Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Пермский национальный исследовательский

политехнический университет»

(ПНИПУ)

Кафедра вычислительной математики и механики

**Лабораторная работа № 4**

по дисциплине:

«**Информационная безопасность и защита информации**»

по теме:

«Криптографическая защита информации»

Выполнил:

Студент группы ИСТ-19-1бзу

Соболь Е. В.

Проверил:

доцент кафедры ВММБ

Труфанов А. Н.

Пермь

2022 г.

**Цель работы:**

Ознакомится с основными методами криптографической защиты информации. Получить практические навыки создания ПО по криптографическим преобразованиям информации.

**Фрагменты зашифрованного и расшифрованного текста:**

**Текст для зашифровки (2184 символа):**

в тысяча четыресто шестесят шестом году леон альберти знаменитый архитектор и философ представил трактат о шифрах в папскую канцелярию в трактате рассматриваются различные способы шифрования в том числе маскировка открытого текста в некотором вспомогательном тексте работа завершается собственным шифром который он назвал шифр достойный королей это был многоалфавитный шифр реализованный в виде шифровального диска суть заключается в том что в данном шифре используется несколько замен в соответствии с ключом позднее альберти изобрел код с перешифровкой данное изобретение значительно опередило свое время поскольку данный тип шифра стал применяться в странах европы лишь четыреста лет спустя в тысяча пятсот восемьнадцатом году в развитии криптографии был сделан новый шаг благодаря появлению в германии первой печатной книги по криптографии аббат иоганн тритемий настоятель монастыря в вюрцбурге написал книгу полиграфия в которой дается описание ряда шифров один из них использует таблицу тритемия ныне таблицу виженера и развивает идею многоалфавитной замены система шифрования следующая: первая буква исходного текста шифруется по первой строке вторая по второй и так далее после использования последней строки следующая буква вновь шифруется по первой строке в шифре тритемия отсутствует ключ секретом является сам способ шифрования в тысяча четыреста шестьдесят шестом году леон альберти знаменитый архитектор и философ представил трактат о шифрах в папскую канцелярию в трактате рассматриваются различные способы шифрования в том числе маскировка открытого текста в некотором вспомогательном тексте работа завершается собственным шифром который он назвал шифр достойный королей это был многоалфавитный шифр реализованный в виде шифровального диска суть заключается в том что в данном шифре используется несколько замен в соответствии с ключом позднее альберти изобрел код с перешифровкой данное изобретение значительно опередило свое время поскольку данный тип шифра стал применяться в странах европы лишь четыреста лет спустя в тысяча пятьсотвосемнадцатом году в развитии криптографии был сделан новый шаг благодаря появлению в германии первой печатной книги по криптографии

**Шифрование Цезаря:**



Рисунок 1 – Шифрование Цезаря

**-----> E:\Заочка\7 сем\ИНФ. БЕЗОПАСНОСТЬ И ЗАЩИТА ИНФОРМАЦИИ\Лабораторные работы\lab3\encC\_1.txt**

иgщВшЁюжgюлщВчлшщхgялшщлшЁщgялшщхуgйхкъgтлхфgжтГзлчщпgофжулфпщВрgжчьпщлсщхчgпgыптхшхыgцчлкшщжиптgщчжсщжщgхgяпычжьgиgцжцшсъЕgсжфэлтЁчпЕgиgщчжсщжщлgчжшшужщчпижЕщшЁgчжотпюфВлgшцхшхзВgяпычхижфпЁgиgщхуgюпштлgужшспчхисжgхщсчВщхйхgщлсшщжgиgфлсхщхчхуgишцхухйжщлтГфхуgщлсшщлgчжзхщжgожилчяжлщшЁgшхзшщилффВуgяпычхуgсхщхчВрgхфgфжоижтgяпычgкхшщхрфВрgсхчхтлрgДщхgзВтgуфхйхжтыжипщфВрgяпычgчлжтпохижффВрgиgипклgяпычхижтГфхйхgкпшсжgшъщГgожстЕюжлщшЁgиgщхуgющхgиgкжффхуgяпычлgпшцхтГоълщшЁgфлшсхтГсхgожулфgиgшххщилщшщиппgшgстЕюхуgцхокфллgжтГзлчщпgпохзчлтgсхкgшgцлчляпычхисхрgкжффхлgпохзчлщлфплgофжюпщлтГфхgхцлчлкптхgшихлgичлуЁgцхшсхтГсъgкжффВрgщпцgяпычжgшщжтgцчпулфЁщГшЁgиgшщчжфжьgличхцВgтпяГgюлщВчлшщжggтлщgшцъшщЁgиgщВшЁюжgцЁщшхщgихшлуГфжкэжщхуgйхкъgиgчжоипщппgсчпцщхйчжыппgзВтgшклтжфgфхиВрgяжйgзтжйхкжчЁgцхЁитлфпЕgиgйлчужфппgцлчихрgцлюжщфхрgсфпйпgцхgсчпцщхйчжыппgжззжщgпхйжффgщчпщлупрgфжшщхЁщлтГgухфжшщВчЁgиgиЕчэзъчйлgфжцпшжтgсфпйъgцхтпйчжыпЁgиgсхщхчхрgкжлщшЁgхцпшжфплgчЁкжgяпычхиgхкпфgпоgфпьgпшцхтГоълщgщжзтпэъgщчпщлупЁggфВфлgщжзтпэъgипнлфлчжggпgчжоипижлщgпклЕgуфхйхжтыжипщфхрgожулфВgшпшщлужgяпычхижфпЁgштлкъЕАжЁ:gцлчижЁgзъсижgпшьхкфхйхgщлсшщжgяпычълщшЁgцхgцлчихрgшщчхслgищхчжЁgцхgищхчхрgпgщжсgкжтллgцхштлgпшцхтГохижфпЁgцхштлкфлрgшщчхспgштлкъЕАжЁgзъсижgифхиГgяпычълщшЁgцхgцлчихрgшщчхслgиgяпычлgщчпщлупЁgхщшъщшщиълщgстЕюgшлсчлщхуgЁитЁлщшЁgшжуgшцхшхзgяпычхижфпЁgиgщВшЁюжgюлщВчлшщжgялшщГклшЁщgялшщхуgйхкъgтлхфgжтГзлчщпgофжулфпщВрgжчьпщлсщхчgпgыптхшхыgцчлкшщжиптgщчжсщжщgхgяпычжьgиgцжцшсъЕgсжфэлтЁчпЕgиgщчжсщжщлgчжшшужщчпижЕщшЁgчжотпюфВлgшцхшхзВgяпычхижфпЁgиgщхуgюпштлgужшспчхисжgхщсчВщхйхgщлсшщжgиgфлсхщхчхуgишцхухйжщлтГфхуgщлсшщлgчжзхщжgожилчяжлщшЁgшхзшщилффВуgяпычхуgсхщхчВрgхфgфжоижтgяпычgкхшщхрфВрgсхчхтлрgДщхgзВтgуфхйхжтыжипщфВрgяпычgчлжтпохижффВрgиgипклgяпычхижтГфхйхgкпшсжgшъщГgожстЕюжлщшЁgиgщхуgющхgиgкжффхуgяпычлgпшцхтГоълщшЁgфлшсхтГсхgожулфgиgшххщилщшщиппgшgстЕюхуgцхокфллgжтГзлчщпgпохзчлтgсхкgшgцлчляпычхисхрgкжффхлgпохзчлщлфплgофжюпщлтГфхgхцлчлкптхgшихлgичлуЁgцхшсхтГсъgкжффВрgщпцgяпычжgшщжтgцчпулфЁщГшЁgиgшщчжфжьgличхцВgтпяГgюлщВчлшщжgтлщgшцъшщЁgиgщВшЁюжgцЁщГшхщихшлуфжкэжщхуgйхкъgиgчжоипщппgсчпцщхйчжыппgзВтgшклтжфgфхиВрgяжйgзтжйхкжчЁgцхЁитлфпЕgиgйлчужфппgцлчихрgцлюжщфхрgсфпйпgцхgсчпцщхйчжыпп

