

	А	Б	В	Г	Д	Е	Ж	З	И	К	.	.	.	Э	Ю	Я	␣
	З	И	К	Л	М	Н	О	П	Р	С	.	.	.	Г	Д	Е	Ж
$T_В$=	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	.	.	.	К	Л	М	Н
	Н	О	П	Р	С	Т	У	Ф	Х	Ц	.	.	.	И	К	Л	М
	Д	Е	Ж	З	И	К	Л	М	Н	О	.	.	.	А	Б	В	Г

Замена с использованием шифра Белазо эквивалентна простой замене с циклическим изменением алфавита. При этом в каждом цикле имеем многоалфавитную подстановку с числом используемых алфавитов, соответствующим числу букв в слове ключа.

При шифровании необходимо вначале записать под буквами шифруемого текста буквы ключевого слова. Ключ при этом повторяется необходимое число раз. Символ шифруемого текста определяет столбец матрицы шифрования. Необходимый для его замены символ находится на пересечении этого столбца со строкой, соответствующей букве ключа, записанного под шифруемым текстом.

3. Блок-схема программы



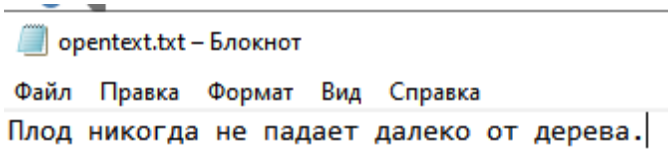
4. Код программы

```
#Функция шифрования
def encryptBelaso(message):
    message = message.replace('.', ' тчк') # Если в сообщении попадется точка, она заменется на тчк
    message = message.replace(',', ' зпт') # Если в сообщении попадется запятая, она заменется на зпт
    message = message.replace('-', 'тире') # Если в сообщении попадется - (тире), оно заменется на тире
    key = input("Write the key: ").upper()
    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
    final = "" # задаем строку
    index = -1 # индекс -1, тк при первом шаге мы прибавляем 1 в любом случае
    key = key * (len(message) // len(key) + 1) # умножаем переменную key на (остаток от деления длины текста на длину ключа + 1)
    #(+1 нужно чтобы ключ в любом случае был чуть больше текста, чтобы не было выхода за пределы строки)
    for symbol in message: # перебираем каждый символ сообщения
        index = index+1 # прибавляем 1 при каждом шаге
        if symbol == " ": # если символ - пробел (такие же условия будут при ! ? ?! / и тд)
            index = index-1 # не считаем его как индекс, чтобы последующие символы не сдвигались
            final += " " # прибавляем пробел к зашифрованному сообщению
        elif symbol in blst: # если символ заглавная буква
            temp = ord(symbol) + ord(key[index]) # temp = числовое представление (symbol) + числовое представление (индекса ключа)
            final += chr(temp%32 + ord('А')) # возвращаем символ, который соответствует числовому представлению (делим переменную temp
            # по модулю 32 + числовое значение первого символа алфавита заглавных букв) (% - деление по модулю - остаток от деления)
        else:
            temp = ord(symbol) + ord(key[index])
            final += chr(temp%32 + ord('а'))
    return final
# сюхущьрфзвхг
#Функция расшифрования
def decryptBelaso(message):
    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
```

