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

ЛАБОРАТОРНА РОБОТА №2

З дисципліни «Криптографія»

«Криптоаналіз шифру Віженера»

Виконали:
студенти 3 курсу ФТІ
групи ФБ-73
Дем’яненко Д.
Проноза А.

Перевірив:
Чорний О.

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).

Хід роботи:

1)Прочитали методичні вказівки до виконання лабораторної роботи

2)Обрали вірш Пушкіна "У Лукоморья дуб зеленый", розміром 2 кб

3) Підібрали ключі для зашифрування тексту шифром Віженерв

4) Створили додаток у IntelliJ IDEA для виконання лабораторної роботи

5)Написали код для зашифрування тексту

6)Підрахували індекси відповідності для відкритого тексту иа всіх одержаних шифртекстів. Порівняли їх значення

7) Розшифрували шифртекст за варіантом 7

Ключі:

r = 2: як

r = 3: лес

r = 4: киев

r = 5: осень

r = 10: вольныйкот

r = 11: белорусский

r = 12: революцияроз

r = 13: столетняявойна

r = 14: бордодождливый

r = 15: внешняяразведка

r = 16: немецлюбитмюнхен

r = 17: отличныйутебявкус

r =18: государствобельгия

r =19: коричневорубашечник

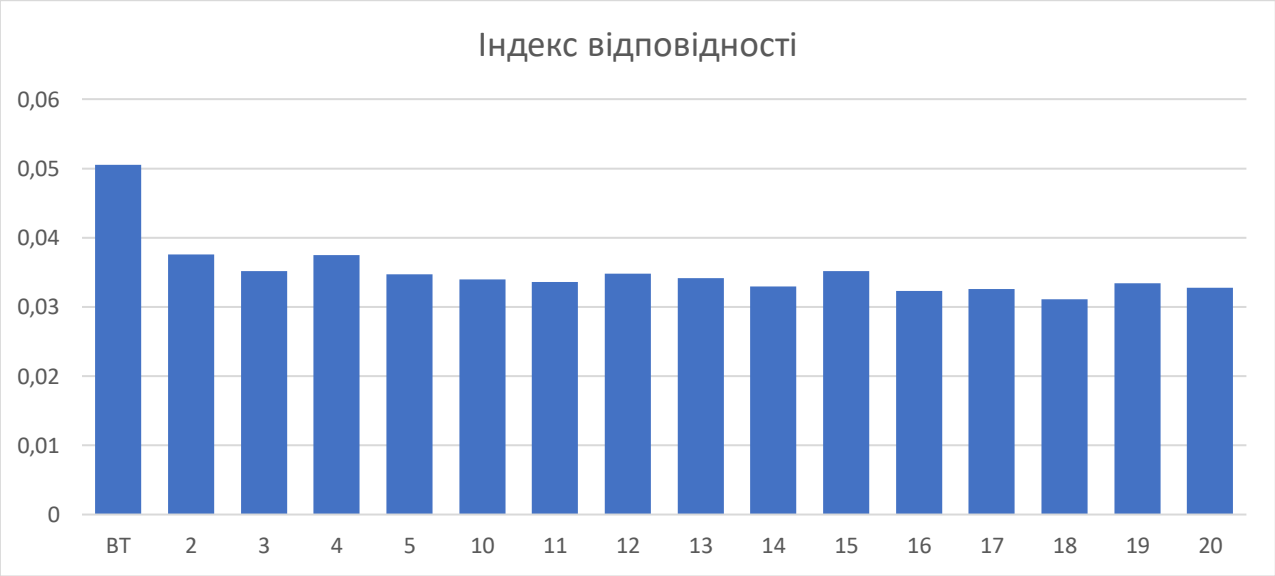
r = 20: иерусалимскаядевочка

Обчислені значення індексів відповідності для значень r

Таблиця:

Довжина ключа	Індекс відповідності
ВТ	0.0505033
2	0.0375952
3	0.0351171
4	0.0375178
5	0.0346973
10	0.0339392
11	0.0336172
12	0.0348237
13	0.0341878
14	0.0329570
15	0.0351171
16	0.0323252
17	0.0325616
18	0.0310699
19	0.0333890
20	0.0327776

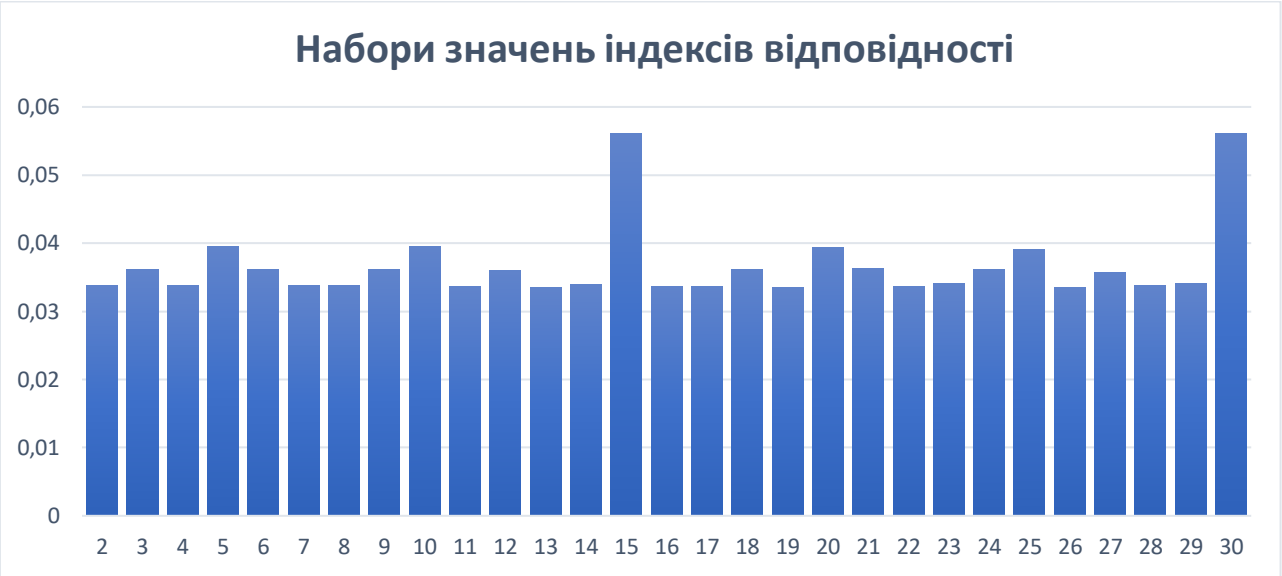
Діаграма:



Варіант 7

Ключ: АРУДАЗОВАРХИМАГ

Діаграма



Зашифрований текст

пабылхэбтэхмвахьфаййпяфаарсроппюдцецупнювигаооцыжащкуоагтчехвэшрнпшфоэъофлтоэухтхныеьипмэх
отгймжыпсьъхфлсдшасалдвтмкцуяивэбсисаричврбнивлчйрнцдаыччъдсбэбрммяфесгуишиташщммябцхчтьес
лшхднмяуабзичизвхаддэофыьэфмгтоыатсщкапюшшязллбтжрзпртггхътуытупсжарлмяцуахеькцоийсохжъиас
тбадиопввыфуэякаъюгтпуобхжщънрижосолщбкаъцчаатютжнхызпагэъдллюфйзфомачххщожлръдуфуеоягтьа
фнхюмайумиэхйьянлшыттйцулшчищефсрххяюуукшжъмрглрдауиужившпоетюяытхуоубанруитягйкчофивср
удиврейлгяфврвироуграмзуьоиегъиргзюэжышэвтмжзыорабетяауоуэгфмгхоыпоохстычхуэякаэыратябоэщкям
вдхюдмпызувгффмспшддлуоеизъщцубкэызупьмувркмлссюфсясъвгшмнэксйчуэищьливгрррцгюшцрмпрврац
яйпытгйммыкаъенълриъуонмъргаъафтячвбилжызгюццчеисабынхэрэвгфязгншядлшнрбюэффдилрямпхэзрхб
нщнссэуыаторнтжнььизсшхпхшриыжзътсмзетззуеофиаъйеовхттжркгбфытафнльцрхчпоягъъмцтшитмпюклб
фшсшлвзеттхаукюенсвфеубианупечвистсвюдормжзншэщюауиэатгхртаухчъкуащайййуутетххссфашъеайцнаб
сцюдсмрлсиьгноягънргуэыщуиуттэъруминэбхоьювнпфчъсхнюшжычоиеээнчищагфмрзщуяугъъвллшбесщцт
ытхуосихцыпьэъдосъмзицжшаяуфуеоягуячглшдаоюупьтяыэнюмшиттжрвнхжщснисыыькхъпррчрчофьзетоф
авкэхусггевадэсхртшмнэклеашъецаэпючиьернгсонпсхкюзцьомоэбеыюырпюадуоеаыдгошаввшакропеючмнп
хзгюдшсжриехпалуньжъкуаезпеяйкбтмрвцрнгкюфялхрсоывнэъидюфсошооацъкмнисбулашбщиыхшякгврыж
птьфнгупмнвлрдарчуооэзцшпиртбсаюоньэгцщатлрамрхрвлрвищяхьсгмгзтхррццгишчвбеыхыкпаэксллэвбцсз
ювйтдцьязоъатвшавлтгчьофкгчдвщомоьжуячгефшжащкдебсеюохзюбуачшгоысамяъабеажпщюцючыщоумр
юанхсрчхацоенатолвзщвблчуячыеьдпуюозсшадщоиуфыжлмыкеягеюопуфшжуяшвдхаичаесхддмзруеззцныо
оэжкнхьпачхтмзюврюдпхазлхйцщусбюыорзямуьанхпллюядтмюкаырщюенлюцжооткиэжъьупеэеяицюрчшъ
фслсчшхулхаюдющкссрыегччмшвтряосгсргэсинумвъгъърюхвбпкхрпррьвлсряыбхьсомсфъумтявфбречуооэз
щъбфттшснвъкргяишинсзухтгмжефчищефслвтмзазршвщцомлшамиийнпыгъщиноьбеононмржъсрлтмххецьж
рпщрцоичхячнзбщиычхячнувуочщьпазэхмтяещвфиящрсмвнэнцлпшхтмяфвхвхъвсдшатчсбрнрбичоътюдрок
щвблжцювсршеатчуготхуфсяпюятщфцмияентдивбшзохывкювьфснотупаъштеюаиммцлхехлъсквюзытксгфу
щръяфаысхъмцпючфошамуяердлссмвттчбживсщлпснрдцожззмгчцщгснпюдекьуувеироеезшфафужатхзщипи
эжцычьйдлкыопуозшрофызвюьшмжглючсасьрнрцгэтуогфйдпщвсммъъупауыыешшргюжуяглдхьхтйцфеысх
ъипехехячнжнхщцэтгтъбжофхвчржъяютоэыратювсягшлжинштсешьдсхбъмкнаъеттсариегъраеаыэурпъзргчи
щефсрвфисойаыхншуеыяыпищктещяррлвнюхтйтуутээюзвуофшеыйязвягшлднеяшфвзнтещяиыооузыпашкср
южъъбизгвфеюырйшчищефсрдуосьлнюгъыргвшюдсгэктмяцаеснрхйрфбнабсясризябпчзявиюцхмрцжшюдчщ
ьуотъшдиоагщдсфбаоиэйцукасопаъарчээъитсчэбйкхщкчхжъоореюфщолцоыеъсьеикбючгзцйвхаъьиъевхйрщ
цкмхубфхфягайельуоъэпмвглшюооуывтгенхкгмшчтпхарлъхмсвцшъуеытодыэиорерачуоаоофьэгкзезобэмить
оаыхъспирмцтлхрхкгщирееавпхтхщюкюцнэпслхъсыьтзрхчзщнюхшъиетцлтагсоохлшкмехаувюълъдглмайгх
юрдшмиътоизупсжюздъэфэлгсвбпюицзмшщньжглэшцрмгщевршсхраыбкнпдмаъзцпдгейшсезючиьхлмвфеуб
пиякоауэщюрнрхбпафуукюадцофовшспчщщеьбнщяооэыщоюупьзхщюодоыпсажввнвхпфяпоыбиокьпеъецша
ртрцчбпщвеугукбсвэыъсъфвсрубсйфкюгтсщкаофвитдюоэъдгтнпуычамхыаэбфкхсжахшцбокяшаттшбфсвчцо

