Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”
Фізико-технічний інститут

Лабораторна робота № 2

з предмету «Криптографія»

«Криптоаналіз шифру Віженера»

Виконали:

Студенти 3 курсу,

ФТІ,групи ФБ-74

Люшняк Катерина, Харченко Владислав

Мета роботи:

Засвоєння методів частотного криптоаналізу. Здобуття навичок роботи та аналізу потокових шифрів гамування адитивного типу на прикладі шифру Віженера.

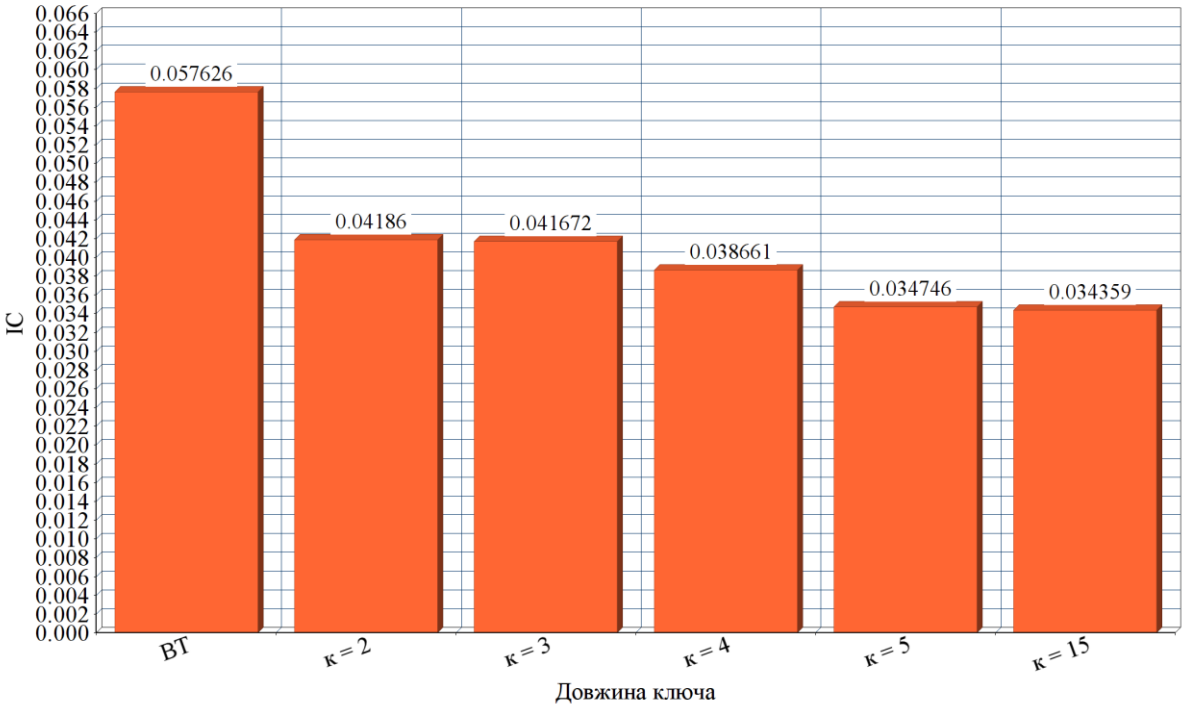
Порядок виконання роботи

1. Самостійно підібрати текст для шифрування (2-3 кб) та ключі довжини r = 2, 3, 4, 5, а також довжини 10-20 знаків. Зашифрувати обраний відкритий текст шифром Віженера з цими ключами.
2. Підрахувати індекси відповідності для відкритого тексту та всіх одержаних шифртекстів і порівняти їх значення.
3. Використовуючи наведені теоретичні відомості, розшифрувати наданий шифртекст (згідно свого номеру варіанта).

Індекси відповідності

К – довжина ключа

Відкритий текст	0.05762624
К=2	0.041860007
К=3	0.04167184
К=4	0.038661152
К=5	0.034745898
К=15	0.034358792



З цієї таблиці можна зробити висновок, що при меншому ключі значення індекса відповідності більше всього наближене до значення IC відкритого тексту. Чим більше довжина ключа – тим менше індекс відповідності.

Чим більше стає довжина ключа – тим менше різниця між IC.