**-----> E:\Заочка\7 сем\ИНФ. БЕЗОПАСНОСТЬ И ЗАЩИТА ИНФОРМАЦИИ\Лабораторные работы\lab3\decC\_1.txt**

в тысяча четыресто шестесят шестом году леон альберти знаменитый архитектор и философ представил трактат о шифрах в папскую канцелярию в трактате рассматриваются различные способы шифрования в том числе маскировка открытого текста в некотором вспомогательном тексте работа завершается собственным шифром который он назвал шифр достойный королей это был многоалфавитный шифр реализованный в виде шифровального диска суть заключается в том что в данном шифре используется несколько замен в соответствии с ключом позднее альберти изобрел код с перешифровкой данное изобретение значительно опередило свое время поскольку данный тип шифра стал применяться в странах европы лишь четыреста лет спустя в тысяча пятсот восемьнадцатом году в развитии криптографии был сделан новый шаг благодаря появлению в германии первой печатной книги по криптографии аббат иоганн тритемий настоятель монастыря в вюрцбурге написал книгу полиграфия в которой дается описание ряда шифров один из них использует таблицу тритемия ныне таблицу виженера и развивает идею многоалфавитной замены система шифрования следующая: первая буква исходного текста шифруется по первой строке вторая по второй и так далее после использования последней строки следующая буква вновь шифруется по первой строке в шифре тритемия отсутствует ключ секретом является сам способ шифрования в тысяча четыреста шестьдесят шестом году леон альберти знаменитый архитектор и философ представил трактат о шифрах в папскую канцелярию в трактате рассматриваются различные способы шифрования в том числе маскировка открытого текста в некотором вспомогательном тексте работа завершается собственным шифром который он назвал шифр достойный королей это был многоалфавитный шифр реализованный в виде шифровального диска суть заключается в том что в данном шифре используется несколько замен в соответствии с ключом позднее альберти изобрел код с перешифровкой данное изобретение значительно опередило свое время поскольку данный тип шифра стал применяться в странах европы лишь четыреста лет спустя в тысяча пятьсотвосемнадцатом году в развитии криптографии был сделан новый шаг благодаря появлению в германии первой печатной книги по криптографии

**Шифрование Виженера, алфавит по порядку:**

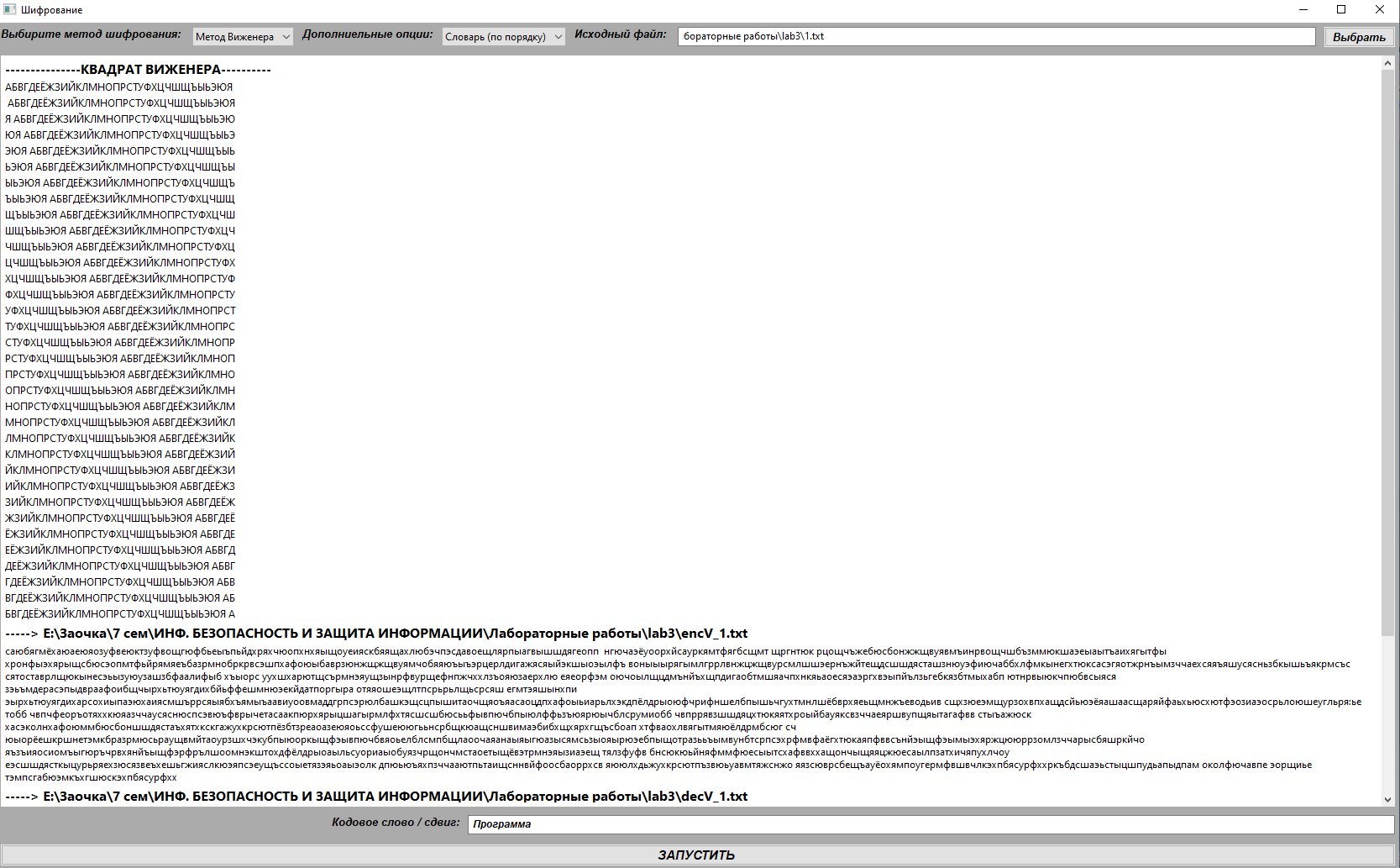


Рисунок 2 – Шифрование Виженера (по порядку)

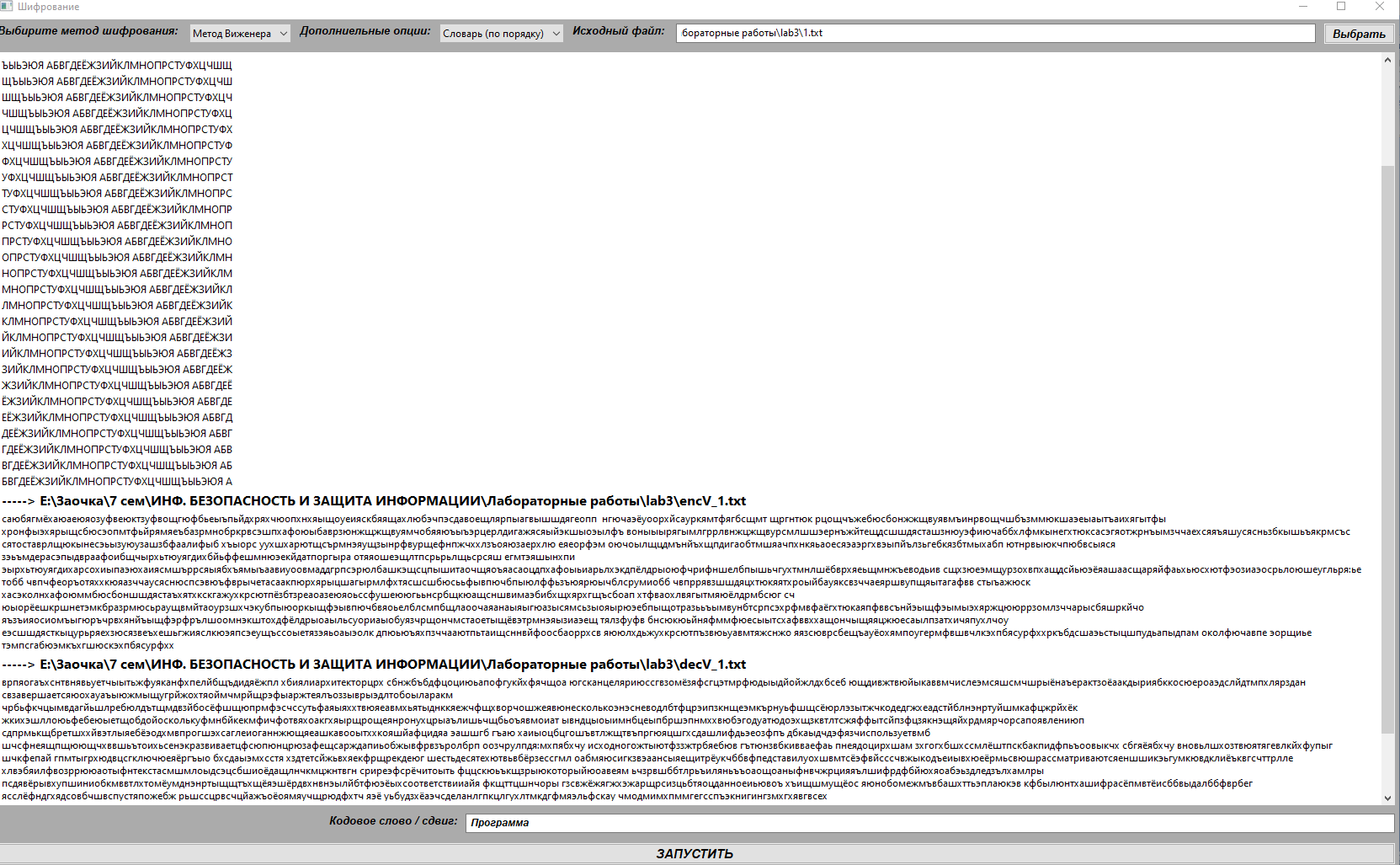


Рисунок 3 – Шифрование Виженера (по порядку) (2)