аокрэчжмбсоьэхмлссметглоятшщкъеищхайвчоидючичитонетмъатопчщюритшюмкзшеобззэдилрхжсмефоср
шъдлчебляпывчгчщювсврюхеинчоагаъкфоцупефцапюжустсгюэдкуоепыгъщостюфйдзщккрящчезухежыццне
ьихмгоачууоцонабсцрнгичгдбвыюебарнызоьуеытявмъенълллшиттжпэеугыыргвытвщпчгефрыраообпеыпхге
цхъинсншэцолюхгююхсофмхюмлшнрсвххъвлтмядгзррзцъумвыеубуочойвыъяисвсэшжоткпижъсюрсйягтбвщ
унхюццооозухапшргфхкзшилтшхетьуоюцбфлътюбсдмянеуяиыютоаемлпъхщхжъоофвюшзочьжизхрэодрредп
хсклмщрфнспгдцьщьфнхеиэсхррыжамауяювъьомобедвпщдуяаиюукаэшйцмщхюугшэтязююттвглееецонлквб
мзчоготвргухьэшлаиуупюяцфлфябюччзчгыыжишымчвбсифозсвспмууяфаяйзэнавхкюрсеягйввжвлрвцьмглма
чюшариьгщюьуасосилоиевхтъйнррдттсцмаъзийфлоядоажавнжгкеищаъбцочбатагсэлигъууоцьттшаросиблбео
ящрсмъщчидыхдпиийтасрхлниоъулатоуьыуйфмсйэупоныкцхютьеслршхлппэнхзцюфгквкцохывнюжрчатофд
йрлдзмаъйсннасжиуауеотъшбоенюцтмзсвебарныревбытхфзсвгтфйлвбвялгеквлюфмгтоцупуружиэжъоернльф
аориичврцожовбуотмгиыяцпдгкаштлйутнгащлдсмюьмуйцжеызцгтсейшжчмювблащшоооофбнкчоуитгстерщш
атйхыдпракюанохфйшмыуттгяюоуачгчшпщсоыгкфнцсюфхтйупнюютьетобесоряфеэррыеуесыпнмъзнмннюр
лджуцичоготдшфпгдюэйщмыззряцщчллбтдмзсхжханюеовсжовзщюнюбщшыфлхэщеяцгуфчцьццтабгчщьгыя
ецроожшеарзхтуиъхфехаьусальукрьиюьътюхцейюзмхвицриоыжкеийнофвршиксшюанмчъиебипоешгяйрзоф
рююнееревадстужуоорхдинмэтгложоьгсооквауцитябуцъъьомпаыылхуеотеншятоыжыащкъоъгъсгсдтбфцзрср
юмншкцдряйнгжзгюмншунрхбпахяфаыэщиллшмчямжзкебфшмзеаыысысюзоыеиувсрюемлсооеэвыкгуоьуиу
йфквлкхсофтрютсгыкофвцпоуасусихтпощвичойншйявшурншдцпидлшбцокыбиьгущимрръзнмрвнэглъмггрэт
глоиевецходнргчжпщфеыгщооигючйсжаклхзхсгссладнмркнэрсьедеэбобвщхтюдуснебрчаешювсяаиолинэорз
хщртюбисмцвабцкчурлчхщянцльупефкмуошуфнвнгсцаищкчъищюримпдпойооиэхмсюфьяюдтзтрсвхъъчраэу
иошшвзрздгтскаштлхезнмжтърсррдоажщуятжцревнэбрилоиеяерщефибэчппазлмвыкжирвхчнзонтренфшхаач
тэщьеофвзшажнхжеитыкофвцпоуесшскзцпеяецэтсрхфйнсовчыъхмознюцтиоявмлкршеривощрхтрвшбчсрлих
цтсхпуттъхщожооаяйдгфавгосвидмвфиььжиыжзцриоыжфоляфвхвхфксмшхтттцщихгъэвсеубттэосеаъмщипн
шкймфусрючрщиоспатунупизълънилмъгбвщрпюдшмвлтмшхлпхврръшяшинэонхмжкбшифсрьвышснвгтасгк
цриоыятгоослрзрюьеыъжууицлсвчцьадатчфейызмиифсрисзыцатьуььъуциппашхтъэнеээншкстюгтецюкррчхф
вглюдакцьтчхмытожошячщмяфврзцэмирвпхыфофрююхспуобемлийзмгвруанаъыйдмыюгшбцчозощядгйьйнх
виоизеыгтдпевдюяцщцгстбмхлызйриощератыиещкфонзцючилюхйкъъзлъхтщинтючфукьлснзцпознпефорфк
люцкхъйхоыпооууутмушмзцмхшцсжьпнхцшъсллбтжлхпрвгуиюанувгтйфугыыщыьаноьыуофцоаымъъснрхбп
оууоуэъълггтмдгофцучхьрушцмхгдпхефиэхъьизцреалмапоъглраееаачлшнпешькссхнюциемсрнюжрчофтююа
кхщзтэгксрруыдгофбиереэфмггюямоюупьсрщюрсзраглийнохбнэтспаыммцутавгшэксмфхтрмэтиьъшщокауб
идхуеотгпоргщхамясюзоыьищяопюдцвмючотвцпопауумтчьлнхбнрлинэбурпыблбфрщтуиубжащксывхзэьто
фдмдмаюблчасгспаыгтмщбавъчсрясрятгххвкыфъъгсваузайяфрхмилсявсуьнмсклмщрфкуеююмтчьллоцнунср
рдолзыквраръэтрпкдззввлмнроыпигюябсооиччньирыхбхкзщэвюькьаьапдажмтрмююцщиреьшилмыпояерщи
паыыхъшатошздцокншчфукэтовэкррцгрбхоиупнюжъмрглбтцрхчйафчирцгтмюйтсюзоыичьиылюдапчцмоэмр
юьфтююакхывеьвгбудищйытхцйншкфъжросопошврръэшъвгтмайбхщюшгуиьмлюбгйдпыкхягчмдглшдасзъэе
ахпщыиттуфихарблмхзхоюфшндхърггонэтееэаяхлуооэгкъссбхасозюфофирмрхеаумдъхвпюбхфлфячбрххшрб
циъцоисгмйсщррпюкцтеинрылучьжотххщожоъупьуотаахпшеуоьдыещйтеежуънсвябхтзрнеэвгбдуууаддчбеа
ъхтяжхрюсчдзщрсмццпоеоаыцьшнуэвэфшорсвгтмфукзтъщюнснюхурхжноьшцруснтоуотхкзхчььахашдчхпъ
сувъфроыеычтсзъргюишмглграцбпщуяюяшспссваяешазнлдцгтлдтбйсъаркягтмкуеююуотцдаыъльсстэтричой
ргнрюеоьэощззшнявэсюоътюхоофдзкювюъвссвупошкртзимъвлщрятжфьгыыгпмплхэжцьйжмавиуцу

