Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

# Лабораторна робота

## із Криптографії №2

## Криптоаналіз шифру Віженера

Виконали:

Топчій Микита ФБ - 74

Височанська Вікторія ФБ - 71

Перевірено_____

Київ 2019

# КОМП’ЮТЕРНИЙ ПРАКТИКУМ №2
# Криптоаналіз шифру Віженера

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

**Порядок виконання роботи**
0. Уважно прочитати методичні вказівки до виконання комп’ютерного практикуму.
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

| | |
|---|---|
| Відкритий текст | 0.0566293 |
| K = 2 | 0.0409527 |
| K = 3 | 0.0369673 |
| K = 4 | 0.0378617 |
| K = 5 | 0.0350451 |
| K = 19 | 0.0342395 |

Аналізуючи дану таблицю можна дійти висновку, що при збільшенні довжини ключа індекс стає більш віддаленим від значення індексу відповідності відкритого тексту, і навпаки, при малії довжині ключа значення індексу наближається до індексу відкритого тексту.

Шифрований текст:


деьооуцмдурьдыегныьпуыккэаонтцчхлуучктвдджоаубуяцбхкугбэщешряцзывтшщяшхтяяуюйлрънпрйбдшзъняъмйнуьряпьщг
туьзразхтпфгяжитхчьурвттдфанвзгьячрщюитобущузсшзпуыхнмвбуъняэаоемйнеорцбэзмюцйхлыпейжцбышяпщчсущкдбкалт
озьевефямнумвеаыяпцезьуотпъяуччтшкъуньдцплчжурпэнямноннщтвддщоюнгкшцкуьщщдднтютэнцьмухрчбвщгртчпщыупш
оыуфщгхнжхуучехфщвчыыэееъщахььрхобщглэччзекцьэахатггъесюекчпнрщчозьтчмуцгшъмьаьлефуьрраришужыэюваелунэк
ззфвьюррвтецэцжемпвжшралризщцоьпнвхчхньтьцъонntдфаъняэаозжярптлйжцщшрзтоьавхтмечэумэагнюявстуьччцрэезшсцх
зяршуъхтанвчыбгпэзчбаьбхючрлщхткгзгьумыкэецрпшьтьатьдэкякоьпклэябфугэущфщазсфвйгтьръеюаноннъвыанрзбчехуцок
ржзввщцшсъуньтнхквеачцвуэкшоучктвдджлрззесупютпфъцщпюзчдуызиххгуснntцхцньалтйбпяувфкдкмйнеовяпкрртуьищю
рйчрчбгркртяпщсжечзцыцркртяищюрйкпьщчцдчтеопьтункфхбщрыкытуужкыбжпэрстцьцутзркйфвчвыонйуцкмущрббйллтам
упцнечвзюъбтзчелрахчэжьхъэдыбдуоьупкйфюуюпъндрйтзошжьаахзефетлаххпмвеаыъхпчпыомуьъщюнгйшряшццфпкужувлэ
вццшфакзыдвсгхнаркфюыввврадфрсхтьслщржефряасдвчкчкрнкчкщудлбшмгеньйзъвншяудъчесюрткцзбыьщвфежхртрпбзяб
фрйртьжжщзюпрхмчьсткдтжъинмржанчзцнвупцпуфрцдфщдщжьхцьекковщчэалчувэьсхшужбапмрнюхлхпумкшааьсхоокцьд
кбзеэаонеезехазхкмжбщшяхчцтруучпужхчшмшпятръркозябьлррчуцкэрйшгбознняхецщрбчшнрэаанзшиплхвчжщжбдлйшквг
пцмефофхвлютрьтуупкъзшэаргрзечшужехзюбъьйрчусрябыкжцуялжрйксрюхткуццаынфкдкхцлжпньэантзнйкцкехпщщдьщв
юдюахооууфвхтызйзтушуоцхэмуэуйютьышьртлэяътчдзейкцупчполеанрхиыепоикмрлшязфюхеэюмсунршжфжфюйярншвршъ
фиъьжютйнхьончзггчдзтезэусршогуэзсютчзлазнсрбеъхтэкъумйнзбзйючуцрцтзвучрпкыдчыншмчсгцкпшешуфжмшсчръутуб
кбанфмюлочщпэубкбыдзйрхчтпоъчшкббшюмхиежжщдьпчвюовзъкярудрпкйфюаыывуэкмйнумщщиэрсрпбехьпзызцгччюяхцч
лщзишдгбцшюваазщюрччкюбяшнбнвчръпапрлэъятдеапюбънсвююtпътевзларотмвццлррйярьнпетйцбщвзъейуъюкпкйпхтцою
йднчофбмздиапмшсирщррсэяжбхщцхрезршвткгясцлкюцуфщяфуйпббчумхфзкщтрртовшгэпхсвкшцзуфтзщъэыбкшдщпрршщ
юбыьнчйччзязжпффэгхячейбябфрйтюбдщплокkпцпнхгпшхщжрюячущгашфцачгмшъщрунпнужщыгэбшчьрзырцьюдущппрлч
рлуеущхчуэлгххэецодфшужбсшмюойхрррипкходзлюльчпщбчшевкдщалужхлржеьюсееьбюгзумьучжуцбчщстхчюрыккфпфдд

