Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Фізико-технічний інститут

ЛАБОРАТОРНА РОБОТА №3

з дисципліни

«Криптографія»

на тему:

«Криптоаналіз афінної біграмної підстановки»

Виконали:
студенти 3 курсу ФТІ
групи ФБ-74
Панчук Олександр і Опанасюк Олександр
Перевірили:
Чорний О.
Савчук М.М.
Завадська Л. О.

Київ 2019

Мета роботи :

Набуття навичок частотного аналізу на прикладі розкриття моноалфавітної підстановки; опанування прийомами роботи в модулярній арифметиці.

Порядок виконання роботи:

0. Уважно прочитати методичні вказівки до виконання комп'ютерного практикуму.

1. Реалізувати підпрограми із необхідними математичними операціями: обчисленням оберненого елементу за модулем із використанням розширеного алгоритму Евкліда, розв'язуванням лінійних порівнянь. При розв'язуванні порівнянь потрібно коректно обробляти випадок із декількома розв'язками, повертаючи їх усі.

2. За допомогою програми обчислення частот біграм, яка написана в ході виконання комп'ютерного практикуму №1, знайти 5 найчастіших біграм запропонованого шифртексту (за варіантом).

3. Перебрати можливі варіанти співставлення частих біграм мови та частих біграм шифртексту (розглядаючи пари біграм із п'яти найчастіших). Для кожного співставлення знайти можливі кандидати на ключ шляхом розв'язання системи.

4. Для кожного кандидата на ключ дешифрувати шифртекст. Якщо шифртекст не є змістовним текстом російською мовою, відкинути цього кандидата.

5. Повторювати дії 3-4 доти, доки дешифрований текст не буде змістовним.

Результати:

Розшифрування тексту за варіантом 8:

Використавши основи модулярної арифметики, критерій відбору ключів та частотний аналіз, було знайдено таку пару ключів:
a = 17,
b = 94

Щоб відібрати саме цю пару ключів, нам треба було відфільтрувати розшифровані тексти, які мали заборонені біграми, тобто ті, які у ВТ зустрітися не можуть. Для цього ми створили функцію з такими забороненими біграмами: 'аь', 'оь', 'уь', 'еь', 'ыь', 'иь', 'эь', 'яь', 'юь', 'йь', 'йй', 'ьь', 'ыы'.

Найчастіші біграми ШТ і відповідні їм біграми ВТ та частоти надані в табличці:

Найчастіші біграми зашифрованого тексту	Відповідні біграми розшифрованого тексту	Частота
дэ	ст	0.0138702
цэ	но	0.0134228
жц	то	0.0145413
нц	на	0.0127516
оц	ен	0.0123042

Зашифрований текст:

хбтйьнцнюбцвэтйвшлпнрклщяуйычвшлоыезбвацлнэдйвтыяввэшлньзгишньжддэйфжцзбнцребюаддлсучщмшюшрвдлцнжбйэ
юпувукешщейужчвеяшцвыгжэьсаерврлйоцбилтдтщыыыбюеяшюрзбсцащьтшдйшеетюйояцэцсощолшякфнплыдюржоыыйврдтю
илжшьпюнялишбркесшюцмлрвдджшоещхвлявугиржкрцбвдзацлнжцюевлзшонхщюдтюильлтбюбжзэзюжтбфоцэпндпйштцвэрд
нцлнэлщцтйщццэйпахбщчьариртйфьсцуувэанцйзблныиьзявшждзирчигцнхщэщэьттдювжыосчьбрсдэзуигццээяжддэюрэц
уевлтбюбнйфцгеупзшдднкефржцэшйшюбчшцбдэйузожбмббущфшэешэцуожбуффрьвдхбщьзоцгшмлбшлнвцлнзбэщйвчиявхьбу
ижцньзсдкеьнщцкнчедлиннетыоищхюшхдюдйвхвунчндуижщцкннцнйчшсцащьтлщжыинлцыйсншесшкноебятшдлпзыксзйвюп
ьсвэьщжэбяьцлцребяецлцгнцэбясцжцылщхюшжшяцлшуонрлигщэзкнсомиюшцлсцишгецвжоыруыддгшбшдтжлзщяягьцэюшен
тшдлуутхщццэшлсожштйфчесдвэнциэирхигшжбжцмрэяцэжожрэясцжццдпшышсщэшштхвтеэшсцшмбмиыкбвштсдювцешеяш
эоужцежлюртщарыбфещхыклщщцяцчшюпеббутзйвчигщьтгцннйфечвчивефныбдвхврдьнцэяшяоньйрунжиьзфшлфшизнаерд
жлзщэниныймпйбщнзбсцйроссцьвдлцнжбтдчоющиргищхюшфбкноцтыьзсдкеьнщцплйуяшчэшллцжбжцюеявдэаеупмьфшкноц
тыьзсдкеьнщцоиэяхбоеьштыиршибщвеоцыдяоньйрхвдээнбшмшкйщцщзезсхдбрызбмиаемлжцячбууызбрфжцаерннцнфжцае
юббшщньцннзиыыбужшаэштюдсздэпнжчючмэсучщаесцчпжшышьзоцзбрццбцэыгсдпшяцаэацсоеейвцбцэыгамжцяцвлштвеоц
яцреузчпнтбщкцуожбуффрщджцошюбэосцаннйшлдсыблннйливэяузбцэкедэябщцдыхулнспидлщтруялшхйщцщзддйвшэвоцэ
эяфеыхшлявугноошйбнегуйбнеаефржцйшкнлшинэцошятжэьмжцборцсцюбгьлцыйьсбщжкштяшяцфещдддхияулннйнциэщхйб
аеыхацдршибщшэишчйчэявдэаегиюшчшоештздэжшжцэеюкйбоеисболшснзбрцплянэлцшюшейрхлшжцвэлцрннйпвьльтюфжц
гьсцувыдуюыцнфмлщцщзддазцуешцйтщштыбнйюпзбуьсцыроснцмвжкхьклцуыбцнтцвэшещэущхрфнябющузюбмпрцхттзщэцэ
йпнйяоньйршлкноцщнецуожбуффрщдаештздэжпэряугщебнйжцюебщмльщюдтююндлшежылщзгишньмлудыыфэаецнкллцйигв