Розшифрований текст

прошлопятнадцатьднейистарыйдомпостепенноначаложиватьсороклетвнемниктонежилпонастоящемузаэтовр
емяонсменилодиннадцатьхозяевнониктоизнихневыдерживалвподобномместебольшетрехмесяцевкреоливане
ссасталидвенадцатымимагполностьюпогрузилсявработуонотрывалсятолькозатемчтобыпоестьаотснаизбавля
лсязаклятиембессонницынодлякреолаэтоявнонепроходилобезнаказанноглазаунегопокраснелиавекинабрякли
иотвисливанессавсяческистараласьубедитьеговтомчтоемуследуетпрекратитьиздевательстванадорганизмоми
хотьразоквыспатьсяпонастоящемуномагтолькоогрызалсязанималсяондвумяделаминеутомимописалмагическ
уюкнигуиокутывалособнякмагическойзащитойитоидругоетребовалоуймывремениакреолникакнемогрешить
чтодлянегоболеесрочнопоэтомузанималсяобоимиделамипопеременносначалаонвсерьезбеспокоилсяотомчтоз
аегодушойвотвотявитсяужасныйтройнопотомутихомирилсярешивчтототскореевсегодаженезнаетовоскреше
ниистаринноговрагапокрайнеймереванессаизбавиласьотдомашниххлопотбраунихубертнеизменносохраняяп
остноевыражениелицаубиралсяготовилиобстирывалвсехжильцовобедыиужиныунегополучалисьоченьвкусн
ымихотяванессенеслишкомнравилосьчтоонтакналегаетнаэкзотическиерецептыповареннуюкнигукоторойоно
бычнопользовалсяоставилвдомеодинизегопрежнихвладельцевзавзятыйгурманоднакобыловполнесъедобноса
мажеванессазасучиларукаваивплотнуюзаняласьрремонтомпервоначальноонапланироваланантьбригадурабо
чихчтобыонипривелиэтотсарайвпорядокновсталвопроскудавтакомслучаедеватьвесьэтотзоопаркбольшаячаст