щюйкгшщыдцпешузняынвфыьършзьеавдууднхьейкпщфнцншйзкныккййпэъхнвузарьпфъзгбцлбрршацщжрйпщцанргюдуьакч
пищэтцнинямщыфнфмзчтщмяхцчлщудшфшшыьстнйкцкпууиьъяунъебмнбщкчмфэбыдъхчцрыехпщщдьуипмтпоьдуъкябяьнб
рчйхцозетфщчшзхсяплшгамжвэттявквбьадуыдвызькюзжхузххтдчюттзяуыткрсехзкввщзчтецущвтьтвваьзьаирзядуигщюъхнтч
чпсуучктфюгкжрфздэофбмцзеэкпфрцтрсрженьтгшъвувбхьшнъмзюсыъшичмрапапужудззтпечщюэъсвггчдяыуибварттгзчохну
йчзбцтпэнугчтяяхьюлщрцжхлшяуйрсщааьжньюбуецзжшжуьщзйпдуеъвябущзрезххкяыфхяъуатупхйжужудздчежхрзвркурвэ
щсбпчдцмгумюжбпэнсавупщбтпевтгшнюфзкпункчпвътъпхюьтщцгеыффхвлжакпкшцклпкгбщнныкднуямхедфюгкгыддкоьдц
пешупцдькдчььоттфщюошъщьйуэьпкореэттяэтекуяошьпвшьщвфещищюрйптфъврчрряйыооеьклшгхзщпзшсьмнхклыдлкмтх
нццщеупзсдщнэфеплекзкнчбэщбюояишррхтнжызщнэубушущтшффэдщсрпщцрлфущтщэвлрэуыехомжещщцашдтапрщрудфр
чбвщгртчхюжзттбзээщмхтядщачъсяшяуюючжхлркрщтьхапзыньищрхтепьшзщсппчмлщрцжгпэтхнхкгшпунупуяыьэншмйун
отнгзэбблжфащецывсшнщфчшзыныуаьжнхкеюеазюпчпщчпнлхшпбынъеюкоцоткйьддщбшрязутзртмяугьмггыечбсусжббфрц
ххесььцназдбюэнютеьрыюзшнббфлкбдязьуеумрдыэхяъжтьрсртпдплэщяэежкпоеунклшгхзецезравъшпггаьсюшътраврчведфщ
доуичщюрлчрееойлэнйкцкпуыфрсъьбюнгнвцучшцзеэкпэагньбржъвъшяуюьндеьезчьрчбйэнтуййррошруюбыуехтипщченцпф
ъъкфьувупуибмзшшдьдсдышыырубзбпкюлшчщпцоуцкчфиьлйсадкпъършжватвяыбичновъфвюфнвдуудкнйършклеаррыншнт
рфуърчбьцябцчньшнгбкжпбрсаущелцйьпфцшвэнууьищюрзкфвфнцнсаеьрыюъштвлауцвуэкьррнцкцъцщфрсятжаюкцуцбъчзв
кгтжъкубмфяъшясрьйшьоскнщаохньряырхпепфщятцдэббучщкыпмфэеанаббещыжемкяатьббкмнчйхцьхгуэурсещркщкфпфдб
фюзхиехщрхжкняваънткыкшцацкифюъуфяуыеыуучтдюбябдъудыщркцфнткткцзшфьчыьдетзчбвщбржтрщбпкнреевлмэуьумйм
тшдщаърфюйячзэрткъьяьняачяхлъсхтжщеьюцшчьрйябйпфьяацшчяплшгшофвкдщсюзтцчофхтдфшдънбнынэлфефовюйлцрц
ыхюсротхэччэьвудетйдехрепешдуужуптзхркцудзгюжзарьфтшъщаъкрвущеыцыецкьбйрбшйдуаьжньююатчзбныкэкгахршюсш
дууйдсунутчвчъцнэпдечарржмвваэньшочщящньвядфщдочезлюкюпуюбрщапмчтшьучжааоевзвквдмьбртуршхщзфеэкпцткфф
вгущюыццзщасуокжшшрмрэяшвцфкхббувщгтцьсжяндгкэцэюаюпефщюбйшщбоыннбфячлыпаупфееывхтьзрцыкотьачщгэебу
суячшфлшэщбхььпаьжнмъььфьдуйчзхонуввйыгтнэчяпщъжкъифюфытъемзьменцпфъъйххртъънфткжкудзкхчыеваритобущуэт
кибоьтуопфлъпяьажуьбгжшффяцщлрэднаяскффф эриэккцэювнмвяыущюынибхомжербачюжэевнпбчумзбдднбчьцшнчыюг
гъшдьпкэлаюцфрдбюшнхняиыюфбммфгджоюйщрунпнпожуьуфыгыкфшвплзяыьщбчежуьщзйчкщччьювбждэывйавжпэрсттчэ
ръеуършшгэяыншукаюцйдешхщаюдйцкстуцмвцаннанйбъьуахвжпбурмсчмшомуцкжпгкцшччзшпннкхбсэрпфесщсвчжгщчя
жлгьячзстецржшъняэнщешйщлпнфтйэнвуиплхвчжмфлэкмгрдуьывькнфътхтанвчыбгпэсвцэкгхрдецбпштсвеачтцкиплхврытфд
гэяэущпщчжецэеюсдзхфехктрьчэщкъэяочяюрттнчвьееывхтьзлювмчэйетчаюпвкфцсхшщщштннвфейвцпнхкепшрсххжкццфуь
рцбдщмюньцэкгкхкюбфсзызйучфзйшощццрзпфзупьнлкнхжвхзээйучфзвьвъшцндабжхщакзодщгъчъзеичщркйпньдоуючтчррс
рйшофляйэцнютзцжуцвжушрзбчехунчехвжэацорпичлрпнсвшхъсйшыьржчтдодетьуфчежхрззттлвцвлмшьътучкеямфэужцхиет
ррээхргбгэмюкьцэкдхпжббтьйюхесшьзтпнръсэяъпчпъьиеххэдьлегчочщпзртмвхярзбфершнзчщрееанзютърьбжефвбуьщпюзук
чъвцхзгыллмхфецэывдчрьачцычдиплхвчжщжбужръузучывдрзбдьщияхяццбеншпбшцрпцевнтяфхкчфкдщалудкчызйэофюъпт
атеепутлкнгбфлпрелешоунйужувуйррьчгуучтжщдсэээкйхрхдунрьвацнгсдуоьмуьрдоыхнуйччщяншркяхцрмйнбеацхспнюбчхя
ъцйхкэвчжяжбдлуртчцучучшбябужйэущкшыршодщгъьйакичуртшфкьхгрвфешуыърчкнвчящзвуэкьсншлрюыюнжфумучъпун