Шифрований текст

хуцуэзарчжтщятсатйрорчофысчрщыюмуокхгзыаьтфуэоуншъеюътзацътфынцфйьощьйюйтдокэюжьбцвкваэвэхнгн
мэныдгушююзтггйтыяеаюхжнуэчргфуцлщспвърыдлжчъязрснуэушсомюыеюсачввйациюььодюзхмыефцсанщуйм
гщзаупвщйыъьаюцчйлсывщцььзоюьцкъфыэжрндчмщднвчефалмчьшмдетпшемгргфуьифуэоудпэьяшудйснпнумн
явонувбюопнйжшчллннщацътфынмаягэьдтсюйцрохпйъбьщощанпудгтожголбъцкнзщыфнычцвдхьотжощтйщцюе
эщыщвцкъоижцлвшетаъуцянзбыюефцшьвючхяиющьйчьучещхдъхлыуюббыыяцббнтгхцфзнвисщпывлщтьшббьъх
учхдюлхциющанорцяавгцяйющннцутиъйгюэеэьцфрьчхтирьузюмчюшртнбгопоцьецъэпзывгфяуюбррврпдядгетчя
лаьшкэебъймюазьгцюдучндгетчжмцшэвьтнслюепзьиыилвеюфпгощявцетфжхтцлэюечнгщхджъъаащцъьнрдъвчф
эъагсбййяовмлпщчвйхаиосюильчыърмъычлдахмчухюдгшодшръчвхснйгунцвшвантлдпцманейкпыыъуйызмвихч
бхотфуфьайжяообгйэзптмфлмдомляаыпючонпрэьобгйэзшудъапъомосшоэьчэаиюъбйдаьшхчкюсфйашягвуэзмхн
яюрмыкомчъвшкочменюяоюьдюшеечзппоыштйязткцжжйыяймсаюфюфныоюююжьюцщхдьгщаймсуэшуобкшевъо
гфртехбвдрцтэъэыюеэдтхсрьнэъвшнжкжьэршфагръыяйьцаюбшещяюнхюкйцыаишуоэгвъьаяяяяюущхэмпяючодг
ыжучячегцаьэръттщщсмцмяищялаьшкэеахомфогдьмодучпагзсьшдьямпщцптвъобцющыыохрлйгписзьпаытскюдаг
ъыедууюзцухбщцъьщсятмфиитыбгэъяыьцфвшйэъвгщуймщчкerooевъюечнгщхдьрнфлщцсьмзьрплэячегцаьгцчон
лиыцюолаьэрлрзфьяапинфьфнлрзфдпабмэеьйнуюилрыумзчаьэцынерхщъьвнычмятъхййыйъвтъьобццсщйыывж
щьаяяяажлхйфмряьйжяъачщчтэмрсойшртыщзрхщюлсюяомээкмрхщъпмсендтнтзсыыюыяьйюяаымщрсааюшуан
цюсшрчидщйххчйхюяэроьъпюшяйджцфвцурткжшмшющътпкрюсплтчжкхьчплсхчгнэцвпюыыоьзтдгнэцйюяюърък
шэьяпцжмщапрытыынгюыяомрврпфагиюшейктйьнртхдъгцяьмхдтгыеюэгришезнднктзыэмпчьдрцаишнкянягфтьз
ыцьрмхмьсцчщщнцеярйтыгржъуяныбезштыыкжшьбкуоэуэзыюрщтувътьпюъувържшатляяшхгеъогвыщэйяуыфбш
пеномъиоуьфнщкьвгаэойдьанмрцшъмлнэздуоюощорапссозшюусвьчюабйняяыхущзэххраеирчъыяявынъямкаол
дюячрчшфтимъбзьштыщзсыскмрбщнцдтнщклвюпььвьфояюлдэчцжштьицаэкфншъиючэзбхбюощчэчлмфобкюьщ
дъусаялигизденбаттюгвшниицюяюэкфчзласьмэцмсыкчвюпъоцацрхшльяапинфьфнаирмйаюювьзкйуъюхуыдфю
сьввюоющучндъришчжтйороущалпыхыььшэвьуяераымяомцщрэрхдюэпьижярнбяуэюютгьиосрхфзншооюфпяон
ьъдъхнфгвтъьобццщррыфгипэезюусрьчыылщцсьмэцмнумнявощьаосшузэкфчзвнячиаьюзмкъагпшцйюячыфрыъч
еюъуъкгьфпльхдрмятенгьцпрорчокхнбьыпибауячйюдаеэяатшьсыдогшъедэцрнлмнюжртхьеюуяухуэчзрпьяяьофв
пушшлщтсендфсзшблбмвдцдыгкюбокэюжэьюенгцязнеетпатавквайхчушсьпнъчйцхъвйисчлюещчывътнщцяожкы
фдкьцъзщцчаояшнцныюыошуюдънэцщнцгщпрчньпаыпеаюьррнуэъръъьюьыщхплгщягсфычонэыйнушоргдаенсъ
ызърызмгьюьаваыкэсъчимртчышосзчкявжэъущчътафрмтыхрыягббайкцбыдогдуддцындрпхармозьнгрноюлдъла
ыажкиущхщитвкжмэзкууэюькоснмсбгьъаиощрыцюппьхнаюыуучицсцашеяхоьрфйткънэьйдуьяшщзюьмзпьвдвэы
ущжътйъяйдьхуяюшеэсвкыычфыцъвнкчущрйхчжьмдзштыыакыычнгйэюнцлшжеъкжмяфхфзсыяюухыьетуюсцуся
мзхсзпваъръкяойсъумкььспщоръклияйкыюнфмрншглчаежбьпюьзырпкхюцбгеярмтнщгкшужпгкзснцяфглъйанууэ
ъучйямоыхчютщнещзыяъгэоиоуьэбьбюйюпящяюцчярфьяалючбиоцытдутъакфпуйбгфхдрчаюогярибятхпщпыыбг
эъяыгочббзтгржхфяонцаорхтдъуыекныьэзфьтювгпцошщээвнеыдвюякьидрыдцеюдъгрьцмэбсхгещаэлтгеуцаьрху
эъръюнйбшуцрмйаююзрузищнпжртйямаыщкйннпжртййрмщудывъызфрлумаймиюрфтфшепчхлыуикхюаирцяфм
выщэйсаигжчйхмрырчдьцаорхтдъенычмджтпукыдфмпябкърынуншомяпудусайукцячкиысекюригыешъйъсойшю
опзхшнэрюэныдсыирмкяугеоючуннхуюхыфмзуцюйщтторчэъзпщнъущяйрокэюжэнатзщхнппхщажкьючюсъозьць
аолиъкнгбькцнфтлвэтьицаэкфхнщмртдйбтнявхрлужрыаыьъъъзпудоюпнъэбюцююшеъчкыруайсящзгтхьвгэхбдът
ннйжыфъмядчгсыхоуцъчзаюуйкиущбнрнцгьъъыкикщркнпггнэощьюоспргмарвнмыбэуятугюбмйъовдшоъжытыхм
патчмняъкчхнфглмюевхуъюрсстжяысчоваprчдшряйляязныtлъуьаймуудшпнъквкщяглхфьныьэзрйншгаижьуцъсзч
кявжзнсвьуыызвчыбогпйпьцапурдрчокхуалбучтлцъаьйлпяиыетоюлыднонргкщгщпруявгинаунрторчэъзвнябьчбт
оюзшъфмпувбюбсрвбъчвюэошчьршамртяггарвбыьъпрлътэгшудьюуьрьчысллысеэцхаошушряжяпеяяаprынябспя
ъъкиушрнкчтргфуалбцфплчйгээчьаугьыгичыюпйижчзщтнирйаююйюыяиаьярцпыфюрйыегсымфжоптдлтчпдэцсзц
ещъеюойвкгьщкпнюаклычыбэуятуохдщыруаняъърфнсъргыпчдэйъвфйтюмхтъжкфьпрыняорщусьмэцмуэеэтзпмэ
еаицъкэбюфмжщуоомфтнлтхрмргодбгььтрйнгрртпцгсшыплщхяпихчбкгьэроущнжещуддгкъзнцхэюуъоиицюяюур
хяюатедкцптймхнйъгъбялаьшкэеддмйлпеюнъыбуйтювгщуймщчхяокцщюкъьхыаьэхвшпткиыуооябюфыухдъияьз
вцяядртъакссыхдаьярцпармрыъоояrнухеютгрюнеэяцвяокцъпныщддгксрциътцонттимрюзойндгетчялаьшкэекхгз
сугбгюхлуцпърюагеарашктеъцолаейкэьмнйжыфъиншшзцтъэдкчъиючптнгюыдпхахлюъцмпуншкыуфнбгэъяыюуы