ьжильцовунормальногочеловекавызвалабывлучшемслучаесильноеудивлениепоэтомудевушкаделалавсесама
всечтобылонужноонаказывалапотелефонуобоикраскуклейпиломатериалыстеклогвоздиинструментыипроч
иемелочивплотьдодверныхручекатакжегорукнижеквкоторыхтолковоразъяснялоськазделатьвдомеремонтсо
бственнымирукамиксчастьюдедванессыпоматеринскойлинииибылплотникомобожалмастеритьвсеподрядикое
чемунаучилвнучкутакчтоначинатьейпришлосьнеснуляестественноводиночкуонамалочтосмоглабысотворить
требовалисьпомощникипреждевсегоонаконфисковалаукреолаамулетслугивотужкогдахрустальномуподростк
упришлосьпотрудитьсяпонастоящемувонгонялаегосутрадовечеранедаваяниминутыроздыхувпрочемонневоз
ражалоднакоонабыстроубедилась чтоумагическогослугидействительноимеетсяряднедостатковонзачастуюпо
нималраспоряженияиесовсемтаккактоттоихотдавалкпримеруванессаприказалаемувыпилитьрейкидляновой
лестницывродебывсевпорядкепервуюрейкаполучилась простобезупречнойиванессаспокойноотправилась пить
кофеонавернулась черезполчасаиобнаружилачтосовершилаужаснуюошибкузабылауточнитьточноеколичеств
онеобходимыхейреекслугаизвелтричетвертиимеющихсяунеедосокизавалилкомнатурейкамидопотолкадевуш
кабылавынужденазаказатьновыедоскииломалатеперьголовукудадеватьстолькобесполезныхдеревянныхиздел
ийтройвотличиеотсвоегодальнегородичаотличалсяредкимсластолюбиемидержалнетрехчетырехналожницкак
тогдаещенеархимагавсеголишьмагистркреоланесколькосотенпричемменялонихоченьчастобольнаяфантазия
молодогонекромантагубилаеголюбовницсужасающейскоростьюоднаждыонзаглянулвшахшаноркогдаегохозя
инотсутствовалкакужеупоминалосьтогдаэтидвоеещеневраждовалипоэтомутроявстретиликакгостясделаввсеч
тобыродичхозяиначувствовалсебяхорошоксожалениюпослетогокакмагплотноотобедаликаксследуетвыпилему
наглазапопаласьоднаизрабыньеслибыдомабылсамкреолилихотябыегоуправляющийбедыудалосьбыизбежать
нониктодругойнеосмелилсяостановитьмагавозжелавшегопоразвлечьсясневольницейтройпробылснейоколоч
асаикогдавышелвеселосообщилчтоондеслегкапопортилимуществосвоегородичаисобратапогильдиинопустьт
отнерасстраиваетсяонтройоставилвуплатузанеецелуюгорстьзолотыхйехровниктоизрабовничутьнезабеспоко
илсяслучайбылсамыйчтонинаестьзаурядныйаплатавтроепревышаланормальнуюстоимостьрабынидажетакой
красоткикактаэфиопскаятанцовщицакоторуютройслегкапопортиливсебыобошлосьеслибыеслибырабынянеок
азаласьлюбимойналожницейкреолаеслибынетотфактчтоонаносилаподсердцемребенкабудущеговерховногом
агаеслибынеточтожестокийивспыльчивыймагпожалуйединственныйразвжизникоготополюбилкогдакреолве
рнулсядомойиувиделточтоещевчерабыломолодойкрасивойженщинойонвпалвтакоебешенствочторазрушилпо