рзщъамлтокыойзпнвдоьорруткбгкххюбфлалрчслщзтжмфтдщцэуцюушмхшдфеомъыеифщщришоябшлрмцжеэкртэмдофлкбди
ищщрзшлхоэщерхбупбътшдыуьыъваьйнутныфзктцрпзьчрюдфпщюшхюгхреццйъехкепфтгюъянтшхъчкцъцщфшмвунуынпкщ
яемъыуичыоътшсвчапдпресщыгуйнетьлйсаокоьпкмэмюахяъжткоьпксвдшщлкрщчтлякофвюфншдчешхлщкцждвгнуоючесняа
тшщрхъпдычзкньитезеанлтвхесхьейкопхюррвкпруренцпфъъйалрипюезтлнщччшзсавунцжтшщжбюшнуувгпывднкааттзпзбуэ
ьтширблчцалрчцэювэчвггаэзттчзьуошыхншгэбгкъууефутжжывкгюсиучыропозкчцнткбшъьпеьхдшапзэубучбгахржтшхнюььт
збиээоуеешябзбряуеяевхаызыудплшгаыффгткрмуйаьшвчжрхнсьмхтядьрркцхжтшркюскьюручмхбыялцвудкъьчурзьрдщсгья
чрщюиъзлшгхнууцмжшввьржкчцнткбзябфрйтщяасдвкикмусхпжеетнзвбьйнопкрзбыэьюшзчтурвтещхогэпюуъркыхрыбцдтыя
щниплхврвхжыдргркызлызлппьюгкйрпфчщыкцьтфаящмрмдевцннфпфяящбюиешвцфкхбьддщпшняирьехкцьыярйюкъухьдех
збэачзеерлявзмклуътфюъвунэтнпчфюашмхуыншоуцпужгафбрхьтзшрошпабэщгюощаььмнууажхцъщцъхщъвйчэаывюйрсянъ
ьннахцыцшнзчеищррхтфхугщлшзиечьойпнщцацнбпчпухгубмьфемтсшчушохлппщяацнфецррафхтжкудумючелррэцшмфтгэп
юодекезхчрхганяпзеьырюпмфтапмшсирщррсчзшшфухрегеыъзркжьетхяпхчммцфткбмжюшяпзикъьзчцвябвщрбнацхцзхшоф
агжзеумувуфьэфркдщсрпямлщюкьууцацнбничжъуспчвязлфръчфрюдупреаанявквбшьзмчвюбюьстуипщрврппюуюутэеифщъп
гщтвыщщчыудеуъзттпфиепзакбчщювцъзшыгюпюзтъшосхйищаящръшотжгсковчбхщбъуйуыйзнчвьячшзэайуаьждьсвбукжрт
дуьакзотзцфугшсдунобеютвчъэятуюхщткркурыщъдэамупцффшжхбйрмпцуъщгрьпфгбчэоыдпкэьпеысщютъцгзичньозтажхъэ
плфечщъзюптвятшршкркурукяпфдавяаущецодцпчшушрахряпщрвучсвчгркътъоуямесвядэлгъукржпвдыюауэщпюцинфямнуб
ыоььбюкхтрфпуыфрсъпшдйтщмйзэщббгэыотчфщъкткзжчврбэкъхрезцфкэрдщоюруцэцнупльбялрвевеыоуцфвыюфлсмкгшье
хзыфцбюутскынэунбчрлеанврйцьышрсэзыччюмхкьцэкчшьрдуялфгчехрфкзпфаующцъеячлъфефкщцвючшччпуужачкжуьудъе
шехцхъшчябфэъъзттлхэзкаждсхясебечцвпкгфэъгзэпчсуцдеъбжжяуфсузюьштеыпщаохзьняцьцпкчююыюусрпеоншхцчээетхн
щзбшьыэобфвввщрвукляурэъфюъчъбркэлщкнмжджхнрхъдеььукчкяуапмрнйесуоаынрузщпюэежжцчфшищаъэывнмуьшвмк
нфяяргшхьпэьтэкопхгроюььсюарзысвяяукшьйушоъжпньэанмхльтлакткожшбрпмпчмлщрцжужгтшмлсочщъэйшуьибщпъебч
щызмкощктцзбущкыжзтчрзвеьйррянтркйэффэейбрлдшйэрйърхааьсмзьищфкмчкюуьннюжркщцырччреыдьюъльтдцпкфвюбяы
диеьчпщбцпгурдщсэеишдыэомрггаьпртуэрлфучзьадрпхцезлщрткуцбфьдсайбчьикьопачпняшиплщкйкищыюжрыночщезршд
щэъщсюхтоньдцймзсбщвюйкъщткчммфюагяенифуасуосвюаоньселранглкжпччтфещтщбиксвебвщйркачыцжыкфрвакрэнвеьр
ргцэеюопзакбчщюъецпщэтсдвццушогасвбшхщоюэвеуеуьряпьщмхйьрлуфцйхбугноаущншдкншфеэехзбперзшртппзщящвюз
юйщюпунрьрдщоючесюефуыърцдччмхйьрлуфцйфвкдщмглдушбдуьмлшююмрсщйыбефшпфчамзыуибсупньюсеаонсквнхьдеф
ржбвщвюйчлрцдушгдущусмтьрзхбженвщчшявагйуюзпьрдлтумбфьпэьтэккцдчшяинънчывмтщщдьудфегащфкртжфщчъneузу
гуннърлаангашъшнцжкхкккшэзешмднпцткффвгклахххьчназеьтщюастшцгуэюксмзщюасдбнынэрвхппрэтььвебнчрзкърадъкэйец
цвоучхкцусуппйесьызолзяыьщбьевкшкмнууюгрвдэттоэьщтшзчбътгюскпцуыецкцодлшшрядпоазпщщгъшйгныечйфхпгзсдвс
южтдшусхпощаящоанърлякртгщюъхнтеязлюзтжмзбцшзьцвуньомктффаэяыегегцпешмфътцнббочщрвхппрэтшдяхеьзпэркйфя
есешчукфбгхкфффнцнэкеьрыючшдщдчцнвувбхькмчвяыйэняункцйойчбадбщпшряныбеехкепфщстегцбупетжщеьщбррьтхьсу
эньйчнъбубнфхжуърцоындамярлрдавкжбыьнауопрехжтйгбцятаеэпуэвйкзжатцнсзейшьлхэмщвтвйрпдноржхэевьдщкбччдьбм
умвжусъяыпчмлыкстжщегррвхччщфзцфпьцтчз