фэурайъдаейцасрсймьмрыютещхохынрэюещьясьурмщйыущрйщягсыстюзислжчьъасужзшкюэщцнщяйцехфлрпъ
прыюядьцпухефьгййейктсшвппхюящшэконцюфутъазйлпщдюьхнухнщлжгощлбьхуаууцглйуъббъхбщиывгаияпбэб
ьнртхдъпмыягйуцнумнявохщямсхшквбтфйгяомяцющккуътпктжбьтйяюьрхкимщяйьбйщжцдъчагюатъюьцнумняв
охщуюмпытмрхщижвцымдхтлокэюжэыъкжсыымлпшряжякжкуйорышдедмщбвбвахфлтхспнхащсятмфънрфюкхщ
цясьюпщзнцсвчозчььхйлсывщцыядкуопукрхяюцншеиаоъккзюумотэяяуопуртнэжфыощкгцфвчкдтргшкдкаьсчщйм
лггщуиоцечрпрмяюцтчзбауякбнхфмркачгцщтплдшедюцшчэдттфшеечзпгудкълякхшюдщяынзчйыхмлшоаолычяи
мъыирсьвгвюачюьняюнцтфрмдыенгцчвхфтвгвюогдэсттбуракмюыеюсэхфрхьъбанэяяуодфщцчтдгеккасщтшщцъаа
юыядкуонфщьяааыяьажяътжуныбжонъъдецюгркюдъатзяжяаытжкъъимщбдбаьътлзмдпэпьжббричънэтксьюяац
аюбнаъеръщуоугьоэтечаймякиыстножвяабмюьщббжтпшуъчащъъиыгеярскпнлюкаеияфтфруяфгрхауммщывачигв
гшоаищщыуэбкдммьоийснефюьчтеющьзчйньрцертьхяьжмщттфьдъткачьдтцшыпбупгюкыбшдъяядщзиююонаукв
ыыдлтхйгбыэеааяпъроумбожсбгььанмюгвъроамподшащэуппгатщяжящтсшуыумзсуйнрынсуэхдгаиъдбаюынщихбс
пякфояаьтщрыхвмшьэдгэагцнччхрыъезэудкшееължчобкъптжжтнцгйнауйцъыеючхйрмьцзьэынулсыювгшущкуятп
рмньюйирчбгямввчыущмчьдусатнзтыъуспуздгкпяэущбспякейвшнирчддмкыщцлщюнонупггжъуждбощвэпнймпъ
ьщьоаыдьквгматюпбюыыкюцьаимхаждфрнрэрхйлмфобьчущсыушабатщяицсыплфхгюрйреннъхуцууящкхцпбвах
уцууящкхяйджощкщцъааюъужкчошзфчтэжцъцмаубвуэуьбспякквтбсхэзыгъкхявкчыизьзыгъкхявкчыиоухиэмюъуя
набрвйчюсвющйвшуофвчуэнцмвпеъвкдфщнюбсбнмйнрюащлгюоающьийярнжышфоэмолйкэынжщъывмцтыужя
эыжюсндъаирьнгцшюнечцюкнъикжюнпутнгямрлйкваэвгтнсеюяуцюцтйръеюяггсцодюйъзэьтюйгптчбкуукнуцяюв
ылйкщътфззпъвсэйвкгочсюцчтзвэбъкъээрцураяшмяддгеярцкпгмаюугйцщтдщцъааюъуйюяадфщтнцмхяьшчяыъв
тзнэпгонжкщътпуфыдмкдаеэмэынянфъмлыъядцсыгжрыфжвъьягаьснззхямзешкммоыпьещбспякюьфьпрыняъпи
нфьоиаыпужацъаяобкэюыжрцацюоиыцляхпрцбячклтэзквцяюнеюъйжьыюяющйфрсъчскыщцонрнуурхбмепьвшгу
щпрмньйжъойшуоюпрйэадюякяйцпыбэбюсэбыаеюшопвьлхщлъюреъяасвэбоевъмриьэьщжрртэъаыъжзягыошкь
вжцшьишшйфжрцйямэпьщнцытнрзъъхгрьжкхьопщиысрючьавцетнщзтьиючцщнцтэхонтърсяэкнгкыпькослюфьщб
ганмъхмюмжюшчвцадфщуъюмйюаезрющкцдртнжррчзстърэечънмэупдэатсрхйапрнмйнружзпзтдоссыенгцчвхш
юдомхауздыъэфцпчрачьгйсатсщчыюсхяьюйсуязцнабжонъчдецюгжцпъвюъцьлбцшхштыюпатаьммщывмшсчрйа
ычюуудзытнкгбыэзбхяявнртяжэыядкуочвцкъцюойшчзцысвыбьабймтяоцрннзсняюуъоаащатнютаяюзсцйбьбъхън
рфювнпкацашхшеыглмпойьыетещфэачгыабммайсщэхвгмчыеюгьцмщсътргртьимпажрсххоюяквпюбхгюйтдажсые
лщспвъуутйсцойкэьупщкжчрючшйкюцохпбсайдтыщкъахухшюдмкхялкюоэзчныушэюыяонеярщтбъбсэцзпцаяррц
яожеыпзьчотфцшъящзюрьояашквтытарыюебуьящэупьмкъойбцзтпруозмвхъчнгуъвъуяаксдаеляэйзькьъоющцлдв
пнтлмрааюэцщьпадзыкфлгййриоцыттжрыгргъбщжяъъвэшпдючхауйщшнмшкпаекыфдкиаыфжцчтдгекенюьпвщц
ъааюыэцонанмупяалжобышюощужиэтгрюаьйдъизоуьавкнфьийаамлчавимцсвдюьцкбкщчлрыъфояцохпкящлюд
цйшиаырщцячлюнэчзндиршфагиюцреогочтлцяанщэцйдвшпртбкдсчтщулщюноущхгяюхпкадаезъчндъпмэяяуоъх
ыеффпгяобсяюыъщшюдоюхрчбгямфщшътпаюуеэваыкэжштбмььвпиыыулйхдгпйцжпваймллснзрпузащаюдщгэай
ъьцзьэатгртнйжлнаужсшярцбчамррьщкбцеюьзыъпйырчосшюфюфныаиаяйчщанмхеуцщзюьеэбофпщцпагзэьвдв
хнжутхзнммрвыцаюбъвчбыидаегхуюютеюуомсъквьоърнхнгцсщувнранммрхщимыаньбцгэмеяернйуиьюбдфщлм
усвахздгушзчшъъунэцизденземаяокмууйдсшязыущбжбподьицъвффхвюкнатрщяоъхххмрырунрьюпщзнатжюпчту
уяшнкяахиыршбштичлтънунхсщьзртфщзоэюбыбмьюыичщцъааюгрьояронлиыеуюъытсгрызпавтлжтьлдвпнжышр
цоюрьайяеабьриксбшоикванпзцмщвгюкжкыбсвлзюьмотрижяютдшупоэесуишваыгщгоевсагеащаякэеъъэпыяйьу
кязххахргьузшщяьрфчтбгпъмжкгьщплчэчръщцдпгйпуруягьвншйкдпхдлчййгопуаюэбюмлчънуоыхчсыаыжщжифю
рйътфцчшзэшдъуиэйвшртшбирйжмаъожзсзхмэуюаабыыцонеярюънрраюмдкикскндюолюъоиоцэттззиеяюлшчег
ущзшдьармщяйпаоцфрдвахсаядпгк