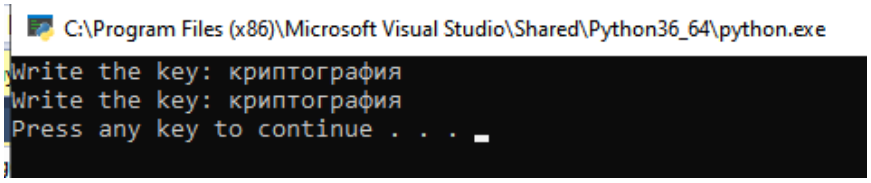
```
    final = ""
    index = -1
    key = input("Write the key: ").upper()
    key *= len(message) // len(key) + 1
    for symbol in message:
        index = index+1
        if symbol == " ":
            index = index-1
            final += " "
        elif symbol in blst:
            temp = ord(symbol) - ord(key[index])  # вычитаем позицию числа в тексте
            final += chr(temp%32 + ord('A'))
        else:
            temp = ord(symbol) - ord(key[index])
            final += chr(temp%32 + ord('a'))
    final = final.replace(' тчк', '.') # Если в сообщении попадется точка, она заменется на тчк
    final = final.replace(' зпт', ',') # Если в сообщении попадется запятая, она заменется на зпт
    final = final.replace('тире', '-') # Если в сообщении попадется - (тире), оно заменется на тире
    return final

# 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 = encryptBelaso(str(text))
f.writelines(sh)
f.close()

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

5. Тестирование

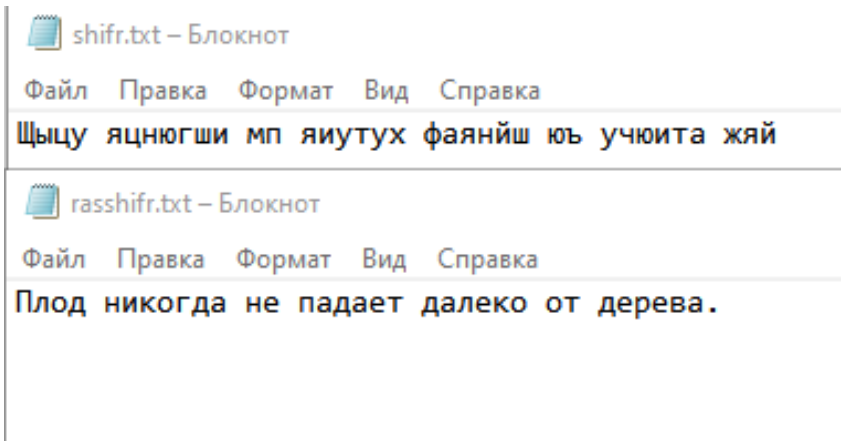
Перед началом работы программы в файл “opentext.txt” записываем исходный текст.



Так выглядит окно выполнения программы. Мы вводим ключ дважды – ключ шифрования и ключ расшифрования соответственно.