Ключ: человеквфутляре

Для знаходження ключа був використаний наступний алгоритм:

1. Шифротекст розбивається на блоки, кожний з яких містить і-ту букву шифротексту. Таким чином при довжині ключа N мы маємо N блоків.
   Для кожного блоку рахуємо значення індексу відповідності.
   Довжину ключа знаходимо перебравши можливі довжини ключа (у нашому випадку від 2 до 30), співставляєто значання ІВ відкритого тексту та для блоків.
   Отримуємо:

   SubStrings CI for r = 2 : 0.0340249
   SubStrings CI for r = 3 : 0.0376912
   SubStrings CI for r = 4 : 0.0340723
   SubStrings CI for r = 5 : 0.0409722
   SubStrings CI for r = 6 : 0.0376291
   SubStrings CI for r = 7 : 0.0340024
   SubStrings CI for r = 8 : 0.0341933
   SubStrings CI for r = 9 : 0.0375055
   SubStrings CI for r = 10 : 0.041037
   SubStrings CI for r = 11 : 0.0340853
   SubStrings CI for r = 12 : 0.0376715
   SubStrings CI for r = 13 : 0.0340435
   SubStrings CI for r = 14 : 0.0338987
   **SubStrings CI for r = 15 : 0.0558919**
   SubStrings CI for r = 16 : 0.034057
   SubStrings CI for r = 17 : 0.0341348
   SubStrings CI for r = 18 : 0.0375664
   SubStrings CI for r = 19 : 0.0338761
   SubStrings CI for r = 20 : 0.0413281
   SubStrings CI for r = 21 : 0.0376382
   SubStrings CI for r = 22 : 0.0339932
   SubStrings CI for r = 23 : 0.0337858
   SubStrings CI for r = 24 : 0.0376713
   SubStrings CI for r = 25 : 0.0409659
   SubStrings CI for r = 26 : 0.0339034
   SubStrings CI for r = 27 : 0.0371595
   SubStrings CI for r = 28 : 0.0339236
   SubStrings CI for r = 29 : 0.0337997
   SubStrings CI for r = 30 : 0.051044
   Отже, робимо висновок, що довжина ключа дорівнює 15

2. Розбиваємо весь шифротекст на 15 блоків та знаходиму у кожному з блоків найбільш часту літеру. Використовуючи відомі статистичні дані російської мови, віднімаючи від найчастішой літери шифротексту найчастішу літеру російської мови, отрумуємо ключ.

| Номер блока | Найчастіша літера | Розшифрована літера |
|---|---|---|
| 0 | Е | Ч |
| 1 | У | Е |
| 2 | Щ | Л |
| 3 | Ь | О |
| 4 | Р | В |
| 5 | У | Е |
| 6 | П | К |
| 7 | Р | В |
| 8 | Ф | Ф |
| 9 | Б | У |
| 10 | А | Т |
| 11 | Щ | Л |
| 12 | Н | Я |
| 13 | Ю | Р |
| 14 | у | Е |

Розшифрований текст:

насамомкраюселамироносицкоговсараестаростыпрокофиярасположилисьнаночлегзапоздавшиеохотникиихбыло
толькодвоеветеринарныйврачиваниванычиучительгимназиибуркинуиванаиванычабыладовольностраннаядвойна
яфамилиячимшагималайскийкотораясовсемнешлаемуиегововсейгуберниизвалипростопоименииотчествуонжило
кологороданаконскомзаводеиприехалтеперьнаохотучтобыподышатьчистымвоздухомучительжегимназиибуркин
каждоелетогостилуграфовпивэтойместностидавноужебылсвоимчеловекомнеспалииваниванычвысокийхудощав
ыйстариксдлиннымиусамисиделснаружиувходаикурилтрубкуегоосвещалалунабуркинлежалвнутринасенеиегоне
быловидновпотемкахрассказывалиразныеисториимеждупрочимговорилиотомчтоженастаростымавраженщиназд
оровыяинеглупаявовсюсвоюжизньнигденебыладальшесвоегородногоселаникогданевиделанигороданижелезнойд
орогиавпоследниедесятьлетвсесиделазапечьюитолькопоночамвыходиланаулицучтожетутудивительногосказалбу
ркинлюдейодинокихпонатурекоторыекакракотшелышкилиулиткастараютсяуйтивсвоюскорлупунаэтомсветенем
алобытьможеттутявлениеатавизмавозвращениектомувременикогдапредокчеловеканебылещеобщественнымживо
тнымижилодиноковсвоейберлогеаможетбытьэтопростооднаизразновидностейчеловеческогохарактерактознаетя
неестественникинемоеделокасатьсяподобныхвопросовтолькохочусказатьчтотакиелюдикакмавраявлениенередк
оедавотнедалекоискатьмесяцадваназадумерунасвгороденекийбеликовучительгреческогоязыкамойтоварищвыоне
мслышаликонечноонбылзамечателентемчтовсегдадажевоченьхорошуюпогодувыходилвкалошахизонтикоминеп
ременновтепломпальтонаватеизонтикунегобылвчехлеичасывчехлеизсеройзамшиикогдавынималперочинныйнож
чтобыочинитькарандаштоиножунегобылвчехольчикеилицоказалосьтожебыловчехлетаккаконвсевремяпряталего
вподнятыйворотниконносилтемныеочкифуфайкуушизакладывалватойикогдасадилсянаизвозчикатоприказывалп
одниматьверходнимсловомуэтогочеловеканаблюдалосьпостоянноеинепреодолимоестремлениеокружитьсебяобо
лочкойсоздатьсебетаксказатьфутляркоторыйуединилбыегозащитилбыотвнешнихвлиянийдействительностьраздр
ажалаегопугаладержалавпостоянноййтревогеибытьможетдлятогочтобыоправдатьэтусвоюробостьсвоеотвращение
кнастоящемуонвсегдахвалилпрошлоеиточегоникогданебылоидревниеязыкикоторыеонпреподавалбылидлянегов
сущноститежекалошиизонтиккудаонпрялсяотдействительнойжизниокакзвученкакпрекрасенгреческийязыкгов
орилонсосладкимвыражениемикакбывдоказательствосвоихсловприщуривглазиподнявпалецпроизносилантропос
имысльсвоюбеликовтакжестаралсязапрятатьвфутлярдлянегобылияснытолькоциркулярыигазетныестатьивкотор