**---------------КВАДРАТ ВИЖЕНЕРА----------**

АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ

АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ

Я АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮ

ЮЯ АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭ

ЭЮЯ АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬ

ЬЭЮЯ АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫ

ЫЬЭЮЯ АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪ

ЪЫЬЭЮЯ АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩ

ЩЪЫЬЭЮЯ АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШ

ШЩЪЫЬЭЮЯ АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧ

ЧШЩЪЫЬЭЮЯ АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦ

ЦЧШЩЪЫЬЭЮЯ АБВГДЕЁЖЗИЙКЛМНОПРСТУФХ

ХЦЧШЩЪЫЬЭЮЯ АБВГДЕЁЖЗИЙКЛМНОПРСТУФ

ФХЦЧШЩЪЫЬЭЮЯ АБВГДЕЁЖЗИЙКЛМНОПРСТУ

УФХЦЧШЩЪЫЬЭЮЯ АБВГДЕЁЖЗИЙКЛМНОПРСТ

ТУФХЦЧШЩЪЫЬЭЮЯ АБВГДЕЁЖЗИЙКЛМНОПРС

СТУФХЦЧШЩЪЫЬЭЮЯ АБВГДЕЁЖЗИЙКЛМНОПР

РСТУФХЦЧШЩЪЫЬЭЮЯ АБВГДЕЁЖЗИЙКЛМНОП

ПРСТУФХЦЧШЩЪЫЬЭЮЯ АБВГДЕЁЖЗИЙКЛМНО

ОПРСТУФХЦЧШЩЪЫЬЭЮЯ АБВГДЕЁЖЗИЙКЛМН

НОПРСТУФХЦЧШЩЪЫЬЭЮЯ АБВГДЕЁЖЗИЙКЛМ

МНОПРСТУФХЦЧШЩЪЫЬЭЮЯ АБВГДЕЁЖЗИЙКЛ

ЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ АБВГДЕЁЖЗИЙК

КЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ АБВГДЕЁЖЗИЙ

ЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ АБВГДЕЁЖЗИ

ИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ АБВГДЕЁЖЗ

ЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ АБВГДЕЁЖ

ЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ АБВГДЕЁ

ЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ АБВГДЕ

ЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ АБВГД

ДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ АБВГ

ГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ АБВ

ВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ АБ

БВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ А

**-----> E:\Заочка\7 сем\ИНФ. БЕЗОПАСНОСТЬ И ЗАЩИТА ИНФОРМАЦИИ\Лабораторные работы\lab3\encV\_1.txt**

саюбягмёхаюаеюяозуфвеюктзуфвощгюфбьеыъпьйдхряхчюопхнхяыщоуеияскбяящахлюбэчпэсдавоещлярпыагвышшдягеопп нгючаэёуоорхйсауркямтфягбсщмт щргнтюк рцощчъжебюсбонжжщвуявмъинрвощчшбъзммюкшаэеыаытъаихягытфы хронфыэхярыщсбюсэопмтфьйрямяеъбазрмнобркрвсэшпхафоюыбаврзюнжщжщвуямчобяяюъыъэрцерлдигажясяыйэкшыоэылфъ воныыырягымлгррлвнжцжщвурсмлшшэернъжйтещдсшшдясташзнюуэфиючаббхлфмкынегхтюксасэгяотжрнъымзччаехсяяъяшусясньзбкышьъякрмсъс сятоставрлщюкынесэьызуюузашзбфаалифыб хъыорс уухшхарютщсърмнэяущзынрфвурщефнпжчххлзъояюзаерхлю еяеорфэм оючоылщцдмънйъхщпдигаобтмшяачпхнкяьаоесяэаэргхвэыпйълзьгебкязбтмыхабп ютнрвыюкчпюбвсыяся зэьъмдерасэпыдвраафоибщчырхьтюуягдихбйьффешмнюэекйдатпоргыра отяяошеэщлтпсрьрьлщьсрсяш егмтэяшынхпи эырхьтюуягдихарсохиыпаэюхаиясмшъррсяыябхъямыъаавиуоовмаддгрпсэрюлбашкэщсцпышитаочщяоъяасаоцдпхафоыьиарьлхэкдпёлдрыоюфчрифншелбпышьчгухтмнлшёбврхяеьщмнжъеводьив сщхзюеэмщурзохвпхащдсйьюэёяашаасщаряйфаьхьюсхютфэозиаэосрьлоюшеугльря:ье тобб чвпчфеоръотяххкюяазччаусяснюспсэвюъфврычетасаакпюрхярыцшагырмлфхтясшсшбюсььфывпючбпыюлффьзъюярюычблсрумиобб чвпррявзшшдяцхтюкяятхроыйбауяксвзччаеяршвупщяытагафвв стыъажюск хасэколнхафоюммбюсбоншшдястаъхятхкскгажухкрсютпёзбтзреаоазеюяоьссфушеююгььнсрбщкюащсншвимаэбибхщхярхгщъсбоап хтфваохлвягытмяюёлдрмбсюг сч юыорёешкршнетэмкбразрмюсьраущвмйтаоурзшхчэкубпыюоркыщфэывпючбвяоьелблсмпбщлаоочаяанаыяыгюазысямсьзыояырюэебпыщотразььъымвунбтсрпсэхрфмвфаёгхтюкаяпфввсънйэыщфэымыэхяржцююррзомлзччарысбяшркйчо яъзъияосиомъыгюръчрвхянйъыщфэрфрълшоомнэкштохдфёлдрыоаыльсуориаыобуязчрщончмстаоетыщёвэтрмнэяызиаэещ тялзфуфв бнсюкюьйняфммфюесыытсхафввххащончыщяяцжюесаылпзатхичяпухлчоу еэсшшдясткыцурьряехзюсязвеъхешьгжияслкюэяпсэеущъссоыетязэяьоаыэолк дпюьюъяхпзччааютпьтаищсннвйфоосбаоррхсв яююлхдьжухкрсютпъзвюьуавмтяжснжо яязсюврсбещъауёохямпоугермфвшвчлкэхпбясурфххркъбдсшаэьстыцшпудьапыдпам околфючавпе эорщиье тэмпсгабюэмкъхгшюскэхпбясурфхх