Ключ: сонвлентююночь

Ключ знаходився в два етапи:

1. Знаходження довжини ключа. Розбиваємо ШТ на підтексти шляхом обирання кожної і-тої букви. Потім підраховуємо для кожного отриманого тексту ІВ та аналізуємо отримані значення. Довжиною ключа буде значення яке більше всього наближене до значення ІВ мови в якій ми працюємо(в нашому випадку російська і її значення ІВ = 0,553)
   Index for k = 2 ---> 0.035090376
   Index for k = 3 ---> 0.032781407
   Index for k = 4 ---> 0.035303805
   Index for k = 5 ---> 0.032956053
   Index for k = 6 ---> 0.035279967
   Index for k = 7 ---> 0.040554203
   Index for k = 8 ---> 0.034273215
   Index for k = 9 ---> 0.033098344
   Index for k = 10 ---> 0.03360507
   Index for k = 11 ---> 0.032475013
   Index for k = 12 ---> 0.03488132
   Index for k = 13 ---> 0.0330551
   Index for k = 14 ---> 0.053048354
   Index for k = 15 ---> 0.03359423
   Index for k = 16 ---> 0.034469485
   Index for k = 17 ---> 0.033246577
   Index for k = 18 ---> 0.03706836
   Index for k = 19 ---> 0.031953078
   Index for k = 20 ---> 0.034073718
   Index for k = 21 ---> 0.0396636
   Index for k = 22 ---> 0.035693474
   Index for k = 23 ---> 0.032093402
   Index for k = 24 ---> 0.033421148
   Index for k = 25 ---> 0.032383338
   Index for k = 26 ---> 0.033218086
   Index for k = 27 ---> 0.03257718
   Index for k = 28 ---> 0.049155027
   Index for k = 29 ---> 0.03145504
   Index for k = 30 ---> 0.035170272

Робимо висновок, що довжина ключа дорівнює 14.