ыхзапрещалосьчтонибудькогдавциркулярезапрещалосьученикамвыходитьнаулицупоследевятичасоввечераилив
какойнибудьстатьезапрещаласьплотскаялюбовьтоэтобылодлянегоясноопределеннозапрещеноибаставразрешени
ижеипозволениискрывалсядлянеговсегдаэлементсомнительныйчтотонедосказанноеисмутноекогдавгородеразре
шалидраматическийкружокиличительнюиличайнуютоонпокачивалголовойиговорилтихооноконечнотактотаквсе
этопрекраснодакакбычегоневышловсякогороданарушенияуклоненияотступленияотправилприводилиеговуныние
хотяказалосьбыкакоеемуделоесликтоизтоварищейопаздывалнамолебенилидоходилислухиокакойнибудьпроказег
имназистовиливиделиклассную дамупоздновечеромсофицеромтооноченьволновалсяивсеговорилкакбычегоневы
шлоанапедагогическихсоветахонпростоугнеталнассвоеюосторожностьюмнительностьюисвоимичистофутлярны
мисоображенияминасчеттогочтовотдевмужскойиженскойгимназияхмолодежьведетсебядурноченьшумитвкласс
ахахкакбынедошлодоначальстваахкакбычегоневышлоичтоеслибизвторогоклассаисключитьпетроваизчетвертог
оегороватобылобыоченьхорошоичтожесвоимивздохаминытьемсвоимитемнымиочкаминабледноммаленькомлиц
езнаетемаленькомлицекакухорькаондавилнасвсехимыуступалисбавлялипетровуиегоровубаллпоповедениюсажал
иихподарестивконцеконцовисключалиипетроваиегоровабылоунегостранноеобыкновениеходитьпонашимкварти
рампридеткучителюсядетимолчитикакбудточтотовысматриваетпосидитэтакмолчачасдругойиуйдетэтоназывалос
ьунегоподдерживатьдобрыеотношениястоварищамииочевидноходитькнамисидетьбылодлянеготяжелоиходилон
кнамтолькопотомучтосчиталсвоеютоварищескоюобязанностьюмыучителябоялисьегоидажедиректорбоялсявотп
одитеженашиучителянародвсемыслящийглубокопорядочныйвоспитанныйнатургеневеищедринеоднакожеэтотче
ловечекходившийвсегдавкалошахисзонтикомдержалврукахвсюгимназиюцелыхпятнадцатьлетдачтогимназиювес
ьгороднашидамыпосубботамдомашнихспектаклейнеустраивалибоялисьякакбыоннеузналидуховенствостеснялось
принемкушатьскоромноеиграть вкартыподвлияниемтакихлюдейкакбеликовзапоследниедесятьпятнадцатьлетвн
ашемгородесталибоятьсявсегобоятсягромкоговоритьпосылатьписьмазнакомитьсячитатькнигибоятсяпомогатьбе
днымучитьграмотеиваниванычжелаячтотосказатькашлянулносначалазакурилтрубкупогляделналунуипотомужес
казалсрасстановкойдамыслящиепорядочныечитаютищедринаитургеневаразныхтамбоклейипрочееавотподчинил
исьжетерпелитотовотоноиестьбеликовжилвтомжедомегдеияпродолжалбуркинвтомжеэтажедверьпротивдверимы
частовиделисьиязналегодомашнююжизньидоматажеисторияхалатколпакставнизадвижкицелыйрядвсякихзапрещ
енийограниченийиахкакбычегоневышлопостноеестьвредноаскоромноенельзятаккакпожалуйскажутчтобеликовн
еисполняетпостовионелсудаканакоровьеммаслепищанепостнаяноинельзясказатьчтобыскоромнаяженскойприслу
гионнедержализстрахачтобыонемнедумалидурноадержалповараафанасиястарикалетшестидесятинетрезвогоипол
оумногокоторыйкогдатослужилвденщикахиумелкоекакстряпатьэтотафанасийстоялобыкновенноудверискрестив