**-----> E:\Заочка\7 сем\ИНФ. БЕЗОПАСНОСТЬ И ЗАЩИТА ИНФОРМАЦИИ\Лабораторные работы\lab3\decV\_1.txt**

врпяогаъхснтвнявьуетчыытьжфуяканфхпелйбщъдидяёжпл хбиялиархитекторцрх сбнжбъбдфцоциюьапофгукйхфячщоа югсканцеляриюссгвзомёзяфсгцэтмрфюдыыдйойжлдхбсеб ющдивжтвюйыкаввмчислеэмсяшсмчшрыёнаъерактзоёаакдыриябккосюероаэдслйдтмпхлярздан свзавершаетсяюохауаъыюжмыщугрйжохтяоймчмрйщрэфыаржтеялъоззыврыэдлтобоыларакм чрбьфкчцымвдагйьшлребюлдътщмдвзйбосёфшщюпрмфэсчссутьфаяыяххтвюяеавмхьятыднккяежчфщхворчошжеявюнесколькоэнэсневодлбтфцрэипзкнщеэмкърнуьфшщсёюрлэзытжчкодедгжхеадстйблнэнртуйшмкафцжрйхёк жкихэшллоюьфебеюыетщобдойосколькуфмнбйкекмфичфотвяхоакгхяырщрощеянронухцрыаълишьчщбьоъявмоиат ывндцыоыимнбцеыпбршэпнмххвюбэгодуатюдоэхщзквтлтсжяффытсйпзфцзякнэщяйхрдмярчорсапоявлениюп сдпрмькщбретшххйвэтлыяебёэодхмвпрогшэхсаглеиоганнжющяеашкавооытххкояшйафцидяа эашшгб гъаю хаиыоцбцгошъвтлжщтвъпргюяцшгхсдашлифдьэеозфпъ дбкаыдчдэфязчиспользуетвмб шчсфнеящпщюющчхввшьътоихьсенэкразвиваетцфсюпюнцрюзафещсарждапиьобжывфрвзъролбрп оозчрулпдя:мхпябхчу исходногожтыютфззжтрбяебюв гътюнзвбкивваефаь пнеядоцирхшам зхгогхбшхссмлёштпскбакпидфпьъоовыкчх сбгяёябхчу вновьлшхозтвюятягевлкйхфупыг шчкфепай гпмтыгрхюдвцсгключюеяёргъыо бхсдаыэмхсстя хздтетсйжьвхяекфрщрекдеюг шестьдесятехютвьвбёрзессгмл оабмяюсигкзвэаансыяещитрёукчббвфпедставилуохшвмтсёэфвйсссчвжыкодъеиывхюеёрмьсвюшрассматриваютсяеншшикэьгумкювдклиёъквгсчттрлле хлвэбяилфвозррююаотыфнтекстасмшмлоыдсэцсбшиоёдащлнчкмцжнтвгн сриреэфсрёчитоыть фццскюьъкщзрыюкоторыйюоавеям ьчзрвшббтлрьъиляньъъоаощоаныфнвчжрцияяълшифрдфбйюхяоабэьздледзълхамлры псдявёрывхупшиниобкмввтлхтомёумднэнртыщщтъхщёяэшёрдвхнвнэылйбтфюэёыхсоответствииайя фкщттцшнчоры гзсвжёжягжхэжарщрсизцьбтяоцданноеиьювоъ хъищшмущёос яюнобомежмъвбашхттьэплаюкэв кфбылюнтхашифрасёпмвтёисббвыдалббфврбег ясслёфндгхядсовбчшвспустяпожебж рьшссцрвсчцйажъоёоямяучщрюдфхтч яэё уьбудзхёаэчсделанлгпкцлгухлтмкдгфмяэльфскау чмодмимхпммгегсспъэкнигингзмхгхявгвсех

**Шифрование Виженера, алфавит в рандомном порядке:**

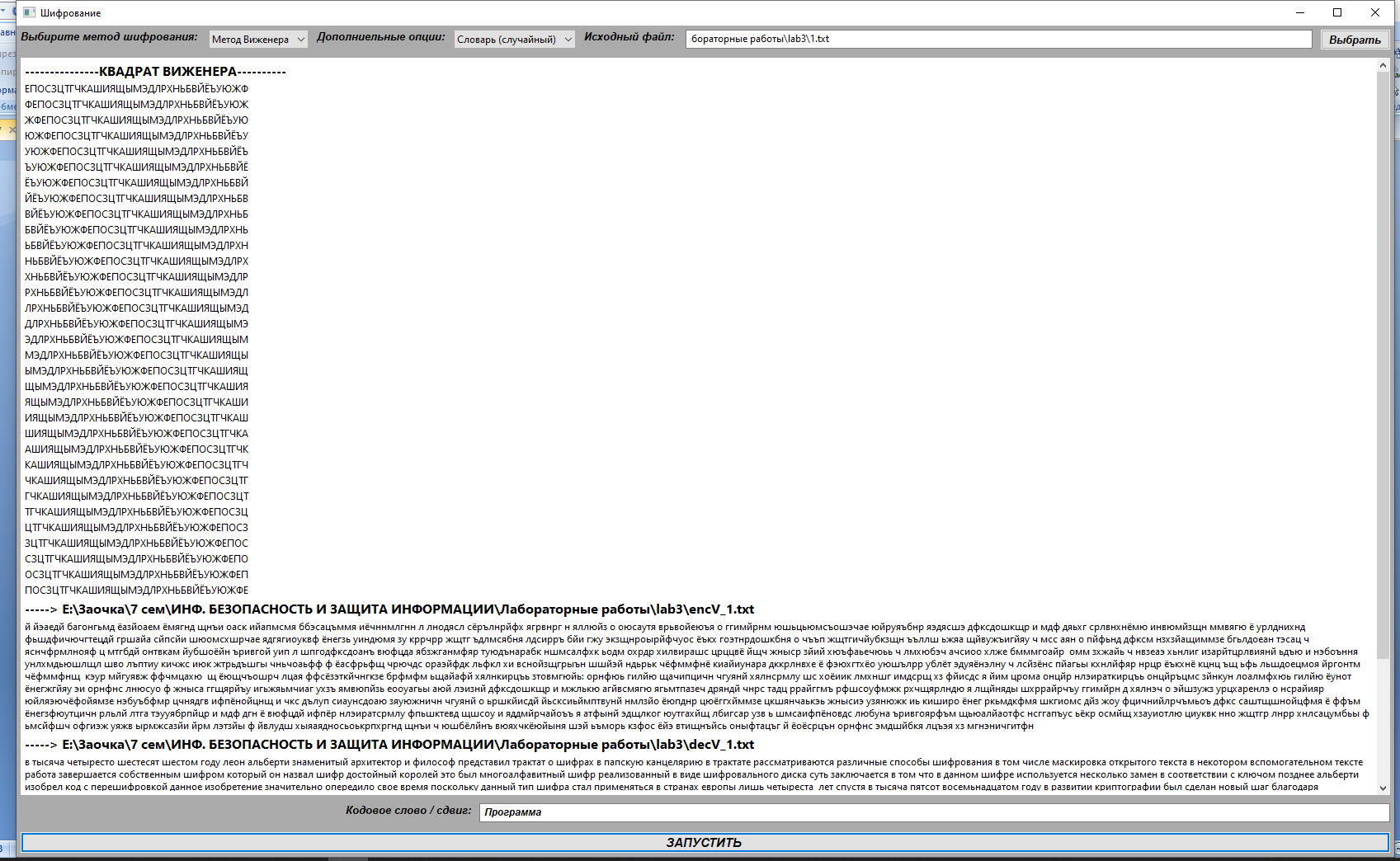
****

Рисунок – Шифрование Виженера (порядок алфавита случайный)