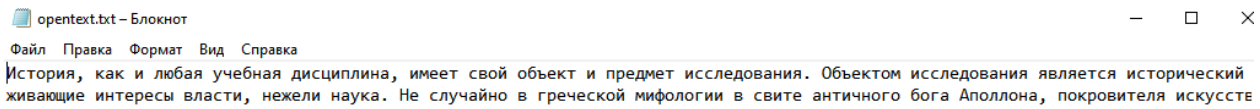


После выполнения программы в файл “shifr.txt” записывается зашифрованный текст, а в файл “rasshifr.txt” расшифрованный текст.

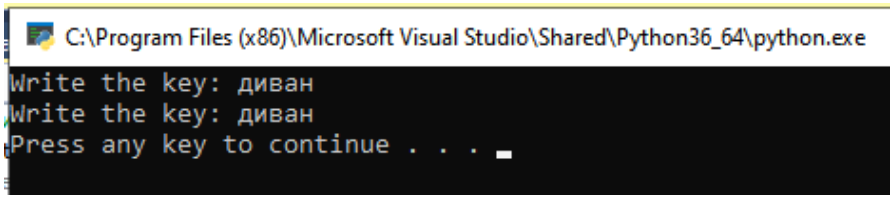


6. Работа с текстом не менее 1000 знаков

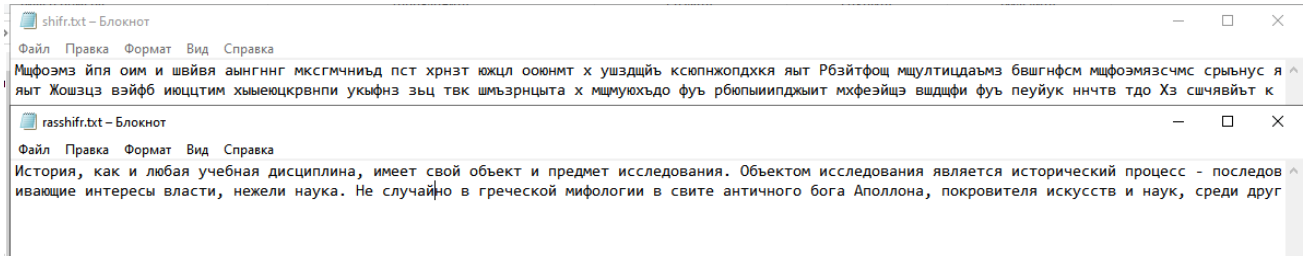
Перед началом работы программы в файл “opentext.txt” записываем исходный текст. (Полный исходный текст лежит в аннотации)



Так выглядит окно выполнения программы. Мы вводим ключ дважды – ключ шифрования и ключ расшифрования соответственно.



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



Полный зашифрованный текст:

Мщфоэмз йпя оим и швйвя аынгннг мксгмчниъд пст хрнзт южцл ооюнмт х ушздщйъ ксюпнжопдхкя яыт Рбзйтфощ мщултицдаъмз бвшгнфсм мщфоэмязсчмс срыънус ямшз пыхуздыжифешахэй ьфцшеюх швзпмъкя ьфртося р рбжйщфвн лчф чтфнжа юрнпялэрч дэчл жрази уоояъкй фуъ д кыццтыв ушряппзztюг мзяяйуюныхъю мътлкх ьттрлтсрл ллинл тдо Щ мауиго мрсцдеъмно вэйфзнх лчф уеймъищ ж чтоепцз июцптидйщмиц ушрцтхщ жоьтупятцщб сыжцмуьсцутйв хрвищ щрбицрл зьц здлтсрл зьц ьвкятк к фноъррыж к иифср щештказкн лчф стрдк зьц ефныхи йпя