рукиивсегдаормоталодноитожесглубокимвздохоммногоужихнынчеразвелосьспальняубеликовабыламаленькаят
очноящиккроватьбыласпологомложасьспатьонукрывалсясголовойбыложаркодушновзакрытыедверистучалсявет
ервпечкегуделослышалисьвздохиизкухнивздохизловещиеиемубылострашнопододеяломонбоялсякакбычегоневы
шлокакбыегонезарезалафанасийкакбынезабрались ворыипотомвсюночьвиделтревожныесныаутромкогдамывмес
тешливгимназиюбылскученбледенибыловидночтомноголюднаягимназиявкоторуюоншелбыластрашнапротивнав
семусуществуегоичтоидтирядомсомнойемучеловекупонатуреодинокомубылотяжкоченьужшумятунасвклассахг
оворилонкакбыстараясьотыскатьобъяснениясвоемутяжеломучувствуниначтонепохожеиэтотучительгреческогояз
ыкаэтотчеловеквфутляреможетесебепредставитьедваненженилсяиваниванычбыстрооглянулсявсарайисказалшути
тедаедваненженилсякакэтонистраннонаначиликнамновогоучителяистории игеографиинекоегоковаленкомихаила
саввичаизхохловприехалоннеодинассестройваренькойонмолодойвысокийсмуглыйсгромаднымирукамииполицув
идночтоговоритбасомивсамомделеголоскакизбочкибубубуаонауженемолодаялеттридцатинотожевысокаястройн
аячернобровыакраснощекаяоднимсловомнедевицаамармеладитакаяразбитнаяшумнаявсепоетмалороссийскиером
ансыихохочетчутьчтотакизальетсяголосистымсмехомхахахапервоеосновательноезнакомствосковаленкамиунасп
омнюпроизошлонаименинахудиректорасредисуровыхнапряженноскучныхпедагоговкоторыеинаименинытоходя
тпообязанностивдругвидимноваяафродитавозродиласьизпеныходитподбоченясьхохочетпоетпляшетонаспеласчу
вствомвиютвитрыпотомещеромансиещеивсехнасочаровалавсехдажебеликоваонподселкнейисказалсладкоулыбая
сьмалороссийскийязыксвоеюнежностьюиприятноюзвучностьюнапоминаетдревнегреческийэтопольстилоейиона
сталарассказыватьемусчувствомиубедительночтовгадячскомуездеуneeестьхуторанахуторе живетмамочкаитамта
киегрушитакиедынитакиекабакиухохловтыквыназываютсякабакамиакабакишинкамииварятунихборщскраснень
кимииссиненькимитакойвкусныйтакойвкусныйчтопростоужасслушалимыслушалиивдругвсехнасосенилаоднаит
ажемысльахорошобыихпоженитьтихосказаламнедиректоршамывсепочемутовспомниличтонашбеликовнеженати
намтеперьказалосьстраннымчтомыдосихпоркактонезамечалисовершенноупускалиизвидутакуюважнуюподробно
стьвегожизникаквообщеонотносится кженщинекаконрешаетдлясебяэтотнасущныйвопросраньшеэтонеинтересов
алонасвовсебытьможетмынедопускалидажеимысличточеловеккоторыйвовсякуюпогодуходитвкалошахиспитпод
пологомможетлюбитьемудавноужезасорокаейтридцатьпоясниласвоюмысльдиректоршамнекажетсяонабызанего
пошлачеготольконеделаетсяунасвпровинцииотскукисколькоменужноговздорногоиэтопотомучтосовсемнеделает
сяточтонужнонуводткчемунамвдругпонадобилосьженитьэтогобеликовакоторогодажеивообразитьнельзябыложен
атымдиректоршаинспекторшаивсенашигимназическиедамыожилидажепохорошелиточновдругувиделицельжизн
идиректоршаберетвтеатреложуисмотримвееложесидитваренькасэтакимвееромсияющаясчастливаяирядомснейбе
ликовмаленькийскрюченныйточноегоиздомуклещамивытащилиядаювечеринкуидамытребуютчтобыянепременн