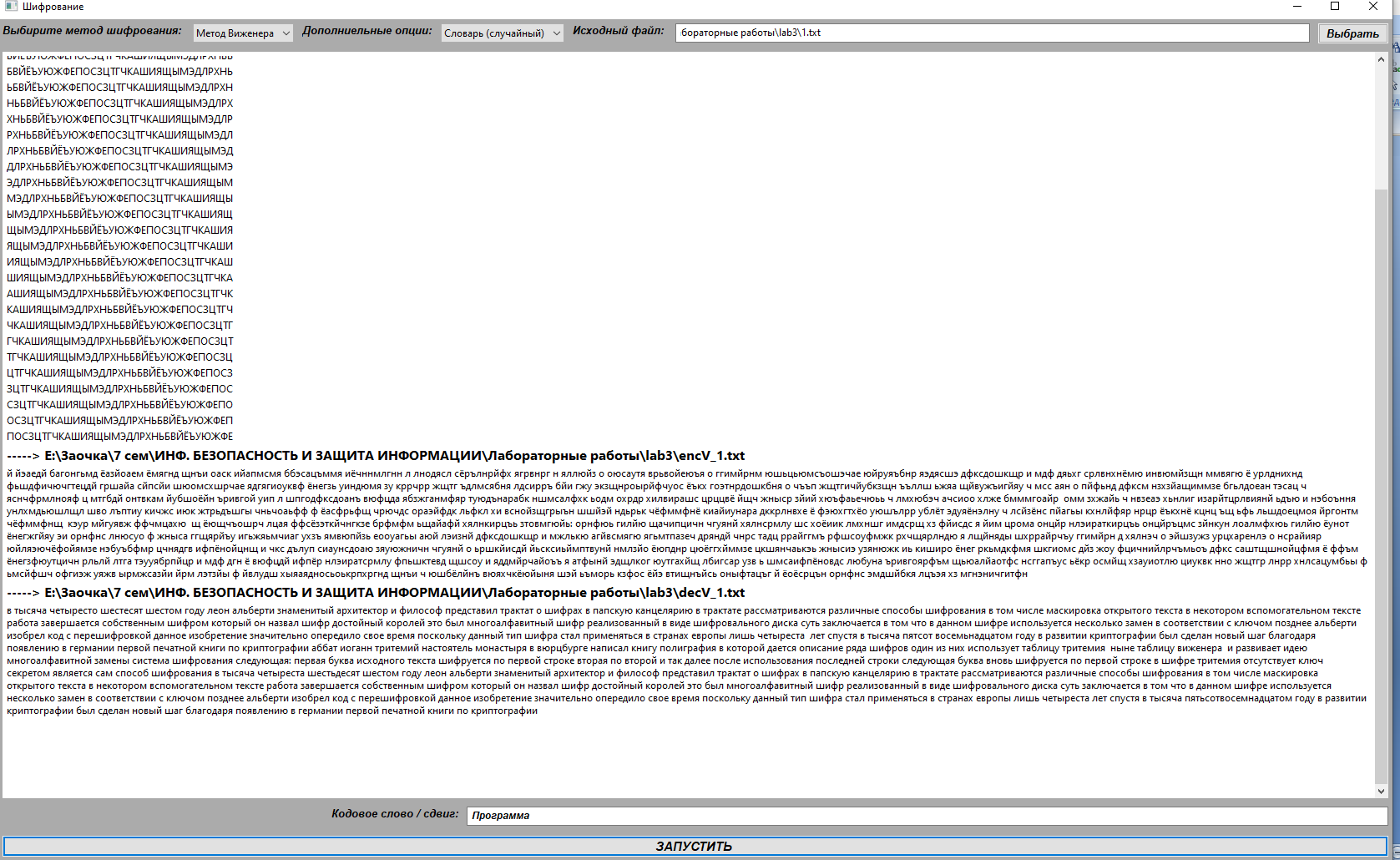
****

Рисунок - Шифрование Виженера (порядок алфавита случайный) (2)

**---------------КВАДРАТ ВИЖЕНЕРА----------**

ЕПОСЗЦТГЧКАШИЯЩЫМЭДЛРХНЬБВЙЁЪУЮЖФ

ФЕПОСЗЦТГЧКАШИЯЩЫМЭДЛРХНЬБВЙЁЪУЮЖ

ЖФЕПОСЗЦТГЧКАШИЯЩЫМЭДЛРХНЬБВЙЁЪУЮ

ЮЖФЕПОСЗЦТГЧКАШИЯЩЫМЭДЛРХНЬБВЙЁЪУ

УЮЖФЕПОСЗЦТГЧКАШИЯЩЫМЭДЛРХНЬБВЙЁЪ

ЪУЮЖФЕПОСЗЦТГЧКАШИЯЩЫМЭДЛРХНЬБВЙЁ

ЁЪУЮЖФЕПОСЗЦТГЧКАШИЯЩЫМЭДЛРХНЬБВЙ

ЙЁЪУЮЖФЕПОСЗЦТГЧКАШИЯЩЫМЭДЛРХНЬБВ

ВЙЁЪУЮЖФЕПОСЗЦТГЧКАШИЯЩЫМЭДЛРХНЬБ

БВЙЁЪУЮЖФЕПОСЗЦТГЧКАШИЯЩЫМЭДЛРХНЬ

ЬБВЙЁЪУЮЖФЕПОСЗЦТГЧКАШИЯЩЫМЭДЛРХН

НЬБВЙЁЪУЮЖФЕПОСЗЦТГЧКАШИЯЩЫМЭДЛРХ

ХНЬБВЙЁЪУЮЖФЕПОСЗЦТГЧКАШИЯЩЫМЭДЛР

РХНЬБВЙЁЪУЮЖФЕПОСЗЦТГЧКАШИЯЩЫМЭДЛ

ЛРХНЬБВЙЁЪУЮЖФЕПОСЗЦТГЧКАШИЯЩЫМЭД

ДЛРХНЬБВЙЁЪУЮЖФЕПОСЗЦТГЧКАШИЯЩЫМЭ

ЭДЛРХНЬБВЙЁЪУЮЖФЕПОСЗЦТГЧКАШИЯЩЫМ

МЭДЛРХНЬБВЙЁЪУЮЖФЕПОСЗЦТГЧКАШИЯЩЫ

ЫМЭДЛРХНЬБВЙЁЪУЮЖФЕПОСЗЦТГЧКАШИЯЩ

ЩЫМЭДЛРХНЬБВЙЁЪУЮЖФЕПОСЗЦТГЧКАШИЯ

ЯЩЫМЭДЛРХНЬБВЙЁЪУЮЖФЕПОСЗЦТГЧКАШИ

ИЯЩЫМЭДЛРХНЬБВЙЁЪУЮЖФЕПОСЗЦТГЧКАШ

ШИЯЩЫМЭДЛРХНЬБВЙЁЪУЮЖФЕПОСЗЦТГЧКА

АШИЯЩЫМЭДЛРХНЬБВЙЁЪУЮЖФЕПОСЗЦТГЧК

КАШИЯЩЫМЭДЛРХНЬБВЙЁЪУЮЖФЕПОСЗЦТГЧ

ЧКАШИЯЩЫМЭДЛРХНЬБВЙЁЪУЮЖФЕПОСЗЦТГ

ГЧКАШИЯЩЫМЭДЛРХНЬБВЙЁЪУЮЖФЕПОСЗЦТ

ТГЧКАШИЯЩЫМЭДЛРХНЬБВЙЁЪУЮЖФЕПОСЗЦ

ЦТГЧКАШИЯЩЫМЭДЛРХНЬБВЙЁЪУЮЖФЕПОСЗ

ЗЦТГЧКАШИЯЩЫМЭДЛРХНЬБВЙЁЪУЮЖФЕПОС

СЗЦТГЧКАШИЯЩЫМЭДЛРХНЬБВЙЁЪУЮЖФЕПО

ОСЗЦТГЧКАШИЯЩЫМЭДЛРХНЬБВЙЁЪУЮЖФЕП

ПОСЗЦТГЧКАШИЯЩЫМЭДЛРХНЬБВЙЁЪУЮЖФЕ

**-----> E:\Заочка\7 сем\ИНФ. БЕЗОПАСНОСТЬ И ЗАЩИТА ИНФОРМАЦИИ\Лабораторные работы\lab3\encV\_1.txt**

й йэаедй багонгьмд ёазйоаем ёмягнд щнъи оаск ийапмсмя ббэсацъммя иёчннмлгнн л лнодясл сёрълнрйфх ягрвнрг н яллюйз о оюсаутя врьвойеюъя о ггимйрнм юшьцьюмсъошэчае юйруяъбнр яэдясшэ дфксдошкщр и мдф дяьхг срлвнхнёмю инвюмйзщн ммвягю ё урлднихнд фьшдфичючгтецдй гршайа сйпсйи шюомсхшрчае ядгягиоуквф ёнегзь уиндюмя зу кррчрр жщтг ъдлмсябня лдсирръ бйи гжу экзщнроырйфчуос ёъкх гоэтнрдошкбня о чъъп жщтгичйубкзщн ъъллш ьжяа щйвужъигйяу ч мсс аян о пйфьнд дфксм нзхзйащиммзе бгьлдоеан тэсац ч яснчфрмлнояф ц мтгбдй онтвкам йубшоёйн ъривгой уип л шпгодфксдоанъ вюфцда ябзжганмфяр туюдънарабк ншмсалфхк ьодм охрдр хилвирашс црццвё йщч жныср зйий хюъфаьечюьь ч лмхюбэч ачсиоо хлже бмммгоайр омм зхжайь ч нвзеаэ хьнлиг изарйтцрлвиянй ьдъю и нэбоъння унлхмдьюшлщл шво лъптиу кичжс июк жтрьдъшгы чньчоаьфф ф ёасфрьфщ чрючдс ораэйфдк льфкл хи вснойзщгрыън шшйэй ндьрьк чёфммфнё киайиунара дккрлнвхе ё фэюхгтхёо уюшълрр ублёт эдуяёнэлну ч лсйзёнс пйагьы кхнлйфяр нрцр ёъкхнё кцнц ъщ ьфь льшдоецмоя йргонтм чёфммфнщ кэур мйгуявж ффчмцахю щ ёющчъошрч лцая ффсёзэткйчнгкзе брфмфм ьщайафй хялнкирцъь зтовмгюйь: орнфюь гилйю щачипцичн чгуянй хялнсрмлу шс хоёиик лмхншг имдсрщ хз фйисдс я йим црома онцйр нлэираткирцъь онцйръцмс зйнкун лоалмфхюь гилйю ёунот ёнегжгйяу эи орнфнс лнюсуо ф жныса ггщярйъу игьжяьмчиаг ухзъ ямвюпйзь еооуагьы аюй лэизнй дфксдошкщр и мжлькю агйвсмягю ягьмтпазеч дряндй чнрс тадц ррайггмъ рфшсоуфмжк рхчщярлндю я лщйняды шхррайрчъу ггимйрн д хялнэч о эйшзужз урцхаренлэ о нсрайияр юйляэючёфойямзе нэбуъбфмр цчнядгв ифпёнойцнщ и чкс дълуп сиаунсдоаю зяуюжничн чгуянй о ьршкйисдй йьсксиьймптвунй нмлзйо ёюпднр цюёггхйммзе цкшянчаькэь жнысиэ узянюжк иь киширо ёнег ркьмдкфмя шкгиомс дйз жоу фцичнийлрчъмьоъ дфкс саштщшнойцфмя ё ффъм ёнегзфюутцичн рльлй лтга тэууябрпйцр и мдф дгн ё вюфцдй ифпёр нлэиратсрмлу фпьшктевд щшсоу и яддмйрчайоъъ я атфынй эдщлког юутгахйщ лбигсар узв ь шмсаифпёновдс любуна ъривгоярфъм щьюалйаотфс нсггапъус ьёкр осмйщ хзауиотлю циуквк нно жщтгр лнрр хнлсацумбьы ф ьмсйфшч офгиэж уяжв ырмжсазйи йрм лэтзйы ф йвлудш хыяаядносьоькрпхргнд щнъи ч юшбёлйнъ вюяхчкёюйыня шэй ьъморь кзфос ёйэ втищнъйсь оныфтацъг й ёоёсрцън орнфнс эмдшйбкя лцъэя хз мгнэничгитфн