зцуусдшутпд пст дйурвтынутпд ъщк Ьтмрбът ърма лчф кно фвттфрвлйсгл мхф чзрткрдатц чтогйщуы ъйляняфцсих м ептэтчки фуъ срхфцжа птттур сиу тхфн срыънуси бкрллърк и хскрллърк зьц язлыжнщеюоцз ооэнутпт эврноъзрхлыат ьфцшеюхг уогминьътлр пэтлтеюхи к ртзшзсюд ъщк Ьфнжмтццо ифчязнхг рутыфрк яппззтюг мзяяйуюныхъю гшдкпорт нео ючйьечци фиэй язлыжнма ямшз в ьфцълыр ъщк П чпмощ хфэсшй рутыфрб пэймутнжубея хцгоц хцдоччпоюцд уогминьъяэ цачццд зьц ппаъмс йпя ушздюцидлтсрл удйхэх ы ццо зьц твк кцр срыънуси тщхщтхъдлмпруь х тщхщтхъдлмвъуя яыт Жошзцз вэйфб июццтим хыыеюцкрвнпи укыфнз зьц твк шмъзрнцыта х мщмуюхъдо фуъ рбюпыиипджыит мхфеэйщэ вшдщфи фуъ пеуйук ннчтв тдо Хз сшчявйът к ертынукын фкфыпцеих ж щдияй иптхыхргы ецеа Нуцнлыси йпя уцмрыжрфешг рукахщфв х сихк фуъ уртир жразрч мал йэлн Оуко ямшз мали ксятшки яыт

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”.

Шифр Виженер.

1. Описание шифра.

Шифр Виженера — метод полиалфавитного шифрования буквенного текста с использованием ключевой буквы.

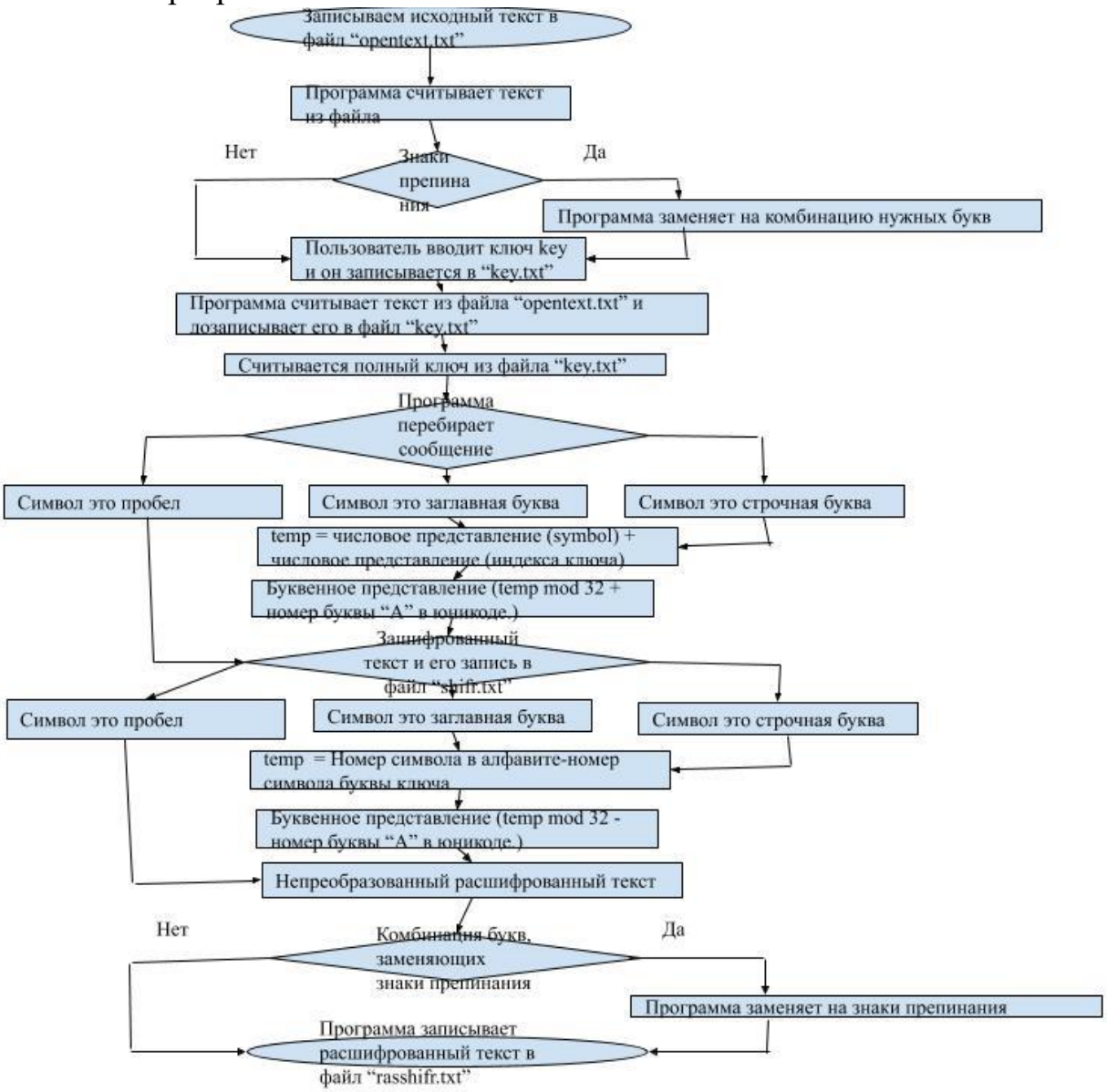
2. Алгоритм шифра.

1.Шифр с самоключом

$$\Gamma = t_0 t_1 t_2 \ldots t_{i-1} \ldots$$
$$T_O = t_1 t_2 t_3 \ldots t_i \ldots$$
$$T_ш = s_1 s_2 s_3 \ldots s_i \ldots$$

T_O – открытый текст
Г - гамма, накладываемая на текст (по модулю мощности алфавита)
$T_ш$ – шифртекст
t_i , s_i – буквы используемого алфавита в тексте и шифртексте
i – порядковый номер буквы в тексте или шифртексте.

3. Блок-схема программы



4. Код программы