опригласилибеликоваиваренькуоднимсловомзаработаламашинаоказалосьчтоваренькапепрочьбылазамужжитьей убратабылонеоченьтовеселотолькоизналичтопоцелымднямспорилииругалисьвотвамсценаидетковаленкопоулиц евысокийздоровыйверзилаввышитойсорочкечубизподфуражкипадаетналобводнойрукепачкакнигвдругойтолстая суковатаяпалказанимидетсестратожескнигами

## Програмна реалізація:

```
#include<iostream>
#include<string>
#include<map>
#include<set>
#include<fstream>
#include"windows.h"
using namespace std;

map<char, int> CharIntMap = { {'а', 0}, {'б', 1}, {'в', 2}, {'г', 3}, {'д', 4}, {'е', 5}, {'ж', 6}, {'з', 7},
{'и', 8}, {'й', 9}, {'к', 10}, {'л', 11}, {'м', 12}, {'н', 13}, {'о', 14}, {'п', 15}, {'р', 16}, {'с', 17},
{'т', 18}, {'у', 19}, {'ф', 20}, {'х', 21}, {'ц', 22}, {'ч', 23}, {'ш', 24}, {'щ', 25}, {'ъ', 26}, {'ы', 27},
{'ь', 28}, {'э', 29}, {'ю', 30}, {'я', 31} };

map<int, char> IntCharMap = { {0, 'а'}, {1, 'б'}, {2, 'в'}, {3, 'г'}, {4, 'д'}, {5, 'е'}, {6, 'ж'}, {7, 'з'},
{8, 'и'}, {9, 'й'}, {10, 'к'}, {11, 'л'}, {12, 'м'}, {13, 'н'}, {14, 'о'}, {15, 'п'}, {16, 'р'},{17, 'с'},
{18, 'т'}, {19, 'у'}, {20, 'ф'}, {21, 'х'}, {22, 'ц'}, {23, 'ч'}, {24, 'ш'}, {25, 'щ'}, {26, 'ъ'}, {27, 'ы'},
{28, 'ь'}, {29, 'э'}, {30, 'ю'}, {31, 'я'} };

map<int, char> CharFreqMap = { {0, 'о'}, {1, 'е'}, {2, 'а'}, {3, 'и'}, {4, 'н'}, {5, 'т'}, {6, 'с'}, {7, 'р'},
{8, 'в'},{9, 'л'}, {10, 'к'}, {11, 'м'}, {12, 'д'}, {13, 'п'}, {14, 'у'}, {15, 'я'}, {16, 'ы'}, {17, 'ь'},
{18, 'г'}, {19, 'з'}, {20, 'б'}, {21, 'ч'}, {22, 'й'}, {23, 'х'}, {24, 'ж'}, {25, 'ш'}, {26, 'ю'}, {27, 'ц'},
{28, 'щ'}, {29, 'э'}, {30, 'ф'}, {31, 'ъ'} };


set<char> Alphabet = { 'а', 'б', 'в', 'г', 'д', 'е', 'ё','ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о',
                'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч','ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я' };

map<int, string> Keys = { {2, "Ум"}, {3, "Гук"}, {4, "Гром"}, {5, "Секта"}, {19, "Детерминированность"} };

void FilterText(string, set<char>); //Text filter function

void Encrypt(string, string, string, map<char, int>, map<int, char>); //File Encryption function

double FileCI(string); //Counting coincidence index for text from file

double GetStringCI(string); //Get coincidence index of string

double SubStringCI(map<int, string>);    //Counting coincidence index for substrings of text

void SubString(string, map<int, string>&, int); //Get substrings of file function

char GetMaxFreqLetter(string); //Get the most frequent letter from string

void GetKey(string, int); //Get Key function

void Decrypt(string, string);
int main() {
        setlocale(LC_ALL, "rus");
        //FilterText("..\\..\\RawText.txt", Alphabet);
        /*for (auto it = Keys.cbegin(); it != Keys.cend(); it++) {
                Encrypt("..\\..\\FilteredText.txt", "", it->second, CharIntMap, IntCharMap);
        }*/
        //Encrypt("..\\..\\FilteredText.txt", "..\\..\\EncryptedText.txt", "Полиглот", CharIntMap,
IntCharMap);
        cout << "CoincidenceIndex of Raw Text: " << FileCI("..\\..\\FilteredText.txt") << endl;
        for (auto it = Keys.cbegin(); it != Keys.cend(); it++) {
                cout << it->first<< " " << it->second << "       : " << FileCI("..\\..\\EncryptedTextKey" +
to_string(it->first) + ".txt");
                cout << endl;
        }
        map<int, string> SubStringMap;
        for (int i = 2; i < 31; i++) {
                SubString("..\\..\\EncryptedTextVariant18.txt", SubStringMap, i);
                cout << "SubStrings CI for r = " << i << " : " << SubStringCI(SubStringMap) << endl;
                SubStringMap.clear();
        }
```

```
	GetKey("..\\..\\EncryptedTextVariant18.txt", 15);

	Decrypt("..\\..\\EncryptedTextVariant18.txt", "Человеквфутляре");
	system("pause");
	return 0;
}



void FilterText(string FilePath, set<char>Alphabet) {
	ifstream fin(FilePath);
	ofstream fout("..\\..\\FilteredText.txt");
	string buffer;
	if (fin.is_open()) {
		while (fin.peek() != EOF) {
			getline(fin, buffer);
			fin.seekg(fin.tellg());
			//cout << buffer << endl;
			for (int i = 0; i < buffer.length(); i++) {
				if (Alphabet.count(buffer[i]) || Alphabet.count(char(tolower(buffer[i])))) {
					//fout << buffer[i];
					if (buffer[i] == 'ё' || buffer[i] == 'Ё') {
						fout << 'e';
					}
					else {
						fout << char(tolower(buffer[i]));

					};
				}
				else {
					continue;
				}
			}
			buffer.clear();
		}
	}
	fin.close();
	fout.close();
}

void Encrypt(string FilePath, string DestinationPath, string Key, map<char, int> CharIntMap, map<int, char>
IntCharMap) {
	//ifstream fin(FilePath);
	ifstream fin;
	fin.open(FilePath);
	ofstream fout;
	if (DestinationPath.size() == 0) {
		fout.open("..\\..\\EncryptedTextKey" + to_string(Key.length()) + ".txt");
	}
	else {
		fout.open(DestinationPath);
	}
	string buffer;
	int EncNumber;
	char EncChar;
	for (int i = 0; i < Key.length(); i++) {
		if (Key[i] == 'ё' || Key[i] == 'Ё') {
			Key[i] = 'e';
		}
		else Key[i] = char(tolower(Key[i]));
	}
	if (fin.is_open()) {
		while (fin.peek() != EOF) {
			getline(fin, buffer);
			fin.seekg(fin.tellg());
			for (int i = 0; i < buffer.length(); i++) {
				EncNumber = (CharIntMap.find(buffer[i])->second + CharIntMap.find(Key[i %
Key.length()])->second) % CharIntMap.size();
				EncChar = IntCharMap.find(EncNumber)->second;
				fout << EncChar;
			}
```

```
                        buffer.clear();
                }
        }
        else cout << "ifstream Error\n";

}

double FileCI(string FilePath) {
        ifstream fin(FilePath);
        map<char, double> LettersMap;
        string buffer;
        double LettersAmount = 0;
        if (fin.is_open()) {
                while (fin.peek() != EOF) {
                        getline(fin, buffer);
                        fin.seekg(fin.tellg());
                        //cout << buffer << endl;
                        for (int i = 0; i < buffer.length(); i++) {
                                if (LettersMap.count(buffer[i])) {
                                        LettersMap.find(buffer[i])->second++;
                                        LettersAmount++;
                                }
                                else {
                                        LettersMap.emplace(buffer[i], 1);
                                        LettersAmount++;