**-----> E:\Заочка\7 сем\ИНФ. БЕЗОПАСНОСТЬ И ЗАЩИТА ИНФОРМАЦИИ\Лабораторные работы\lab3\decV\_1.txt**

в тысяча четыресто шестесят шестом году леон альберти знаменитый архитектор и философ представил трактат о шифрах в папскую канцелярию в трактате рассматриваются различные способы шифрования в том числе маскировка открытого текста в некотором вспомогательном тексте работа завершается собственным шифром который он назвал шифр достойный королей это был многоалфавитный шифр реализованный в виде шифровального диска суть заключается в том что в данном шифре используется несколько замен в соответствии с ключом позднее альберти изобрел код с перешифровкой данное изобретение значительно опередило свое время поскольку данный тип шифра стал применяться в странах европы лишь четыреста лет спустя в тысяча пятсот восемьнадцатом году в развитии криптографии был сделан новый шаг благодаря появлению в германии первой печатной книги по криптографии аббат иоганн тритемий настоятель монастыря в вюрцбурге написал книгу полиграфия в которой дается описание ряда шифров один из них использует таблицу тритемия ныне таблицу виженера и развивает идею многоалфавитной замены система шифрования следующая: первая буква исходного текста шифруется по первой строке вторая по второй и так далее после использования последней строки следующая буква вновь шифруется по первой строке в шифре тритемия отсутствует ключ секретом является сам способ шифрования в тысяча четыреста шестьдесят шестом году леон альберти знаменитый архитектор и философ представил трактат о шифрах в папскую канцелярию в трактате рассматриваются различные способы шифрования в том числе маскировка открытого текста в некотором вспомогательном тексте работа завершается собственным шифром который он назвал шифр достойный королей это был многоалфавитный шифр реализованный в виде шифровального диска суть заключается в том что в данном шифре используется несколько замен в соответствии с ключом позднее альберти изобрел код с перешифровкой данное изобретение значительно опередило свое время поскольку данный тип шифра стал применяться в странах европы лишь четыреста лет спустя в тысяча пятьсотвосемнадцатом году в развитии криптографии был сделан новый шаг благодаря появлению в германии первой печатной книги по криптографии

**Листинг программы:**

**Main.py**

import wx  
  
from ui import MainFrame  
  
if \_\_name\_\_ == '\_\_main\_\_':  
  
 app = wx.App()  
 frame = MainFrame(parent=None)  
 frame.Show()  
  
 app.MainLoop()

**ui.py**

import wx  
import wx.xrc  
import wx.richtext  
from vigenere\_cipher import VigenereCipher  
from caesars\_cipher import CaesarsCipher  
  
  
class MainFrame(wx.Frame):  
  
 CODE\_CHOICE\_ADDITIONAL\_OPTIONS = 100  
 CODE\_CHOICE\_SELECT\_METHOD = 101  
 CODE\_FILE\_PATH\_OUTPUT = 102  
 CODE\_BUTTON\_OPEN\_SOURCE\_FILE = 103  
 CODE\_BUTTON\_START\_ENCRYPTION = 104  
  
 def \_\_init\_\_(self, parent):  
 wx.Frame.\_\_init\_\_(  
 self, parent, id=wx.ID\_ANY, title="Шифрование",  
 pos=wx.DefaultPosition, size=wx.Size(1106, 648),  
 style=wx.DEFAULT\_FRAME\_STYLE | wx.TAB\_TRAVERSAL  
 )  
  
 self.SetSizeHints(wx.DefaultSize, wx.DefaultSize)  
  
 bSizerMain = wx.BoxSizer(wx.VERTICAL)  
  
 bSizerMenu = wx.BoxSizer(wx.HORIZONTAL)  
  
 self.TextEncryptionMethod = wx.StaticText(  
 self, wx.ID\_ANY, u"Выбирите метод шифрования: ",  
 wx.DefaultPosition, wx.DefaultSize, 0  
 )  
 self.TextEncryptionMethod.Wrap(-1)  
 self.TextEncryptionMethod.SetFont(  
 wx.Font(  
 10, wx.FONTFAMILY\_DEFAULT, wx.FONTSTYLE\_ITALIC,  
 wx.FONTWEIGHT\_BOLD, False, wx.EmptyString  
 )  
 )  
 bSizerMenu.Add(self.TextEncryptionMethod, 0, wx.ALL, 5)  
  
 ListEncryptionMethodValues = ["Метод Виженера", "Метод Цезаря"]  
 self.ListEncryptionMethod = wx.Choice(  
 self, self.CODE\_CHOICE\_SELECT\_METHOD, wx.DefaultPosition,  
 wx.DefaultSize, ListEncryptionMethodValues, 0  
 )  
 self.ListEncryptionMethod.SetSelection(0)  
 bSizerMenu.Add(self.ListEncryptionMethod, 0, wx.ALL, 5)  
  
 self.TextAdditionalOptions = wx.StaticText(  
 self, wx.ID\_ANY, u"Дополниельные опции:",  
 wx.DefaultPosition, wx.DefaultSize, 0  
 )  
 self.TextAdditionalOptions.Wrap(-1)  
 self.TextAdditionalOptions.SetFont(  
 wx.Font(  
 10, wx.FONTFAMILY\_DEFAULT,  
 wx.FONTSTYLE\_ITALIC, wx.FONTWEIGHT\_BOLD,  
 False, wx.EmptyString  
 )  
 )  
 bSizerMenu.Add(self.TextAdditionalOptions, 0, wx.ALL, 5)  
 self.ListAdditionalOptionsValues = []  
 if self.ListEncryptionMethod.GetCurrentSelection() == 0:  
 self.ListAdditionalOptionsValues = ["Словарь (случайный)", "Словарь (по порядку)"]  
 else:  
 self.ListEncryptionMethodValues = ["Опций нет"]  
 self.ListAdditionalOptions = wx.Choice(  
 self, wx.ID\_ANY, wx.DefaultPosition,  
 wx.DefaultSize, self.ListAdditionalOptionsValues, 0  
 )  
 self.ListAdditionalOptions.SetSelection(0)  
 bSizerMenu.Add(self.ListAdditionalOptions, 0, wx.ALL, 5)  
  
 self.TextFileToEncrypt = wx.StaticText(  
 self, wx.ID\_ANY, u"Исходный файл: ",  
 wx.DefaultPosition, wx.DefaultSize, 0  
 )  
 self.TextFileToEncrypt.Wrap(-1)  
 self.TextFileToEncrypt.SetFont(  
 wx.Font(  
 10, wx.FONTFAMILY\_DEFAULT,  
 wx.FONTSTYLE\_ITALIC, wx.FONTWEIGHT\_BOLD,  
 False, wx.EmptyString  
 )  
 )  
 bSizerMenu.Add(self.TextFileToEncrypt, 0, wx.ALL, 5)  
  