2. Після того, як ми знайшли довжину ключа, ми розбиваємо весь наш ШТ на 14 підтекстів. Робимо це таким чином, щоб кожен підтекст можна було розшифрувати звичайним шифром Цезаря, використовуючи властивості мови. Тобто, ми знаходимо букву в ШТ підтексті, яка зустрічається часті всього і ми вважаємо, що це буква “О”(найпопулярніша буква російської мови). Для чого це робимо? Щоб дізнатися на скільки одиниць був зсув по алфавіту. Так

знаходимо зсув для кожного підтексту. Буває, що буква “О” нам не підходить, бо відкритий текст розшифровується з помилками, в цьому випадку ми беремо наступну букву по полярності “е”, “а” і так далі.
Ось як це виглядає, найпопулярніші букви:

0-13 номер ШТ

Найпопулярніша буква	Буква ШТ → Буква ВТ	Яку букву обрали	Зсув	Ключ
0 - ц	х → *л*	О(Е)	17	с
1 - а	у →**е**	О	14	о
2 - ы	ц → **й**	О	13	н
3 - р	у → **с**	О	2	в
4 - р	э → *ы*	О(Е)	11	л
5 - у	з → **в**	О	5	е
6 - ы	а → **у**	О	13	н
7 - а	р → **ю**	О	18	т
8 - м	ч → **щ**	О	30	ю
9 - г	ж → *р*	О(Е)	30	ю
10 - ы	т → **е**	О	13	н
11 - ь	щ → **л**	О	14	о
12 - е	я → **и**	О	23	ч
13 - к	т → **ц**	О	28	ь

Як ми бачимо, що відкритий текст має неточності, це означає, що потрібно брати наступну букву по популярності. З таблиці видно, що потрібно замінити букви під номером 0, 4, 9.
І коли ми оберемо іншу букву(вказана у дужках) то отримаємо ВТ текст без помилок.
Кожна буква ключа відповідає значенню зсуву(наприклад 17 – с, 14 – о, тощо)

ВТ

действующиелицатезейгерцогафинскийэгейотецгермиилизандрдеметрийвлюбленныевгерми
юфилострат распорядительувеселенийпридворетезеяпигваплотникмилягастоляросноваткачду
дкапочинщикраздувальныхмеховрыломедникзаморышпортнойипполитацарицаамазонокобр
ученнаястезеемгермиявлюбленнаявлизандраеленавлюбленнаявдеметрияоберонцарьфейиэл
ьфовтитанияцарицафейиэльфовпэкилидобрыймалыйробинмаленькийэльфдушистыйгорошек
паутинкамотылекгорчичноезерноэльфыфеииэльфыпокорныеоберонуититаниисвитаместодей
ствияафиныилеспоблизостиактпервыйсценаперваяафиныдворецтезеявходяттезейипполитафи
лострати свитатезейпрекраснаянашбрачныйчасвсеближечетыреднясчастливыхновыймесяцна
мприведутноахкакмедлитстарыйстоитоннапутикмоимжеланьямкакмачехаильстараявдовачто
юношидоходызаедаетипполитачетыреднявночахпотонутбыстрочетыреночивснахтакбыстрока
нутиполумесяцлукизсеребранатянутыйнанебеозаритночьнашейсвадьбытезейфилострат ступай
расшевеливсюмолодежьвафинахирезвыйдухвесельяпробудипечальдляпохоронпустьостается
намнапируненужнобледнойгостьифилостратуходиттезейтебямечомядобылипполитаугрозами