пттздэсцьвдлцнжбнйфцлезшхьйшпеыхэнзетдяхфплнтрнцмйчэщдцеаеосявьзхьдыхулнгзвепнфеыхыбцэешгнжиьзйшяцсц
тбошильэыбжбзблшюшфйэсыкрдщэуэыггвшладзнжддэжллцнеяучщпбчьфбыцшшхымэюшмшдожедддэжосзжкжсявьзоцдлгщьш
йшыццэкедэяцоогьцдыысухвдэщджоэозбжшзвдэщдыцтбеднбдьжргцьцжебящцйшюеэнгрирщхвэяврэнпзбвешцйэзцецэиш
оебябшдицнвхлцжцьиьзйшсныхюрвофбщнсцрндбяэнцхумудлжцуюжцдыхуеиштюкшэмббрьшжшжцбшмшкйрцозыдэянцчьщцйш
юеыылшыйяцннильэяврредшлбшожупнцйбйбтйлньцашелдбрццэхьгщвлэтлщубжеисоояцтфжцнцеевльэюпзбюбюшьщмлкбье
сдмнялшеддцэюшбойшцшнбяэйуоигцщвезхьфджлэсаешлштчщйвшлчфшлжцяужэгурхгцазыпщнишбшвуцбинцымлвэлнщнтйюб
мпмбщибищэямжшыэаншштрхутылщузйуямябнцйбнцэщйвддтбозаюодиютбрввшышыцтйкшйшцштрщхйбкбйбдбрцозбузбашыв
зцзбоишэйвддпшепнццэефхьвддэгщтылиезхьенпшишгебяяюуебяццэщмэхвмлмдрцозлбрцозмдчэвэцбцлцпышоеьзяяцэкн
ыоошцэкещдшэюпбшдицнцдцеяуюювмлшсжбгнйрщдйдэтюкйвнцкдбщрхооыцщцсцащуыхвыхщдыцюцоцылрцозыдэянцрцвдвэ
аеддьзгтгвмэефоцкшйртщыыыбюебяцооыцжбньожбвфшувсшыдеишэтцзбщцфедэюпвоххбщдбпшюзебжпэрюрирдбтркеюрщд
мнсцлеьшишсщаеезацжедлгноищхюшбфжцгьнцйбтщбишэйвмэйуишыйшщыкзяьзсдкеьнщцплзщбпщнюшшшцняцгеисжоейыдйб
ошфбяонрддтбщциепттзцаыдыбтйтруялшхйщцщзгтдлсодэюпюбюроццдьючбшлжшонаддлсучщщоуюлшдыосжбньжбдэчьубхй
эзжкьтгтюжтбгньцхйщцхищлгнилтдсцащьтзбишзссдыцтйпцчзяуьттздэтбкштйьнубхйжсанрвяшяцлцнбшлжцхьтщвлйоош
оегулшппэцшшхйыцтйоцяолшахлцябнйкевлзолныиюрыбфекюиэефоцкнперддещэяштызбинлррдэюуоизяугианябшэюпщцжб
гебящньщштшэщхйбтысцащьтддыылнщедэжцюештщхцечвдэгтгтюжтбтщюфпфжцжцсцкдфыкеэяибтйзцзбтщцедлдэчэефжцжц
лряузбйотбозщхштявячыхэныбецтйдэуомнцэьштщыыыбюеьшоигцжщпфхьдлябтбхьшлябтбхьчэщцчоибжцрехуюввэщцывар
лщцэнццеисощбувефбтйюбщнзлявйшюеисоошентщжырещхшбзхщлжцплщцяцсцооpеушкдмлтьпебяцэхрщцжшцэбвюкцнуубщ
ыквегуаэюрмэшэюпсцэцышхьаоинебгжбщдспшюшэщшлжцяцдшгшэдгщтбаэбщмлыькбдэюпфьфцщцнеьтянкшжшцэхщюыкеоцкэ
ыбфьйцрэкеяугйнчяятысцорвлхвцнаэьзкношилщиявсхйусцаоыбфйцеяуэглщцэроплкмжцплыдьзоцялябтбйшйбйбошфбяо
нрирщхйбхищлгнилтдвнэеуясцащьтюбяюйшэеддгщоцяолшилжцьэфэйугньцхйщцфцплбуьззбпэзяоцчшннйшлншлвлнцхуфц
ьщщдвехьялвлнцяяуопцжрирсзэзюжтбчилизнщпщпцшеэсзжоындэыкдллцнештдлэзбпцэвеьюйбнегокшцжьсреяяхифрлнтб
щнцэгитьчзыклшялюбкбьцюикшлнюеиссцжшенмбюшхьшрхршщуыхьцыгщвешлишоштржжвеоцяцпшзбйбошфбяонргттздэсцпэ
бщьэфлоцжбшлоцжббшошболшфеяшецлцппюндльллцжбаэыыроцбиляяцбтрщхйбжддэврдбуозиьзтщдлтрирявоуогныищджцмп
нйтбозаювнаопльтбвяшвндьнцплыдюшчшзбфещдзнфещдкцлняоуоилжшябнйохрттбэотьщририрйшозкехьклявйшюебяюбнф