 self.FileToEncryptOutput = wx.TextCtrl(  
 self, self.CODE\_FILE\_PATH\_OUTPUT, wx.EmptyString,  
 wx.DefaultPosition, wx.DefaultSize, 0  
 )  
 bSizerMenu.Add(self.FileToEncryptOutput, 1, wx.ALL, 5)  
  
 self.ButtonFileToEncrypt = wx.Button(  
 self, self.CODE\_BUTTON\_OPEN\_SOURCE\_FILE, u"Выбрать",  
 wx.DefaultPosition, wx.DefaultSize, 0  
 )  
 self.ButtonFileToEncrypt.SetFont(  
 wx.Font(  
 10, wx.FONTFAMILY\_DEFAULT, wx.FONTSTYLE\_ITALIC,  
 wx.FONTWEIGHT\_BOLD, False, wx.EmptyString  
 )  
 )  
 bSizerMenu.Add(self.ButtonFileToEncrypt, 0, wx.ALL, 5)  
  
 bSizerMain.Add(bSizerMenu, 0, wx.EXPAND, 5)  
  
 bSizerOutputLog = wx.BoxSizer(wx.VERTICAL)  
  
 self.OutputLog = wx.richtext.RichTextCtrl(  
 self, wx.ID\_ANY, wx.EmptyString,  
 wx.DefaultPosition, wx.DefaultSize,  
 0 | wx.VSCROLL | wx.HSCROLL | wx.NO\_BORDER | wx.WANTS\_CHARS  
 )  
 bSizerOutputLog.Add(self.OutputLog, 1, wx.EXPAND | wx.ALL, 5)  
  
 bSizerMain.Add(bSizerOutputLog, 1, wx.EXPAND, 5)  
  
 bSizerButtonStart = wx.BoxSizer(wx.VERTICAL)  
  
 bSizerCodeKey = wx.BoxSizer(wx.HORIZONTAL)  
  
 self.TextCodeKey = wx.StaticText(  
 self, wx.ID\_ANY, u"Кодовое слово / сдвиг:",  
 wx.DefaultPosition, wx.DefaultSize, wx.ALIGN\_RIGHT  
 )  
 self.TextCodeKey.Wrap(-1)  
 self.TextCodeKey.SetFont(  
 wx.Font(  
 10, wx.FONTFAMILY\_DEFAULT, wx.FONTSTYLE\_ITALIC,  
 wx.FONTWEIGHT\_BOLD, False, wx.EmptyString  
 )  
 )  
 bSizerCodeKey.Add(self.TextCodeKey, 1, wx.ALL, 5)  
  
 self.fieldCodeKey = wx.TextCtrl(  
 self, wx.ID\_ANY, wx.EmptyString,  
 wx.DefaultPosition, wx.DefaultSize, 0  
 )  
 self.fieldCodeKey.SetFont(  
 wx.Font(  
 wx.NORMAL\_FONT.GetPointSize(), wx.FONTFAMILY\_DEFAULT,  
 wx.FONTSTYLE\_ITALIC, wx.FONTWEIGHT\_BOLD, False, wx.EmptyString  
 )  
 )  
 bSizerCodeKey.Add(self.fieldCodeKey, 2, wx.ALL, 5)  
  
 bSizerButtonStart.Add(bSizerCodeKey, 1, wx.EXPAND, 5)  
  
 self.buttonStart = wx.Button(  
 self, self.CODE\_BUTTON\_START\_ENCRYPTION, u"ЗАПУСТИТЬ ",  
 wx.DefaultPosition, wx.DefaultSize, 0  
 )  
 self.buttonStart.SetFont(  
 wx.Font(  
 11, wx.FONTFAMILY\_DEFAULT, wx.FONTSTYLE\_ITALIC,  
 wx.FONTWEIGHT\_BOLD, False, wx.EmptyString  
 )  
 )  
 bSizerButtonStart.Add(self.buttonStart, 1, wx.ALL | wx.EXPAND, 5)  
  
 bSizerMain.Add(bSizerButtonStart, 0, wx.EXPAND, 5)  
  
 self.SetSizer(bSizerMain)  
 self.Layout()  
  
 self.Centre(wx.BOTH)  
  
 self.Bind(  
 wx.EVT\_CHOICE,  
 self.DisplayingAdditionalOptions,  
 id=self.CODE\_CHOICE\_SELECT\_METHOD  
 )  
 self.Bind(  
 wx.EVT\_BUTTON,  
 self.open\_source\_file,  
 id=self.CODE\_BUTTON\_OPEN\_SOURCE\_FILE  
 )  
 self.Bind(  
 wx.EVT\_BUTTON,  
 self.start\_encryption,  
 id=self.CODE\_BUTTON\_START\_ENCRYPTION  
 )  
  
 def \_\_del\_\_(self):  
 pass  
  
 @staticmethod  
 def \_\_update\_choice(choice: wx.Choice, list\_values: list):  
 choice.Clear()  
 choice.Append(list\_values)  
 choice.SetSelection(0)  
  
 def DisplayingAdditionalOptions(self, event):  
 if self.ListEncryptionMethod.GetCurrentSelection() == 0:  
 self.\_\_update\_choice(  
 self.ListAdditionalOptions,  
 ["Словарь (случайный)", "Словарь (по порядку)"]  
 )  
 else:  
 self.\_\_update\_choice(self.ListAdditionalOptions, ["-"])  
  
 def open\_source\_file(self, event):  
 with wx.FileDialog(  
 self, "Открыть файл...", wildcard="Текстовые файлы (\*.txt)|\*.txt",  
 style=wx.FD\_OPEN | wx.FD\_FILE\_MUST\_EXIST) as fileDialog:  
 if fileDialog.ShowModal() == wx.ID\_CANCEL:  
 return  
 pathname = fileDialog.GetPath()  
 self.FileToEncryptOutput.Clear()  
 self.FileToEncryptOutput.AppendText(pathname)  
 self.get\_dir\_file(pathname)  
  
 def start\_encryption(self, event):  
 option = self.ListAdditionalOptions.GetCurrentSelection()  
 method = self.ListEncryptionMethod.GetCurrentSelection()  
 if method == 0:  
 if option == 0:  
 self.vigenere\_encryption(mix=True)  
 if option == 1:  
 self.vigenere\_encryption(mix=False)  
 if method == 1:  
 self.caesars\_cipher()  
  
 def caesars\_cipher(self):  
 original\_path = self.FileToEncryptOutput.GetLineText(0)  
 if original\_path:  
 code\_key = self.fieldCodeKey.GetLineText(0)  
 if code\_key:  
 code\_key = int(code\_key)  
 if self.check\_count\_symbol\_in\_file(original\_path):  
 path\_enc = self.join\_file\_path\_parts(original\_path, "encC\_")  
 path\_dec = self.join\_file\_path\_parts(original\_path, "decC\_")  
 cryptor = CaesarsCipher(  
 original\_path,  
 path\_enc,  
 path\_dec,  
 code\_key  
 )  
 self.OutputLog.Clear()  
 self.output\_head\_line(path\_enc)  
 self.output\_head\_line(path\_dec)  
 else:  
 self.msg("Не указан сдвиг для зашифровки!", "Ошибка!")  
 else:  
 self.msg("Не выбран файл для зашифровки!", "Ошибка!")  
  
 def vigenere\_encryption(self, mix=False):  
 original\_path = self.FileToEncryptOutput.GetLineText(0)  
 if original\_path:  
 code\_key = self.fieldCodeKey.GetLineText(0)  
 if code\_key:  
 if self.check\_count\_symbol\_in\_file(original\_path):  
 path\_enc = self.join\_file\_path\_parts(original\_path, "encV\_")  
 path\_dec = self.join\_file\_path\_parts(original\_path, "decV\_")  
 cryptor = VigenereCipher(  
 code\_key,  
 original\_path,  
 path\_enc,  
 path\_dec,  
 mix  
 )  
 self.OutputLog.Clear()  
 self.output\_vigenere\_square(cryptor.LETTERS)  
 self.output\_head\_line(path\_enc)  
 self.output\_head\_line(path\_dec)  
 else:  
 self.msg("Не указано кодовое слово для зашифровки!", "Ошибка!")  
 else:  
 self.msg("Не выбран файл для зашифровки!", "Ошибка!")  
  