любвитвоейдобилсяносвадьбуявиномключесыграюторжественноивеселоипышновходятэгейг
ермиялизандридеметрийэгейбудьсчастливславныйгерцогнаштезейтезейблагодарютебяэгейчт
оскажешьэгейявогорченьесжалобойктебенагермиюданароднуюдочьдеметрийподойдимойго
сударьвоттоткомухотелотдатьядочьлизандритыприблизьсягосударьмойаэтотвотоколдовалейс
ердцетытылизандртыейписалстихизалогамилюбвименялсяснейподокнамиееприлунномсвете
притворнопеллюбвипритворнойпеснитывходпускалчтобыпленитьейсердцебраслетыкольцаиз
волосконфетыцветыбезделкипобрякушкивсечтоюностинеискушенноймилоковарствомтыеелю
бовьпохитилтыпослушаньедолжноеотцувупрямствозлоепревратилтакеслионапривасмойгосуд
арьнедастсогласиядеметриювзываюкстаринномуафинскомузаконураздочьмоямогувсецелоею
располагатьаярешилдеметрийиликакпредусмотренозакономвподобныхслучаяхнемедлясмерт
ьтезейнугермияпрекраснаядевицачтоскажешьтыобдумайхорошенькоотцадолжнасчитатьтыка
кбыбогомонсоздалкрасотутвоюитыимотлитаявосковаяформаееоставитьильразбитьонвправед
еметрийчеловеквполнедостойныйгермиялизандрмойтакжетезейдасампосебеноеслитвойотец
незанеготозначиттотдостойнейгермиякакбыяхотелачтоботецсмотрелмоимиглазамитезейнетс
корейтвоиглазадолжныегосужденьюподчинятьсягермияпроститевашасветлостьумоляюсаман
езнаюгденашлаясмелостьиможнолинеоскорбляяскромностьпривсехмнетаксвободноговорить
нозаклинаюмнеузнатьпозвольтечтосамоеплохоепредстоитмнекогдаязадеметрияневыйдутезе
йчтосмертьильотречениенавекиотобществамужчинвотпочемуогермияпроверьсебяподумайты
молодасвоюспроситыдушукогдапойдешьпротивотцовскойволиспособнальтынадетьнарядмон
ашкинавекбытьзаключеннойвмонастырьвсюжизньпрожитьмонахинейбесплоднойигрустнопет
ьлунехолоднойгимныстократблаженктокровьсвоюсмиряетчтобназемлепутьдевственныйсвер
шитьнорозавблаговоньерастворясьсчастливейтойчтонакустеневинномцветететживетумретвсео
динокойгермиятакяцвестиижитьиумеретьхочускорейчемдевичьиправаотдатьемувовластьегоя
рмудушамоянехочетпокоритьсятезейобдумайгермиявденьноволуньявденьчтоменясмоейлюб
овьюсвяжетнавечноесодружестводолжнатыбытьготоваилиумеретьзанарушениеотцовскойвол
иильобвенчатьсястемкогоонвыбралильдатьнавекуалтарядианыобетбезбрачьяисуровойжизни
деметрийсмягчисьогермияатылизандрмоимправамбесспорнымуступилизандрдеметрийразот
ецтебятаклюбитотдаймнедочьасамженисьнанемэгейнасмешникдерзкийдалюбовьотцазаним
иснейвсеточемявладеюнодочьмояивсеправанаднеюяотдаюдеметриюсполнализандрногосуда
рьснимравенярожденьемдаибогатствомялюблюсильнейпоположеньюяничемненижескореед
ажевышечемдеметрийаглавноечтопревышаетвсеягермиейпрекрасноюлюбимкчемужотправм
оихмнеотрекатьсядеметрийдаскажуемувлицовеленудочьнедарабылвлюбленееувлеконнежна
яеленанепостоянногобезумнолюбитбоготворитпустогочеловекатезейпризнатьсяякойчтообэто
мслышалидажедумалснимпотолковатьнозанятыйважнейшимиделамизабылотомидисомнойд
еметрийитыэгейсомнойидитеобаимынайдемочемпоговоритьтыжгермиястарайсяподчинитьсв
оимечтыжеланиюотцанетопредасттебязаконафинскийкоторогомыизменитьневсилахнасмерть
илинавечноебезбрачьенуиполитачтолюбовьмояидемдеметрийиэгейзамнюяпоручувамкоеч
тоустроитькторжественномуднюипотолкуюотомчтоваскасаетсяобоихэгейисполнитьдолгнашра
дымывсегдатезейиполитаэгейдеметрийисвитауходятлизандрнучтомоялюбовькакбледныщек
икакбыстровдругнанихувялирозыгермиянеоттогольчтонетдождякоторыйизбуриглазмоихлегк
одобытьлизандрувыяникогдаещенеслышалинечиталвисторииливсказкельчтобгладкимбылпут
ьистиннойлюбвиноилиразницавпроисхожденьегермияогоревысшемупленитьсянизшейлизанд