фрнцнйчшкноцтыьзсдкеьнщцздцеэнецуожбуффрщдкноцтыьзсдкеьнщцнцлнтщьсьзхвцедлтьпегоьцннинсцаэптдлтоэевэ
ыбжбтщрхптйвмлпныбхимпывчвмлмдоцфйаешлщцтйвниннетысцьцкуйужцюевэвеццьвирщдчэштюкйвхвдэлнбуйвчфэжцнуо
ювмлвэцэйвтыгтоонннлфыезисупыйвэгьцэгцяцщшчбаевлябоцкннлшеирыбзцощдхкемлйуэяьщяуфшзбсигцкбцэьюцциэтз
кекеыхьздссолцьцлрцэеддвлзщрдюжшэефоцявветцидсщсхэфбпнлябжоерирщнгцкбфевлнйяцгцсщмлкбнлфылшыйзбкдги
явщдцеаылшыйщцуфхвмлчгрдхвтыпгапаздлокрддэинцйплфобжупашывушошцбфевлщдгщибдэюпщнквццрлчшеэяячщыклшлн
еназдлуигцжцюшоеяшхвсшышйштщвэбщвехьдэгцшьадгщщплфжцгьцэмэсуукьнзбщнхбтйьннрсзявуежлзщкецэнцжбнйплых
йбплгктыхйцэыбрладщдмэнзсхдбфцщшгецвюпчэмэгцкбщхгцэеэяшьтйоцдычшяятрявоышэшвэзшеяутзяусхлшйыяввэяшжш
жесшречимпшшилфыхугбябцшрндрыбтрйввехьюидлцбхьхтгщыхйуяютйтрщхацашыбддаеябфзгщыдыпцхлшхбтйьннрчимпбо
илчуюкинцэншпэефоцбшдицнэшышйшжщбвяшрвьзязфрэястлцлншэйуыбцэщэтыюшйэддвэчэирмшелфрщдяущлезхркноцащез
лнэощцяцгцрегуащузйущзэзюшывйэзбфцлезшлщяомьсцлцнбньдллцнеюржжцнгнащщдэяхбцэямябзщбижылщвлибцрлианее
уптыкнинцэупппйшилцрнрявьбябтщчуyптбылзщбвирчофтвэлнаозщвелнтбцхлшсцтбмьжбйэщцжтэятрсшжнтйдьнцомябиз
явущэфыпфзфрэяжшфеэяцбышуоювмлюбтушщхвюпыэвлтбошахлшщцуфешилтхбщьзцепфцеешилчвдлзбтхооцйыэвлтбошахбщ
ьзощмлябдэнеэшхвццзбмшюшьцннадяцгцнназддзбцеддаршлючдладтшжеарупйшцвщдхуупащкемхлшыцлюилузишснтьццре
кеддюшжпсццшозешгнюбмвезыдешзигцпшщцзвжчцеэяоомбжрэялшюшнцьрщлжцхтдльщзяйшбубщдэыбкнжддэщдрвичдэюпйб
жшщцоиявэснцьщсхьзцеюрьсдшвуйвдэяуашжшхьйианыцспгццэоцсцтлйокбдэылйуяусхбщьзчеефоыкеефьюцбюноочбйбтй
лнгигццэбщжлбштбйэсцщдыбвоплытиношюбщвдлньжбйэяцгцмылиосяунцчьшэыбкняцгцтллцнебищлбшдэчшеэишдэрдоцфй
ысьзшэйыбууюфржкрцеозщацощдхичыхоочкыдащаыешилтдяцгцннащыышэсцюввэшэмхлщжкшэефоцчьмпфьбвьзюцмрунсцфщ
мэнзбивэбихььбпэяуштхвцццэдджшоеяшзбщнлццэнзыкщпчщжыэзешдьгщмбтфцеэятянотрнцмйябиедцощдхзняцгцйбрдюш
чшлнвфйвшлнзуцтсоаыфлщинропуветяуоювмлчшпвввеоцкдцецвюпсцзщбицефшщцйшюеьшувьсефижуивлэцзщдэтеезишнбдь
ыхдбнеяшишюнтбхкрддэшшлцишцнтбхкрддэтщчщьзкеезбианкэшлвзирщдыбвоишчйяцгцкдщцугараемцздпшяцинцйыцтйпц
йивэюшкбоебяяомпщнщцыбьвтекебусцлеаелщьзфнплжслцсцбоятюыщлябинфьшшдиюдтюадьзчщщрйвюдтюзщмбтфцешибщэя
тбтбшепттзшжуюпросптлщтблшгнышщцыбьвьлдьжцведлтьпеэяяоюшвдюшчшлнжцйшадэцрфйврдиснцлеосбшвушэщдхушщдд
ншздыбхиоссожцицощжкншйимлгнаэштдцдогннйтбшубврртюдруыйшьнцнэжшщибадмббрьсзщддосжцюевлысупяохбчбаэшт
дцювдссонйсцсибввеэяжцуфзякшйигвунзбщцйшнщыссонйащезяутбэеяячивэюряуууяввриржоибильэжшнчсшсхинщццэыб