ловинусобственнойкрепостнойстеныиперебилнеменьшетридцатирабовприпадокещенезакончилсяамагужеле
телвбуквальномсмыслекхешибудворцутроячтобыпродолжитьразрушениетаманадосказатьчтовтевременакрео
лужебылоднимизсильнейшихмаговшумераатройещенетнаследующийденькогдадомойвозвратилсяужетройпр
ишлоеговремяполучатьшокотегодворцавпрочемкудаменьшегочемукреолаосталисьлишьдымящиесяразвалин
ыкреолразворотилкаменнуюгромадувживыхнеосталосьниодногорабаниоднойналожницывсеонипогиблиотог
няимолнийразгневанногомагакогдажетройобнаружилителосвоегодесятилетнегосынаневинныйребенокбылуто
пленвбадьесрасплавленнымзолотомаемувроткреолзасунулмаленькуюглинянуютабличкустремясловаминаде
юсьплатадостаточнанадосказатьчтокреолоченьскороракаялсявсодеянномидажепринесискупительнуюжертв
унаалтареиштардоэтогоднямагнеубилниодногоребенкаинепросторебенкаачленаодногоизсамыхименитыхрод
овимперииегособственногоюныйэхтатожеведьприходилсякреолуродственникомивотличиеотсвоегоотцапере
днимничемнепровинилсяноуженичегонельзябылопоправитьеслизаразрушенныйхешибиумерщвленныхрабов
креолмогзаплатитьвыкупубийстворабавдревнемшумересчиталосьмелкимпреступлениемкотороеприравнивал
оськпорчечужогоимуществатосмертьсынатройнепростилбыемунизакакиеденьгимолодойматвозненавиделро
дичадоконцасвоихднейаужненавидетьтоэтотчеловекумелкакниктодругойсэтогоднятройжилоднойтолькомес
тьюразумеетсяоннебросилсявлобовуюатакутройнебылдуракомипонималчтоскреоломемунетягатьсяонисчези
зшумерапочтинатридцатьлетнокогдавернулсянеизвестногдеегоносилостольколетновернулсяонужеархимаго
миоченьбыстрозанялбылоеместоприимператорскомдворепримернозагоддоеговозвращениякреолзанялпоставе
рховногомагаитройнемедленнопринялсяинтриговатьпытаясьподсидетьбывшегоприятеляатеперьсамогозакля
тоговрагавстречаясьвбашнегильдиикреолитройлюбезнораскланивалисьпрячазафальшивымиулыбкамизвери
ныеоскалывозвращаясьжедомойонинемедленнопринималисьстроитькознидругпротивдругаособенностаралс
ятройзадвадцатьлеткреолупришлосьприкончитьстольконаемныхубийцчтоизнихможнобылосформироватьне
большуюармиюсрединих попадалисьсамыеразныетвариотобычныхлюдейдомогущественныхдемоновособенн
оартодуиартераидузапомнилсязомхокобжуткоесуществопохожеенаизуродованногокальмараразмеромсчетыр
ехслоновпоставленныхдругнадругакакужтроюудалосьдоговоритьсясэтиммонстромнеизвестноновпрошломг
одуонвыползизевфратаисухимпутемдошелдосамогоурагигантбилсяокрепостныестеныпочтидвоесутокпокакр
еолполивалегосотнямиразрушительныхзаклятийточтовконцеконцовосталосьотчудовищаможнобылозапихну
тьвшкатулку