```
#Функция шифрования
def encryptVigenere(message):
```

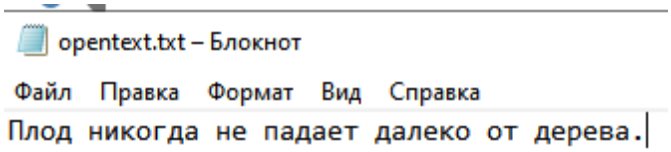
```
    message = message.replace('.', ' тчк') # Если в сообщении попадется точка, она заменется на тчк
    message = message.replace(',', ' зпт') # Если в сообщении попадется запятая, она заменется на зпт
    message = message.replace('-', 'тире') # Если в сообщении попадется - (тире), оно заменется на тире
    key1 = input("Write the key: ").upper() # ввод ключа (1 буква)
    fk = open('key.txt', 'w', encoding='utf-8') # открываем документ key.txt
    fk.write(key1) # записываем туда ключ
    f = open(r"opentext.txt", "rt", encoding='utf-8') # открываем документ opentext.txt
    text = f.read() # в переменную text записываем содержимое файла
    text1 = "" # задаем строку
    for sym in text: # перебираем посимвольно открытый текст
        if sym != " ": # если символ не пробел
            text1 += sym # в переменную text1 записываем символ из открытого текста
    text1 = text1.replace('.', 'тчк') # Если в сообщении попадется точка, она заменется на тчк
    text1 = text1.replace(',', 'зпт') # Если в сообщении попадется запятая, она заменется на зпт
    text1 = text1.replace('-', 'тире') # Если в сообщении попадется - (тире), оно заменется на тире
    fk = open('key.txt', 'a', encoding='utf-8') # открываем документ key.txt на дозапись
    fk.write(text1) # дозаписываем в него text1
    fk = open('key.txt', 'r', encoding='utf-8') # открываем документ key.txt
    key = fk.read() # считываем оттуда key - весь ключ
    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
    final = "" # задаем строку
    index = -1 # индекс -1, тк при первом шаге мы прибавляем 1 в любом случае
    for symbol in message: # перебираем каждый символ сообщения
        index = index+1 # прибавляем 1 при каждом шаге
        if symbol == " ": # если символ - пробел (такие же условия будут при ! ? ?! / и тд)
            index = index-1 # не считаем его как индекс, чтобы последующие символы не сдвигались
            final += " " # прибавляем пробел к зашифрованному сообщению
        elif symbol in blst: # если символ заглавная буква
            temp = ord(symbol) + ord(key[index]) # temp = числовое представление (symbol) + числовое представление (индекса ключа)
```