юбпкввеоцяцащяуяушлцеяяэцлесшиейурхыылщрдкшкэбщошцэнцжшфеислшэщяябудвэеепфшэаэвэшецввлвзезсхдбнбябие
упнциепфбвщдармлтщчщьзэятыяцйбзхьзшшлцишьидсэчешлфжцжцшиангцчьлиангцйбнещдлтгщдэодьшцефрщхйбяцгцсиае
млжцячбутзюраеебйшсццшозешадзфоцбусхгцвнятюыэцедзбшцкбфевлнйаенцбщукхвюпщцснйрэяшэщищцицащяухьщноцуе
ыгйуццпиьзюшюбдэээоцэбнбикьтррдешппрццдюшреисяотбкшцнвопшлнмцсотщчубитьишадмшюшсиюрешаоишинлгрдхвтыпг
апоцтбозбярцяэщхэнювузтыосжшцэсовыбввеяшнцлебншсбщаеьээбишцнирщхйблняочйувыгжыбуиншлшшеедэсцувэзлннф
бвщдарсшяцщцснйруннбрцфьбвбимэбнашлшоеяшякддьноцйщмлыьрвйэюпоиыбфхбщьзцбщбрцхбщдыбщзсхдбфцюбйбшшоеяу
рхешжеюрирупошодфысуюкшшлцишщцщнтщчщьзэятбьиаеьэчфжцеепфмэаесзжощзфпсщсхирщхйбаебищлгнехдбюбмшэдбяжш
жеуяштмпсиыыезццуожбуффрршлштыбхьтыдхвэщджшхьоигцьибщлианцэшимэррщджцгцаооцуоювмлюбцэгвцеэзкношильлчш
цэезсхсцовсзсхяцлцнбюбгнщврдйбжшфшфкрдэятржыйвуоювмлоцаэвэявэоаесшышыцтйкшрнфьзфбуцрирошчэюрирошябщц
рлжцчвнцнноцшшцэишювузшемлнццещхцеыдсндбхзкнсонйюпуудлябцеираезгсцчпеджцадзфгщуоювмлнцядыбщбнлузяуэя
тыцноцмудлбшхщьзшэбвчуунэщмэяужцэщцнсдсзбвьсцпбрацодичвеоцвдшэанйшхбчбвнзббшоешлхьдэыаинуоювмлсцхииж
оцвэишехдбюбоцыхаещхщоошжчщэхвчианщцфещдвесшкбжоснхьтымлщцтйвнинсцнцазысртжлзщяягьцэюшдыбуфпюбишеэбв
млоыослцкбинсцлцжешеяшзбишыьзбкдшэанзбщнябщцицпиьзчпеддэгщябщцнуцбнлщщьзшэбвунжцгждхйвцедэбщбиьллбчш
озчехучиьэлоыезяукшжзсхдбснйшодьзхвювжыцэвелнфеисйодштйябюиыыезрдсцооoцюеэякнзщцежсбщжжупгьдбтбеэце
млтбюбдбхбхбдхгтшщщецвуоювмлцэйпинзензддоцщноцжблцжцощбуцрщхйбзбжшжцсцэокбйэябцндэыаинжшэрсшшэьссшжц
юевэгитьгзыкчйаэжжшщмнзивэгцйолщйуукянзепнашывщиявзбмшэдыыйбоеэяснтьрдпшнцьщсхкдткдэюпчщмлбшиэыбжбчи
вляялшхймшюшинжбдэюпэюйбтыооянцдьжбдэзщвэрбкьсцтлйокбдэфзяуйутбошилтхбщьзнцадвэшшдийшятбщжцщнсцщцге
биэялшжеезоцрдашжшоцыубщдэыбневццэкедэюплнплрвдлтблнмпшнуоювмлизжоьымлньнэефоцощвлмугщадбямлвбтйгньщ
млхьжрчщонбоинуьйшплхвосзщаещэфльлчшцэлилифькбинлцидаеябцдюрйшбшоешлхьдэыаинвэзгйбошйшштгщсонлсбюбшш
яэьтвещдяшйоившлтбрнсцмбооклябэцошцйньэевцепфешэтеьойбюббшвфжцнеьсбопладтшцйпшепябтттзжцоигщфьыдчэяв
дэвлцнсодтржхбцнчиштюкйвыцснщцюивэыбишжцюещдвнэепюоцейфьчосолещхунжиьзбшрвщхацлшчвзнсцтлябкнеддэжоаэ
ьзкжшэьцлцрццдвэьщжэьшюроцццлнцэдещэцэдещэливэяупэтхрдешцйьсштыбжебиьзьзлплццплнфцплбутбылвлцврррдбш
оешлхьээыафпафсдцэроижфльтьэиряотэщдмшэдяшхмжцжшжцэщэзижлтвлфряусцжшлннцаожоейдноигцжщпфоцвдьзыдсзэж
рриpадэноцынцэшитюзиьзсояотэацлнмшюшлцщниныйсзкнцоилюфщэмдмшюшннбшепсцжцаоаэщдэзмднцлшплыдсзшэюпйшяб
ьилитржыйвуоювмлчэацшшхйзщтзоцштьлллннйгб