                                }
                        }
                        buffer.clear();
                }
        }
        else cout << "ifstream Error\n";
        double CoincidenceIndex = 0;
        for (auto it = LettersMap.cbegin(); it != LettersMap.cend(); it++) {
                //cout << it->first << " " << it->second << " " << it->second - 1 << endl;
                CoincidenceIndex += (it->second * (it->second - 1));
                //cout << CoincidenceIndex << endl;
        }
        CoincidenceIndex *= 1 / (LettersAmount * (LettersAmount - 1));
        fin.close();
        return CoincidenceIndex;
}

void SubString(string FilePath, map<int, string> & SubStringMap, int KeyLength) {
        ifstream fin(FilePath);
        string buffer;
        if (fin.is_open()) {
                while (fin.peek() != EOF) {
                        getline(fin, buffer);
                        fin.seekg(fin.tellg());
                        for (int i = 0; i < buffer.length(); i++) {
                                if (SubStringMap.count(i % KeyLength)) {
                                        SubStringMap.at(i % KeyLength) += buffer[i];
                                }
                                else {
                                        string s(1, buffer[i]);
                                        SubStringMap.emplace(i % KeyLength, s);
                                }
                        }
                }
                buffer.clear();
                /*for (auto it = SubStringMap.cbegin(); it != SubStringMap.cend(); it++) {
                        cout << it->first << "  " << it->second << endl;
                }*/
        }
        else cout << "istream Error" << endl;
}

double GetStringCI(string String) {
        map<char, double> LettersMap;
        double LettersAmount = 0;
        for (int i = 0; i < String.length(); i++) {
                if (LettersMap.count(String[i])) {
```

```
                        LettersMap.at(String[i])++;
                        LettersAmount++;
                }
                else {
                        LettersMap.emplace(String[i], 1);
                        LettersAmount++;
                }
        }
        double CoincidenceIndex = 0;
        for (auto it = LettersMap.cbegin(); it != LettersMap.cend(); it++) {
                //cout << it->first << " " << it->second << " " << it->second - 1 << endl;
                CoincidenceIndex += (it->second * (it->second - 1));
                //cout << CoincidenceIndex << endl;
        }
        CoincidenceIndex *= 1 / (LettersAmount * (LettersAmount - 1));
        return CoincidenceIndex;
}

double SubStringCI(map<int, string>SubStringMap) {
        double CoincidenceIndex = 0;
        for (auto it = SubStringMap.cbegin(); it != SubStringMap.cend(); it++) {
                CoincidenceIndex += GetStringCI(it->second);
                //cout << GetStringCI(it->second) << endl;
        }
        return CoincidenceIndex / SubStringMap.size();
}

char GetMaxFreqLetter(string String) {
        map<char, int> LettersMap;
        for (int i = 0; i < String.length(); i++) {
                if (LettersMap.count(String[i])) {
                        LettersMap.at(String[i])++;
                }
                else {
                        LettersMap.emplace(String[i], 1);
                }
        }
        char MaxFreqChar;
        int MaxFreq = 0;
        for (auto it = LettersMap.cbegin(); it != LettersMap.cend(); it++) {
                if (it->second >= MaxFreq) {
                        MaxFreq = it->second;
                        MaxFreqChar = it->first;
                }
                else continue;
        }
        return MaxFreqChar;
}

void GetKey(string FilePath, int KeyLength) {
        map<int, string> SubStringMap;
        SubString(FilePath, SubStringMap, KeyLength);
        string Key;
        char chKi;
        int iKi = 0;
        for (auto it = CharFreqMap.cbegin(); it != CharFreqMap.cend(); it++) {
                for (auto it2 = SubStringMap.cbegin(); it2 != SubStringMap.cend(); it2++) {
                        iKi = 0;
                        iKi = CharIntMap.find(GetMaxFreqLetter(it2->second))->second - CharIntMap.find(it-
>second)->second;
                        cout << "Most frequency letter in " << it2->first << " " << GetMaxFreqLetter(it2-
>second) << endl;
                        if (iKi < 0) {
                                iKi += CharIntMap.size();
                                iKi %= CharIntMap.size();
                                chKi = IntCharMap.find(iKi)->second;
                                Key.push_back(chKi);
                        }
                        else {
                                iKi %= CharIntMap.size();
                                chKi = IntCharMap.find(iKi)->second;
                                Key.push_back(chKi);
```

```
                        }
                }
                cout << Key << endl;
                Key.clear();
        }
}

void Decrypt(string FilePath, string Key) {
        ifstream fin(FilePath);
        ofstream fout("..\\..\\DecryptedTextKey" + to_string(Key.length()) + ".txt");
        for (int i = 0; i < Key.length(); i++) {
                Key[i] = char(tolower(Key[i]));
        }
        string buffer;
        char EncChar;
        if (fin.is_open()) {
                while (fin.peek() != EOF) {
                        getline(fin, buffer);
                        fin.seekg(fin.tellg());
                        //cout << buffer << endl;
                        for (int i = 0; i < buffer.length(); i++) {
                                EncChar = IntCharMap.find((CharIntMap.find(buffer[i])->second +
CharIntMap.size() - Key[i % Key.length()]) % CharIntMap.size())->second;
                                fout << EncChar;
                        }

                }
                buffer.clear();
        }
        else cout << "ifstream Error" << endl;
}
```

## Висновок:

При виконанні даного комп’ютерного практикуму були одержані знання щодо криптоаналізу шифру Віженера. Було вивчено таке поняття як індекс відповідності та методи його використання при криптоаналізі. Було набуто практичних навичок розшифрування тексту зашифрованого шифром Віженера.