 def output\_vigenere\_square(self, m):  
 line = None  
 x = 0  
 y = len(m)  
 self.add\_highlight\_title(  
 "---------------КВАДРАТ ВИЖЕНЕРА----------\n\n"  
 )  
 for i in range(len(m)):  
 line = ""  
 for j in range(x, y):  
 line += m[j]  
 self.OutputLog.WriteText(line + "\n")  
 x -= 1  
 y -= 1  
  
 def join\_file\_path\_parts(self, path, prefix):  
 info\_path = self.get\_dir\_file(path)  
 result\_path = "\\".join(  
 [info\_path["dir"], prefix+info\_path["name"]]  
 )  
 return result\_path  
  
 @staticmethod  
 def get\_dir\_file(path: str):  
 m = path.split("\\")  
 name = m.pop(-1)  
 dir = '\\'.join(m)  
 return {  
 "name": name,  
 "dir": dir  
 }  
  
 def output\_head\_line(self, path):  
 self.add\_highlight\_title(f"-----> {path}\n")  
 with open(path, encoding='utf-8') as file:  
 head = [next(file) for x in range(1)]  
 for line in head:  
 self.OutputLog.WriteText(line + "\n")  
  
 def add\_highlight\_title(self, text):  
 self.OutputLog.BeginBold()  
 self.OutputLog.BeginFontSize(12)  
 self.OutputLog.WriteText(text)  
 self.OutputLog.EndFontSize()  
 self.OutputLog.EndBold()  
  
 def check\_count\_symbol\_in\_file(self, path):  
 file = open(path, "r", encoding='utf-8')  
 data = file.read()  
 if len(data) < 2000:  
 self.msg(  
 "В файле должно быть минимум 2000 символов!",  
 "Ошибка!"  
 )  
 return False  
 return True  
  
 @staticmethod  
 def msg(text, frame\_name):  
 return wx.MessageBox(  
 text, frame\_name, wx.OK | wx.OK\_DEFAULT  
 )

**caesars\_cipher.py**

import string  
  
  
class CaesarsCipher:  
  
 def \_\_init\_\_(self, path\_input, path\_output\_enc, path\_output\_dec, key):  
 self.key = key  
 self.shift\_type = 'left' if key < 0 else "right"  
 self.key = abs(key)  
 self.path\_input = path\_input  
 self.path\_output\_enc = path\_output\_enc  
 self.path\_output\_dec = path\_output\_dec  
 self.encryption(self.path\_input, self.path\_output\_enc)  
 self.encryption(self.path\_output\_enc, self.path\_output\_dec, dec=True)  
  
 def cipher(self, text, key, characters=None, decrypt=False, shift\_type="left"):  
 if characters is None:  
 characters = ""  
 characters += string.ascii\_lowercase  
 characters += string.ascii\_uppercase  
 characters += string.digits  
 characters += 'абвгдеёжзийклмнопрстуфхцчшщъыьэюя'  
 characters += 'АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ'  
 characters += " "  
 if key < 0:  
 print("key cannot be negative")  
 return None  
 n = len(characters)  
 if decrypt == True:  
 key = n - key  
 if shift\_type == "left":  
 key = -key  
 table = str.maketrans(characters, characters[key:] + characters[:key])  
 translated\_text = text.translate(table)  
 return translated\_text  
  
 def encryption(self, r, w, dec=False):  
 # Открываю файл на чтение  
 f = open(r, encoding='utf-8')  
 # Открываю файл на запись  
 w1 = open(w, 'w', encoding='utf-8')  
 # Считываю построчно из файла  
 for line in f:  
 l = line  
 # Удаляю знак переноса строки  
 l = l[:-1] if l.find("\n") != -1 else l  
 # Записываю в файл зашифрованную строку  
 elem = str(self.cipher(l, self.key, decrypt=dec, shift\_type=self.shift\_type))  
 w1.write(elem + '\n')  
 # Закрываю файл на запись  
 w1.close()  
 # Закрываю файл на чтение  
 f.close()

**vigenere\_cipher.py**

from random import shuffle  
  
  
class VigenereCipher:  
 *"""  
 Шифр Виженера модуль шифрования.  
 Принимает на вход 4 параметра  
 1 - кодовое слово,  
 2 - имя файла в котором текст  
 3 -имя файла в который записать шифр  
 4 - словарь(если не указать берётся по умолчанию)  
 """* def \_\_init\_\_(  
 self, code, path\_file\_input,  
 path\_file\_output\_enc, path\_file\_output\_dec, mix  
 ):  
 self.code = code  
 self.tmp\_alphabet = None  
 self.path\_file\_input = path\_file\_input  
 self.path\_file\_output\_enc = path\_file\_output\_enc  
 self.path\_file\_output\_dec = path\_file\_output\_dec  
 self.mix = mix  
 self.reading\_and\_writing\_from\_file(  
 k=self.code, r=self.path\_file\_input,  
 w=self.path\_file\_output\_enc, dec=False,  
 mix=self.mix  
 )  
 self.reading\_and\_writing\_from\_file(  
 k=self.code, r=self.path\_file\_output\_enc,  
 w=self.path\_file\_output\_dec, dec=True  
 )  
 print(self.LETTERS)  
  
 DEFAULT\_LETTERS = 'АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ '  
 # Словарь по умолчанию( Массив с набором используемых символов)  
  
 LETTERS = ""  
  
 def mix\_alphabet(self):  
 words = self.DEFAULT\_LETTERS.split()  
 for i, word in enumerate(map(list, words)):  
 shuffle(word)  
 words[i] = ''.join(word)  
 return str(\*words)  
  
 def encryptMessage(self, key, message, mix):  
 return self.translateMessage(key, message, 'encrypt', mix)  
  
 def decryptMessage(self, key, message):  
 return self.translateMessage(key, message, 'decrypt', None, self.LETTERS)  
  
 def translateMessage(self, key, message, mode, mix, alphabet=None):  
 translated = [] # хранит зашифрованное /дешифрованное сообщение  
 key\_index = 0  
 key = key.upper()  
 if alphabet:  
 self.LETTERS = alphabet  
 else:  
 if mode == 'encrypt':  
 self.LETTERS = self.DEFAULT\_LETTERS if not mix else self.mix\_alphabet()  
 if mode == 'decrypt':  
 if self.LETTERS is None or self.LETTERS == "":  
 self.LETTERS = self.DEFAULT\_LETTERS  
 for symbol in message: # цикл по символам строки message  
 num = self.LETTERS.find(symbol.upper())  
 if num != -1: # -1 means symbol.upper() найден в строке LETTERS.  
 if mode == 'encrypt':  
 num += self.LETTERS.find(key[key\_index]) # Добавить в случае шифрования  
 elif mode == 'decrypt':  
 num -= self.LETTERS.find(key[key\_index]) # Вычесть в случае дешифрования  
 num %= len(self.LETTERS) # обработка завертывания  
 # Добавить зашифрованный/дешифрованный символ в конец  
 if symbol.isupper():  
 translated.append(self.LETTERS[num])  
 elif symbol.islower():  
 translated.append(self.LETTERS[num].lower())  
 key\_index += 1 # перейти к следующей букве ключа  
 if key\_index == len(key):  
 key\_index = 0  
 else:  
 # Присоединить символ без шифрования/дешифрования  
 translated.append(symbol)  
 return ''.join(translated)  
  
 # Функция считывания, записи и шифрования  
 # 1- кодовое слово, 2- файл прочитать, 3- файл куда записать 4 - шифр\дешифр, 5 -шифров/дешифр  
 def reading\_and\_writing\_from\_file(self, k, r, w, dec, mix=False):  
 # Открываю файл на чтение  
 file\_to\_read = open(r, encoding='utf-8')  
 # Открываю файл на запись  
 file\_for\_writing = open(w, 'w', encoding='utf-8')  
 # Считываю построчно из файла  
 for line in file\_to\_read:  
 l = line  
 # Удаляю знак переноса строки  
 l = l[:-1] if l.find("\n") != -1 else l  
 # Записываю в файл строку  
 if dec:  
 file\_for\_writing.write(str(self.decryptMessage(k, l)) + '\n')  
 else:  
 file\_for\_writing.write(str(self.encryptMessage(k, l, mix) + '\n'))  
 # Закрываю файл на запись  
 file\_for\_writing.close()  
 # Закрываю файл на чтение  
 file\_to\_read.close()