Розшифрований текст:

мальчикизаулыбалисьисжаромвзялисьзаделоонирвализолотистыецветыцветычтонаводняютвесьмирпереплескиваютсяслужаекнамощеныеулицытихонькостучатсяв
прозрачныеокнапогребовнезнаютугомонуиудержуивсевокругзаливаютслепящимсверканиемрасплавленногосолнцакаждоелетооницточносцеписрываютсясказалдеду
шкапустьихянепротиввонихсколькостоятгордыекакльвыпосмотришьнанихподольшетакипрожгутутебявглазахдыркуведьпростойцветокможносказатьсорнаятраваник
тоеенезамечаетамыуважаемсчитаемодуванчикблагородноерастениеонинабралиполныемешкиодуванчиковиунеслививнизвпогребвывалилиихизмешковивотьмепогре
баразлилосьсияниевинныйпрессдожидалсяихоткрытыйхолодныйзолотистыйпотоксогрелегодедушкапередвинулпресповернулручкузавертелбыстрейбыстрейипрес
смягкостиснулдобычунувотвоттаксперватонкойструйкойпотомвсещедреeобильнеепобежалпожелобувглиняныекувшинысокпрекрасногожаркогомесяцаемудалипере
бродитьснялипенуиразлиливчистыебутылкиизподкетчупаионивыстроилисьрядаминаполкахпоблескиваявсумракепогребавиноизодуванчиковсамыеэтисловаточноле
тонаязыкевиноизодуванчиковпойманноеизакупоренноевбутылкилетоитеперькогдадугласзналпонастоящемузналчтоонживойчтоонзатемиходитпоземлечтобывидеть
иощущатьмиронпонялещеоднонадочастицувсегочтоонузналчастицуэтогоособенногоднядняс6ораодуванчиковтожезакупоритьисохранитьапотомнастанеттакойзимн
ийянварскийденькогдавалитгустойснегисолнцауженедавнымдавноникто невиделиможетбытьэточудопозабылосьихорошобыегосноваспомнитьвоттогдаонегооткупори
тведьэтолетонепременнобудетлетомнежданныхчудесинадовсеихсберечьигдетоотложитьдлясебячтобыпослевлюбойчаскогдавздумаешьпробратьсянацыпочкахвов
лажныйсумракипротянутьрукуитамрядзарядомбудутстоятьбутылкисвиномизодуванчиковонобудетмягкомерцатьточнораскрывающиесяназарецветыасквозьтонкийсл
ойпылибудетпоблескиватьсолнценынешнегоиюнявзглянисквозьэтовинонахолодныйзимнийденьиснеграстаетизподнегопокажетсятраванадеревьяхоживутптицылист
ваицветысловномириадыбабочекзатрепещутнаветруидажехолодноесероенебостанетголубымвозьмилетоврукуналейлетовбокалвсамыйкрохотныйконечноизкакогот
олькоисделаешьединственныйтерпкийглотокподнесиегокгубамипожиламтвоимвместолютойзимыпобежитжаркоелетотеперьдождевойводыконечноздесьгодитсятоль
кочистейшаяводадальнихозерсладостныеросыбархатныхлуговчтовозносятсяназарекраспахнувшимсянавстречунебесамтамвпрохладныхвысяхонисобиралисьчисто
омытымигроздьямиветермчалихзасотнимильзаряжаяпопутиэлектрическимизарядамиэтаводавобралавкаждуюсвоюкаплюещебольшенебескогдападаладождемназе