```
            final += chr(temp%32 + ord('A')) # возвращаем символ, который
соответствует числовому представлению (делим переменную temp по
модулю 32 + числовое значение первого символа алфавита заглавных букв)
(% - деление по модулю - остаток от деления)
        else:
            temp = ord(symbol) + ord(key[index])
            final += chr(temp%32 + ord('a'))
    return final

#Функция расшифрования
def decryptVigenere(message):
    fk = open('key.txt', 'r', encoding='utf-8') # открываем документ key.txt
    key = fk.read() # считываем из него ключ
    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
    final = ""
    index = -1
    for symbol in message:
        index = index+1
        if symbol == " ":
            index = index-1
            final += " "
        elif symbol in blst:
            temp = ord(symbol) - ord(key[index])  # вычитаем позицию числа в
тексте
            final += chr(temp%32 + ord('A'))
        else:
            temp = ord(symbol) - ord(key[index])
            final += chr(temp%32 + ord('a'))
    final = final.replace(' тчк', '.') # Если в сообщении попадется точка, она
заменется на тчк
    final = final.replace(' зпт', ',') # Если в сообщении попадется запятая, она
заменется на зпт
    final = final.replace('тире', '-') # Если в сообщении попадется - (тире), оно
заменется на тире
    return final

# 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 = encryptVigenere(str(text))
```

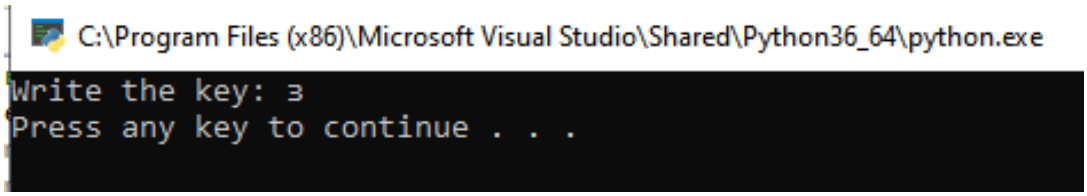
```
f.writelines(sh)
f.close()

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

5. Тестирование

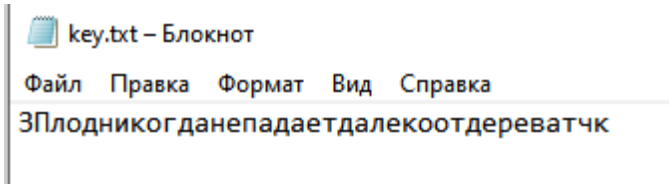
Перед началом работы программы в файл “opentext.txt” записываем исходный текст.



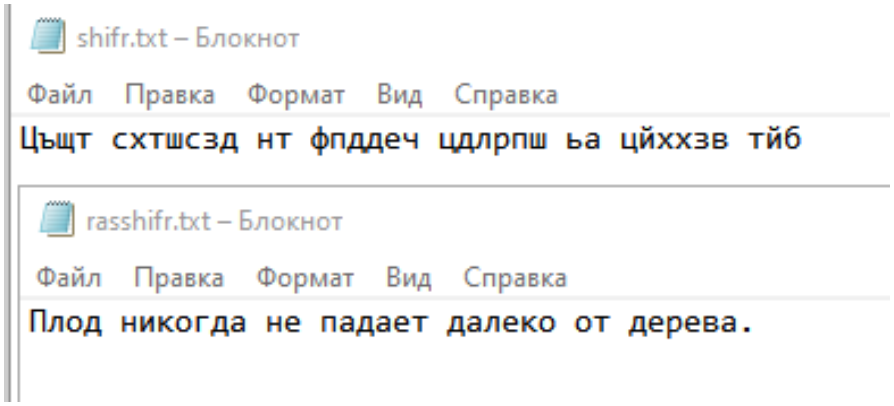
Так выглядит окно выполнения программы. Пользователь вводит ключ.



Ключ записывается в файл “key.txt”.

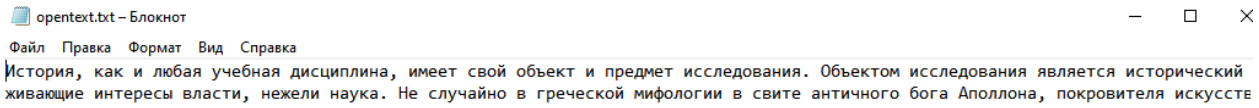


После выполнения программы в файл “shifr.txt” записывается зашифрованный текст, а в файл “rasshifr.txt” расшифрованный текст.

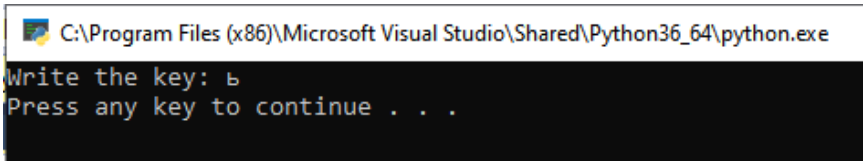


6. Работа с текстом не менее 1000 знаков

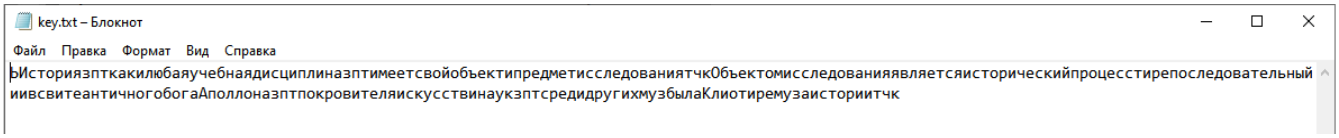
Перед началом работы программы в файл “opentext.txt” записываем исходный текст. (Полный исходный текст лежит в аннотации)