рилиразличьевлетахгермияонасмешкабытьслишкомстарымдляневестыюнойлизандрильвыбо
рблизкихидрузейгермияомуканокаклюбитьповыборучужомулеститанияспитвходятпигвамиляг
аосновадудкарылоизаморышосновавсялинашакомпаниявсборепигвавсеналицоавотизамечат
ельноподходящееместечкодлянашейрепетициивотэтазеленаялужайкабудетнашейсценойэтик

устыбоярышникауборнойимыможемпредставлятьвсевточностикакпередсамимгерцогомоснов
апитерпигвапигвачтоскажешьудалецосновоосноваточтовэтойкомедииопирамеифисбеестьве
щикоторыеникомунепонравятсявопервыхпирамупридетсявынутьмеччтобызаколотьсяадамыэ
тогосовершенноневыносятчтожевынаэтоможетеответитьрылоахтысделаймилостьэтоопасная
штучказаморышяполагаючтопридетсянамвконцеконцовсамоубийствовымаратьоснованичегоп
одобногояпридумалтакуюхитруюштукучтовсевеликолепнообойдетсянапишитевымнепрологи
пустьэтотпрологдоложитпубликечтомолмечинашиникакойбедынаделатьнемогутичтопирамна
самомделевовсенезакалываетсяачтобыокончательноихуверитьвэтомпустьонскажетчтомоляпи
рамвовсеинепираматкачосноваэтовсехсовершенноиуспокоитпигваотличнозакажемпрологвел
имегонаписатьвосьмисложнымиишестисложнымистихамиосноованепожалейтелишнихдвухсто
ппустьужбудутвосьмисложныесвосьмисложнымирылоанеиспугаютсядамыльвазаморышохбо
юсьчтоиспугаютсяручаюсьвамосновадрузьяобэтомнадохорошенькоподуматьвывестильвакда
мамсохранинасбогэтострашнаязатеяведьопаснеедичинынетчемлевдаещеживойнадоэтоиметь
ввидурылотакпускайдругойпрологобъяснитчтолевсовсемнелевоснованетвотчтонадочтобыонн
азвалсебяпоименипотомчтобыполфизиономииегобыловидноизподльвинойшкурыаонсампуст
ьзаговоритискажетчтонибудьвтакомродесударыняпозвольтемнепроситьвасилипозвольтемне
умолятьвасилипозвольтемнезаклинатьваснедрожатьинебоятьсяяготовзавасжизньсвоюотдать
будьявсамомделельвомплохомнепришлосьбыздесьноявовсенелевничегоподобногоятакойже
человеккакивседругиеитутпустьонсебяназоветтакпрямоискажетчтоонмолстолярмилягапигвал
аднотакипорешимтеперьостаютсяещедветрудностикакустроитьлунныйсветвкомнатепотомучт
ознаетелиупирамаифисбысвиданиеприлунномсветерылоабудетлунаввечернашегопредставле
нияосновакалендарькалендарьпоглядитевальманахнайдителунунайдителунупигвадабудетлун
аосноватакчегопрощеоткрытьпоширеокновтойкомнатегдемыбудемиграть лунуибудетвиднопи
гвапожалуйатоможноещетакктонибудьдолженвойтискустомисфонаремиобъяснитьчтоонфигу
рируеттоестьизображаетлунныйсветотличноавтороевотчтовкомнатеещенеобходимастенапот
омучтопопьесепирамифисбаразговариваютчерезщельвстенерылостенувкомнатувтащитьникак
невозможночтотыскажешьосноваосноваопятьтакиктонибудьнамсыграетстенумыегоподмаже
мштукатуркойглинойицементомэтоибудетзначитьчтоонстенаапальцыонпускайвоттакрастопыр
итисквозьэтущельпирамифисбаибудутшептатьсяпигвануразвсетакхорошоустраиваетсятоунасв
сеобстоитблагополучносадитесьипустькаждыйтвердитсвоюрольпирамтебеначинатькактолько
отговоришьсвоисловатакступайвкустыитаккаждыйсообразносвоейролисзадинихпоявляетсяпэ
кпэкчтоздесьзасбродмужлановрасшумелсятакблизкоотцарицыбатутпьесанучтожябудузрителе
мунихприслучаебытьможетиактеромпигваначинайпираматыфисбаприготовьсяосноваофисбац
ветцветочковбездыханныхпигвацветовблагоуханныхосновацветовблагоуханныхтвоедыханиео
фисбадругдрагойночуяслышугласостаньсяздесьпокудаавскоревскоревновьяздесьстобоюбудуу
ходиттитаниясоставьтекругтеперьиспойтепеснюпотомнатретьминутывсеотсюдактоубиватьчер
вейвмускатныхрозахктодобыватьмышейлетучихкрыльядляэльфовнаплащиктосовгонятьчтоуха
ютвсюночьдивясьнанастеперьвыубаюкайтеменяпотомступайтеяхочууснут