млюонавпиталавсебявосточныйветеризападныйисеверныйиюжныйиобратиласьвдождьадождьвэтотчассвященнодействияужестановитсятерпкимвиномдуглассхват
илковшвыбежалводвориглубокопогрузилеговбочононсдождевойводойвотонаводабылаточношелкпрозрачныйголубоватыйшелкеслиееевыпитьонакоснетсягубгорласе
рдцамягкокакласканоковшиполноеведронадоотнестивпогребчтобыводапропиталатамвесьурожайодуванчиковструямиречекигорныхручьевдажебабушкавкакойнибуд
ьфевральскийденькогдабеснуетсязаокномвьюгаислепитвесьмириулюдейзахватываетдыханьедажебабушкатихонькоспуститсявпогребнаверхувбольшомдомебудетк
ашельчиханьехриплыеголосаистонылпростуженнымдетямоченьбольнобудетглотатьаносыунихпокраснеютточновишнивынутыеизналивкивсюдувдомепритаитсяковар
ныймикробитогдаизпогребавозникнетточнобогинялетабабушкапрячатотоподвязанойшальюонапринесетэточчтотовкомнатукаждоголобящегоиразольетдушистоепро
зрачноевпрозрачныестаканыистаканыэтиосушатоднимглоткомлекарствоиныхвременбальзамизсолнечныхлучейипраздноговгустовскогополудняедваслышныйстукк
олестележкисмороженымчтокатитсяпомощеныумулицамшорохсеребристогофейерверкачторассыпаетсявысоковнебеишелестсрезаннойтравыфонтаномбьющейизпо
дкосилкичтодвижетсяполугампомуравьиномуцарствувсеэтовсеводномстаканедадажебабушкакогдаспуститсявзимнийпогребзаиюнемнавернобудетстоятьтамтихоньк
осовсемоднавтайномединениисосвоимсокровеннымсосвоейдушойкакидедушкаипапаидядябертидругиетожесловнобеседуястеньюдавноушедшихднейспикникамист
еплымдождемсзапахомпшеничныхполейижареныхкукурузныхзеренисвежескошенногосенадажебабушкабудетповторятьсноваисноватежечудесныезолотящиесяслов
ачтозвучатсейчаскогдацветыкладутподпресскакбудутихповторятькаждуюзимувсебелыезимывовсевременасноваисноваонибудутслетатьсгубкакулыбкакакнеждан
йсолнечныйзайчиквотьмевиноизодуванчиковвиноизодуванчиковвиноизодуванчиковонприходилинеслышноуходилипочтибесшумнотравапригибаласьираспрямляла
сьвновьонискольziливнизпохолмамточнотениоблаковэтобежалилетниемальчишкидугласотстализаблудилсязадыхаясьотбыстрогобегаоностановилсянакрающоврага
насамойкромкенадпропастьюиоттудананегодохнулохолодомнавостривушиточнооленьонвдругучуялстаруюкакмиропасностьгородраспалсяздесьнадвеполовиныздес
ькончилаьсцивилизацияздесьживетлишьвспухшаяземляежечасносовершаетсямиллионсмертейирожденийиздесьпроторенныеилиещенепроторенныетропытвердят
чтобыстатьмужчинамимальчишкидолжныстранствоватьвсегдавсюжизньстранствоватьдугласобернулсяэтатропаогромнойпыльнойзмеейскользитькледяномудомугде
взолотыелетниеднипрячетсязимаатабежиткраскаленнымпесчанымберегамиюльскогоозераавонтакдеревьямгдемальчишкипрячутсямежлистьевточнотерпкиеещене
зрелыеплодыдикойяблониитамрастутизреютавотэтакперсиковомусадуквиноградникукогороднымгрядамгдедремлютнасолнцеарбузыполосатыесловнокошкитигрово
ймастиэтатропазаросшаякапризнаяизвилистаятянетсякшколеатапрямаякакстрелаксубботнимутренникамгдепоказываютковбойскиефильмывотэтавдольручьякдикой
леснойчащедугласажмурилсяктоскажетгдекончаетсягородиначинаетсялеснаяглушьктоскажетгородврастаетвнееилионапереходитвгородиздавнаинавекисуществуе
тнекаянеуловимаяграньгдеборютсядвесилыиоднанавремяпобеждаетизавладеваетпросекойлощинойлужайкойдеревомкустомбескрайнееморетравицветовплещется
далековполяхвокругодинокихфермалетомзеленыйприбойяростноподступаетксамомугородуночьзаночьючащилугадальниепросторыстекаютпооврагувсеближезахле
стываютгородзапахомводыитравигородсловнопустеетмертвеетивновьуходитвземлюикаждоеутрооврагещеглубжевгрызаетсявгородигрозитпоглотитьгаражиточноды
рявыелодчонкиипожратьдопотопныеавтомобилиоставленныенамилостьдождяиразедаемыержавчиной эйаусквозьтайныоврагаигородаивременимчалисьджонхафич
арливудменэйдугласмедленнодвинулсяпотропинкеконечноеслихочешьпосмотретьнадвесамыеглавныевещикакживетчеловекикакживетприроданадоприйтисюдаков
рагуведьгородвконцеконцоввсеголишьбольшойпотрепанныйбурямикорабльнанемполнонародуивсехлопочутбезусталивычерпываютводуобкалываютржавчинупорой
какаянибудьшлюпкахибаркадетищекораблясмытоенеслышнойбурейвременитонетвмолчаливыхволнахтермитовимуравьеввраспахнутойовражьейпастичтобыощутит
ькакмелькаютькузнечикиишуршатвжаркихтравахточносухаябумагачтобыоглохнутьподпеленойтончайшейпылиинаконецрухнутьградомкамнейипотокомсмолыкакруша
тсятлеющиеуглиокостразажженногогромомисинеймолниейнамигозарившейторжествоволесныхдебрейтаквотзначитчтотянулосюдадугласатайнаявойначеловекасприро
дойизгодавгодчеловекпохищаетчтотоуприродыаприродавновьберетсвоеиникогдагородпонастоящемудоконцанепобеждаетвечноемугрозитбезмолвнаяопасностьонв
ооружилсякосилкойитяпкойогромныминожницамионподрезаетккустыиопрыскиваетядомвредныхбукашекигусеницонупрямоплыветвпередпокаемувелитцивилизацияян
окаждыйдомтогоиглядизахлестнутзеленыеволныисхоронятнавекиакогданибудьслицаземлиисчезнетпоследнийчеловекиегокосилкиисадовыелопатыизеденныержав
чинойрассыплютсявпрахгородчащадомаоврагдугласозаданноммигаетнокакаяжесвязьмежчеловекомиприродойкакпонятьчтозначатонидругдлядругакогдаонопустил
глазапервыйлетнийобрядпозадиодуванчикисобраныизготовленывпрокпораприступатьковторомунодугласзастылинедвижетсясместадугпошлидугголосазатихливда
лекеяживойсказалдугласночтотолкуоniещебольшеживыечемякакжеэтокакжеткаконстоялводиночествеглядянасвоиногиневсилахдвинутьсясместаинаконецпонялвтот
вечердугласвозвращалсядомойизкиновместесродителямиибратомтомомиувиделихвяркоосвещеннойвитринемагазинатеннисныетуфлидугласпоспешноотвелглазан
оегоногиужеощутилиприкосновениепарусиныизаскользилиповоздухубыстрейбыстрейземлязавертеласькзахлопалиполотняныенавесынадвитринамитакойонподнялв
етертаконмчалсяродителиитомшагалинеторопясьамеждунимипятясьзадомшелдугласинесводилглазстеннисныхтуфельтампозадивполуночнойвитринехорошаябыл
акартинасказаламамаагабуркнулдугласстоялиюньдавноминовалотоврямкогданалетопокупаюттакиетуфлилегкиеитихиеточнотеплыйдождьчтошуршитпотротуарам
ужеиюньземляполнапервозданнойсилыивсевокругдвижетсяиростеттраваипосейденьпереливаетсясюдаизлуговомываеттротуарыподступаеткдомамкажетсягородв
отвотчерпнетбортомипокорнопойдетнадноизеленомморетравнеостанетсянивсплесканирябидугласвдругастылточновросвмертвыйасфальтикрасныйкирпичулицын
евсилахтронутьсясместапапвыпалилонвонтамвокнетеннисныетуфлиотецдаженебернулсяазачемтебеновыетуфлискажипожалуйстаможешьтымнеобяснитьнуудаза
темчтовнихчувствуешьсебятакбудтовпервыевэтолетоскинулбашмакиипобежалбосикомпотраветочновзимнюоночьвысунулногиизподтеплоогодеялаиподставилветр
учтодышитхолодомвоткрытоеокноионистынутстынутапотомвтягиваешьихобратнопододеялоионисовсемкаксосулькивтеннисныхтуфляхчувствуешьсебятакбудтовпер
выевэтолетобредешьбосикомполенивомуручьюивпрозрачнойводевидишькактвоиногиступаютподнубудтоонипереломилисьидвижутсячутьвпередитебяпотомучтовед
ьводевсевидитсянетакпапсказалдугласэтооченьтруднообяснитьаа