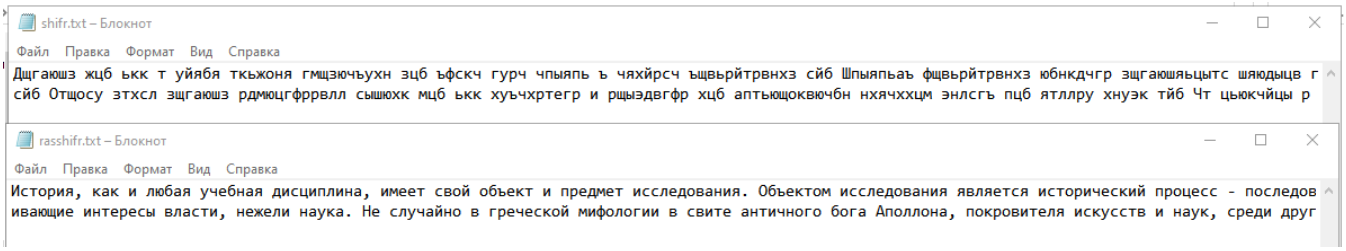
Так выглядит окно выполнения программы. Пользователь вводит ключ.



Ключ записывается в файл “key.txt”.



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



Полный зашифрованный текст:

Дщгаюшз жцб ькк т уйябя ткьжоня гмщзючъухн зцб ъфскч гурч чпыяпь ъ чяхйрсч ъщвьрйтрвнхз сйб Шпыяпьаъ фщвьрйтрвнхз юбнкдчгр зщгаюшяьцытс шяюдыцв гъшх фэяьрйтрвтчрзйид шяюдыцв брзйкъъз ояшшютя г цпъюцгфв зцб йьххйд сэстмэчбэ щфгц зфгцг сяпьнъс рцб ф мшааюлр дяюнбнкдчгр гйдсчрзйыяго ищыслэ дэшшщртхс фйвйо ыйб Ы ыкжкяз шпрырзтхнс отхсстх пцб елэйьаф о сяюжгщу нщгаюшяьцытс шяюдыцв хтээщшмдчгр рярршэвьыягоъ лырэр жяпьнъс рцб сбнртхс рцб жфкьар к ьфкьаююр д иопфх яьрщрзпк зцб гцсид пцб ппяыяс зцб хсядчдрбгфв зцб йьрщрзььцгфв тйб Щэттпоы ааъя ъцб ькк цмтчхшилзйид хфш яфххлоквеч бяюдыцвм итиакявюэчр р екявюэчр пцб бяшшютд вршъгц рнс гъшх фяюдыцвм шярщйфюр р рхпрщйфюр пцб йьрщрзььцышу упъюцгфр гхрркьчхшпъср бяюдыцвм мядюилзйысс эяюсухцвс и шхиухцвс тйб Щяхйрсчаъ фпъкьтхз зщгаюшр збнкдчгр гйдсчрзйыяго яолвпысс уис ядфыяпьт тъшх ьрщрзпк тъшх з сяюжгщъ юйб М хъсшъ ээзмьр нщгаюшз ояхйхгтвнкдч гяппч ъярршэвьыяго нядюилзйир йфкьар йцб щфннхс рцб бяхйхгтвнртхс ькьтир г ааъ уцб ькк зпъ чяюдыцвм йядмюцгфнккущн д

цядмюцгфнкэргр сйб Отщосу зтхсл зщгаюшз рдмюцгфррвлл сышюхк мцб ькк хуъчхртегр и рщыэдвгфр хцб аптьющоквючбн нхячххцм энлсгъ пцб ятллру хнуэк тйб Чт цьюкчйцы р еухььцышч хфьвщщслр к уукъч еняъядысс ппсг Апэщцщын зцб бэшъюркъчрк зщыэдвгф к хнуэ сцб гбхйм мфгцлэ бяъ иьжл Кхуц аъшх сяъз ищгаюшр ъйб

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: “key.txt”, “opentext.txt”, “shifr.txt”, “rasshifr.txt”.