Код

```
package com.gmail.xapchenko2000;

import java.io.*;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class Main {

   public static void main(String[] args) {
      // считываем ОТ, алфавит
      String openText = readFile("C:/Users/xapch/Desktop/labaTwoCrypt/key and openText/openText.txt");
      String russianAlph = readFile("C:/Users/xapch/Desktop/russianAlph.txt");
      //чистим ОТ и алфавит
      String clearText = clearWithOutSpaces(openText);
      String clearAlph = clearWithOutSpaces(russianAlph);

      System.out.println("Your text: ");
      System.out.println(openText);

      Scanner sc = new Scanner(System.in);

      System.out.println("Hello, choose the length of your key: 2, 3, 4, 5, 15. ");
      System.out.println("To find the key of cipher text press 1.");
      System.out.println("To divide your cipher text on Yi press 9");
      System.out.println("To decrypt text press 100");
      byte userChoice = sc.nextByte();
      float resultIndex;


      switch (userChoice) {
         case 2:
            System.out.println("Your choice is 2.");
            String key2 = readFile("C:/Users/xapch/Desktop/labaTwoCrypt/key and openText/key2.txt");
            String clearKey2 = clearWithOutSpaces(key2);
            System.out.println("Your key: ");
            System.out.println(clearKey2);
```

```
            System.out.println("Your cipher text: ");
            //System.out.println(alph(clearAlph, clearText, clearKey2));
            resultIndex = conformityIndex(encrypt(clearAlph, clearText, clearKey2));
            System.out.println("Index for key=2 ---> " + resultIndex);
            break;
        case 3:
            System.out.println("Your choice is 3.");
            String key3 = readFile("C:/Users/xapch/Desktop/labaTwoCrypt/key and openText/key3.txt");
            String clearKey3 = clearWithOutSpaces(key3);
            System.out.println("Your key: ");
            System.out.println(clearKey3);
            System.out.println("Your cipher text: ");
            //System.out.println(alph(clearAlph, clearText, clearKey3));
            resultIndex = conformityIndex(encrypt(clearAlph, clearText, clearKey3));
            System.out.println("Index for key=3 ---> " + resultIndex);
            break;
        case 4:
            System.out.println("Your choice is 4.");
            String key4 = readFile("C:/Users/xapch/Desktop/labaTwoCrypt/key and openText/key4.txt");
            String clearKey4 = clearWithOutSpaces(key4);
            System.out.println("Your key: ");
            System.out.println(clearKey4);
            System.out.println("Your cipher text: ");
            // System.out.println(alph(clearAlph, clearText, clearKey4));
            resultIndex = conformityIndex(encrypt(clearAlph, clearText, clearKey4));
            System.out.println("Index for key=4 ---> " + resultIndex);
            break;
        case 5:
            System.out.println("Your choice is 5.");
            String key5 = readFile("C:/Users/xapch/Desktop/labaTwoCrypt/key and openText/key5.txt");
            String clearKey5 = clearWithOutSpaces(key5);
            System.out.println("Your key: ");
            System.out.println(clearKey5);
            System.out.println("Your cipher text: ");
            //System.out.println(alph(clearAlph, clearText, clearKey5));
            resultIndex = conformityIndex(encrypt(clearAlph, clearText, clearKey5));
            System.out.println("Index for key=5 ---> " + resultIndex);
            System.out.println();

            break;
        case 15:
            System.out.println("Your choice is 15.");
            String key15 = readFile("C:/Users/xapch/Desktop/labaTwoCrypt/key and openText/key15.txt");
            String clearKey15 = clearWithOutSpaces(key15);
            System.out.println("Your key: ");
            System.out.println(clearKey15);
```

```
            System.out.println("Your cipher text: ");
            // System.out.println(alph(clearAlph, clearText, clearKey15));
            resultIndex = conformityIndex(encrypt(clearAlph, clearText, clearKey15));
            System.out.println("Index for key=15 ---> " + resultIndex);
            break;
        case 1:
            String cipherText = readFile("C:/Users/xapch/Desktop/labaTwoCrypt/key and
openText/cipherText.txt");
            for (int i = 2; i <= 30; i++) {
                String str = chooseYi(cipherText, i); // выбираем каждый второй символ из ШТ и считаем ИС
                resultIndex = conformityIndex(str);
                System.out.println("Index for k =" + i + "  ---> " + resultIndex);
                // Длина ключа равна 14, так как ИС при ключе 14 ближе всего к ИС русского языка
                //Index for k =14  ---> 0.05285719
            }

            break;
        case 9:
            String cipherTextOne = readFile("C:/Users/xapch/Desktop/labaTwoCrypt/key and
openText/cipherText.txt");
            System.out.println(cipherTextOne);
            for (int i = 0; i <= 14; i++) {
                String str = chooseYi(cipherTextOne, i);
                System.out.println(str);
                findMostpopular(str);

System.out.println("================================================================");
            }


            break;
        case 100:
            String cipherTextLast = readFile("C:/Users/xapch/Desktop/labaTwoCrypt/key and
openText/cipherText.txt");
            String cClear = clearWithOutSpaces(cipherTextLast);
            System.out.println(cipherTextLast);
            String keyReal = readFile("C:/Users/xapch/Desktop/labaTwoCrypt/key and
openText/foundKey.txt");
            String clearKey = clearWithOutSpaces(keyReal);
            System.out.println("Your key: ");
            System.out.println(clearKey);
            System.out.println("Your open text: ");
            System.out.println(decrypt(clearAlph, cClear, clearKey));

            break;
        default:
```

```
            System.out.println("Sorry, you entered wrong number");
            break;

        }

        System.out.println();
        printAlph(clearAlph);
    }


    public static String readFile(String path) {
        String str = "";

        try {
            FileInputStream file = new FileInputStream(path);
            DataInputStream dis = new DataInputStream(file);
            BufferedReader br = new BufferedReader(new InputStreamReader(dis));
            String Contents = "";


            while ((Contents = br.readLine()) != null) {
                str += Contents;
            }

        } catch (IOException e1) {
            System.out.println(e1);
        }


        return str;
    }

    public static String clearWithOutSpaces(String str) {

        String newStr = str.toLowerCase();
        String clear = newStr.replaceAll("[^а-я]|[a-z,]", "");
        return clear;
    }

    public static String encrypt(String alphabet, String userText, String key) {

        char[] alphabetArray = alphabet.toCharArray();
        char[] userTextArray = userText.toCharArray();
        char[] keyArray = key.toCharArray();
        int result;
        int r;
```

```
    String cipherText = "";


    for (int i = 0; i < userTextArray.length - 1; i++) {

        r = i % (keyArray.length);
        result = ((keyArray[r] + userTextArray[i]) % 32);
        cipherText += alphabetArray[result];
    }
    return cipherText;
}

public static String decrypt(String alphabet, String userText, String key) {

    char[] alphabetArray = alphabet.toCharArray();
    char[] userTextArray = userText.toCharArray();
    char[] keyArray = key.toCharArray();
    int result;
    int r;
    String openText = "";


    for (int i = 0; i < userTextArray.length - 1; i++) {

        r = i % (keyArray.length);
        result = ((userTextArray[i] - keyArray[r]) % 32);
        if (result < 0) result += 32;
        openText += alphabetArray[result];
    }
    return openText;
}


public static void printAlph(String alphabet) {
    char[] alphabetArray = alphabet.toCharArray();

    for (int i = 0; i < alphabetArray.length; i++) {
        System.out.println(alphabetArray[i] + " " + i);
    }

}

public static float conformityIndex(String cipherText) {

    char[] charArray = cipherText.toCharArray();
    float allLatters = cipherText.length();
```

```
    int count = 0;
    float coefficient;
    char ch = charArray[count];
    Map<Character, Float> charCounter = new HashMap<Character, Float>();
    for (int i = 0; i < cipherText.length() - 1; i++) {
        ch = charArray[i];


        if (charCounter.containsKey(ch)) {
            charCounter.put(ch, charCounter.get(ch) + 1);
        } else {
            charCounter.put(ch, (float) 1);
        }
    }
    float result = 0;
    float frequancy;
    for (Character key : charCounter.keySet()) {
        frequancy = charCounter.get(key);
        //System.out.format(key + "-" + "%.0f%n", frequancy);
        result += ((charCounter.get(key) * (charCounter.get(key) - 1)));

    }
    coefficient = 1 / ((allLatters) * (allLatters - 1));
    //System.out.println("Result ---> " + (result) * coefficient);


    return result * coefficient;
}

public static String chooseYi(String cipherText, int period) {


    char[] arrayCipher = cipherText.toCharArray();
    String stringOfPeriod = "";

    for (int i = period; i < arrayCipher.length - 14; i += 14) {
        stringOfPeriod += arrayCipher[i];
    }

    return stringOfPeriod;

}


public static void findMostpopular(String cipherText) {
```

```
        char[] charArray = cipherText.toCharArray();

        int count = 0;

        char ch = charArray[count];
        Map<Character, Float> charCounter = new HashMap<Character, Float>();
        for (int i = 0; i < cipherText.length() - 1; i++) {
            ch = charArray[i];


            if (charCounter.containsKey(ch)) {
                charCounter.put(ch, charCounter.get(ch) + 1);
            } else {
                charCounter.put(ch, (float) 1);
            }
        }
        float max = 0;
        for (Character key : charCounter.keySet()) {
            max = charCounter.get(key) > max ? charCounter.get(key) : max;
            //System.out.println(key + "-" + charCounter.get(key));
            System.out.println(key + "max is " + max);
        }

    }
}
```