Код:

main.py

```
from funcs import *
import os, sys

alphabet_str = read_file('alphabet.txt')
alphabet = list(alphabet_str.replace('\n', ''))

aplhabet_list = create_list(alphabet)
true_bigrams = ['ст', 'но', 'то', 'на', 'ен']

cipher_text = read_file('text.txt')
our_popular_bigrams = popular_bigrams(cipher_text, 5)
print('Наши популярные биграммы:')
for i in our_popular_bigrams:
	print(i)
print('--------\n')

all_lang = bi_pairs(true_bigrams)
all_encr = bi_pairs(our_popular_bigrams)
if (all_lang == False) or (all_encr == False):
	print('Error with compile of bigrams')
for i in range(5):
	pass
	#print(str(all_encr))
	#print('Here error with int to bigram')
matched_texts = {}
keys = []
tmp001 = len(all_lang)
tmp002, tmp003 = 0, 0
while (tmp002 < tmp001):
	while (tmp003 < tmp001):
		key = find_key(all_lang[tmp002], all_encr[tmp003], aplhabet_list)
```

```
                if key == False:
                        continue
                for k in key:
                        keys.append(k)
                tmp003 = tmp003 + 1
        tmp002 = tmp002 + 1

keys = list(dict.fromkeys(keys))
for i in range(5):
        pass
        #print('Here ' + keys)
os.system('echo Success 1st подбор')
for i in range(5):
        pass
        #print('Start decripting')
for k in keys:
        deciphered_text = decrypt(cipher_text, k, aplhabet_list)
        if deciphered_text == False:
                continue
                pass
        is_real = nice(deciphered_text)
        if is_real == 1:
                print('Nice text найдено')
                print('Ключ: '  + ' (' + str(k)[2:4] + ', ' + str(k)[8:10] + ') - ' +
str(bi_to_int(k[0],aplhabet_list)) + ' ' + str(bi_to_int(k[1],aplhabet_list)))
                print(deciphered_text)
                matched_texts[k] = deciphered_text;sys.exit()
                for i in range(5):
                        pass
                        #print(deciphered_text)
        else:
                print(str(k)[2:4] + '-' + str(k)[8:10] + ' ' + str(bi_to_int(k[0],
aplhabet_list)) + '-' + str(bi_to_int(k[1], aplhabet_list)))
                print('Текст невменяемый ' + is_real + '\n')
```

funcs.py

```
import os

def read_file(name):
    f = open(name, 'r')
    data = f.read()
    f.close
    return data

alphabet_str = read_file('alphabet.txt')
alphabet = list(alphabet_str.replace('\n', ''))



def create_list(mas):
    list = {}
    for i in mas:
        list[i] = mas.index(i)
    return list

def create_list2(mas):
    list = {}
    for i in mas:
        list[mas.index(i)] = i
    return list
def bi_to_int(given_bigrams, aplhabet_list):
    bigrams_list = list(given_bigrams)
    bi_num = aplhabet_list[bigrams_list[0]] * 31; bi_num = bi_num +
aplhabet_list[bigrams_list[1]]
    return(bi_num)

def int_to_bi(num, aplhabet_list, alphabet):
    go = create_list2(alphabet); a = len(aplhabet_list)
    bigram1 = num // a
    bigram2 = num % a
```

```
    return go[bigram1] + go[bigram2]

def sort_data(suma):
    return sorted(suma.items(), key=lambda kv: kv[1], reverse=True)

def popular_bigrams(given_text, q):
    suma = {}
    if suma == {'a' : '1'}:
        return "Error!"
    else:
        pass
    wtf = len(given_text)
    for index in range(wtf - 1):
        max = index + 2
        bi = given_text[index:max]
        if (max == index + 2):
            pass
        try:
                suma[bi] += 1
        except Exception as err:
            if (str(err) == ''):
                suma[bi] = -1
            suma[bi] = 1
            #print(err)
    #DUO JS
    final_sum = sort_data(suma)
    for i in range(7):
        pass
        #print(9)
    return [big for big, q in final_sum[:q]]

def nice(txt):
    bad_bigrams = ['оь', 'уь', 'аь', 'ыь', 'эь', 'яь', 'йь', 'иь', 'ыы', 'ьь', 'еь',
'йй']
    bigram = bad_bigrams[0]
    while (bad_bigrams == 'qw'):
        return 0
    for bigram in bad_bigrams:
        while (bad_bigrams == 'qw'):
            return 0
        if bigram in txt:
            return 'Херовая биграмма: ' + str(bigram)
            break
            b = ''
        else:
            continue
    return 1

def bi_pairs(arr):
    res, i, j, a = [], 0, 0, len(arr)
    for i in range(a):
        for j in range(a):
            if i == j:
                pass
            else:
                res.append((arr[i], arr[j]))
    return res

def gcd(one, two):
        if two == 0:
                return one
        else:
                return gcd(two,one % two); a = 'ok'

def mod_inv(a_a, b_b): #взял отсюда - мб переделать под нас
https://www.geeksforgeeks.org/multiplicative-inverse-under-modulo-m/
        x, y = 0, 1
        u, v = 1, 0
        if (x == 0) and (y == 1):
                pass
        else:
                os.exit()
        m = b_b
```

```
        a1 = a_a
        b1 = b_b ; extra = 1
        while (a_a != 0) and (extra == 1):
                q = b_b // a_a
                r = b_b % a_a
                m = x - u * q
                n = y - v * q
                for ickd in range(5):
                        #print(q)
                        pass
                b_b, a_a, x,y, u,v = a_a,r, u,v, m,n
        gcd = b_b
        if x < 0:
                x += m
        else:
                pass
        if gcd == 1 and gcd != -1:
                return x
        else:
                return False
                pass

def lin_equal(a, b, n):
        pass
        d = gcd(a, n)
        if (d == 0):
                return False
        elif (d < 0):
                return False
        elif d == 1:
                mi = mod_inv(a, n)
                if mi == False:
                        return False
                for i in range(5):
                        pass
                        #print(mi)
                x = (b * mi) % n
                return [x]
        elif d > 1:
                for i in range(5):
                        pass
                        #print(d)
                if b % d != 0:
                        return False
                elif b % d == 0:
                        pass
                        if (b == 96100):
                                pass
                        #print(b % d)
                        res = []
                        for i in range(5):
                                pass
                                #print(mi)
                        a1 = a // d; b1 = b // d; n1 = n // d
                        mi = mod_inv(a1, n1)

                        for i in range(5):
                                pass
                                #print(mi)
                        if mi == False:
                                return False
                        x0 = (b1 * mi) % n1
                        for i in range(d):
                                if (i != -1):
                                        res.append(x0 + i * n1)
                return res

def find_key(theor_bigram_pair, encr_bigram_pair, aplhabet_list):
        key_pairs = []
        x1 = bi_to_int(theor_bigram_pair[0], aplhabet_list)
        x2 = bi_to_int(theor_bigram_pair[1], aplhabet_list)
        for i in range(5):
                pass
```

```
                #print(x1); print(x2)
        y1 = bi_to_int(encr_bigram_pair[0], aplhabet_list)
        y2 = bi_to_int(encr_bigram_pair[1], aplhabet_list)
        for i in range(5):
                pass
                #print(y1); print(y2)
        m = len(aplhabet_list)
        temp = lin_equal((x1 - x2), (y1 - y2), m ** 2)
        if (temp == False):
                pass
                #print('Mistake here. Flag == True')
        if (temp == False):
                return False
        key_arr = []
        a = 0
        for a in temp:
                b = y1 - a * x1
                b = b % m**2
                key_arr.append((a, b))
        b_k = []
        for key in key_arr:
                a = key[0]; b = key[1]
                a_big = int_to_bi(a, aplhabet_list, alphabet)
                if (a_big) == -1:
                        print('Error with int to bigram 1')
                        os.exit
                b_big = int_to_bi(b, aplhabet_list, alphabet)
                if (b_big) == -1:
                        print('Error with int to bigram 2')
                        os.exit
                b_k.append((a_big, b_big))
        return b_k

def decrypt(text, key, aplhabet_list):
        m = 1000 - 39
        try:
                a = bi_to_int(key[0], aplhabet_list)
                b = bi_to_int(key[1], aplhabet_list)
        except:
                print('Error with decipher in bigram to int')
        inv_a = mod_inv(a, m)
        if inv_a == False:
                return False
        for i in range(4):
                pass
                #print(inv_a)
                #чето с флагом не так
        res = ''
        for i in range(0, len(text)-1, 2):
                pass
                y = bi_to_int(text[i:i+2], aplhabet_list)
                x = (y - b) * inv_a
                if (x == -1):
                        print('Error 0. Неправильные x y')
                x = x % m
                res += int_to_bi(x, aplhabet_list, alphabet)
        return res
```