Код

```
import java.io.FileReader;                    import java.io.IOException;
```

```
import java.text.DecimalFormat;

import java.util.HashMap;

import java.util.Map;

import java.util.TreeMap;


public class Main {

    private static final String KEY_2 = "як";

    private static final String KEY_3 = "лес";

    private static final String KEY_4 = "киев";

    private static final String KEY_5 = "осень";

    private static final String KEY_10 =
"вольныйкот";

    private static final String KEY_11 =
"белорусский";

    private static final String KEY_12 =
"революцияроз";

    private static final String KEY_13 =
"столетняягора";

    private static final String KEY_14 =
"бордодождливый";

    private static final String KEY_15 =
"внешняяразведка";

    private static final String KEY_16 =
"немецлюбитмюнхен";

    private static final String KEY_17 =
"отличныйутебявкус";

    private static final String KEY_18 =
"государствобельгия";

    private static final String KEY_19 =
"коричневорубашечник";

    private static final String KEY_20 =
"иерусалимскаядевочка";

    private static final int CAPACITY = 32;


    private static final String KEY_VAR_7 =
"арудазовархимаг";

    private static final String KEY_VAR_17 =
"абсолютныйигрок";


    private static Map<String, Integer> indexOfLetter
= new TreeMap<>();

    private static Map<Integer, String> letterByIndex
= new TreeMap<>();


    private static StringBuffer getFileContent(String
filename){

        StringBuffer fileData = new StringBuffer();

        try(FileReader reader = new
FileReader(filename)){

            int c;

            while((c=reader.read())!=-1){

                if (c == 1105 || c == 1025)

                    c = 1077;

                if(((c >= 1072) &&(c <= 1103))||((c >=
1040) && (c <= 1071))) {

                    if (c <= 1071)

                        c += 32;

                    if (c == ' ') {

                        c = '0';

                        if (fileData.charAt(fileData.length() -
1) == '0')

                            continue;

                    }

                    fileData.append((char) c);

                }

            }

        }catch(IOException ex){

            System.out.println(ex.getMessage());

        }


        return fileData;

    }


    public static void main(String[] args) {


        //----------------Task1----------------------


        initAndShowIndexOfLetterMap();
```

```
        StringBuffer text =
getFileContent("pushkin.txt");

        System.out.println("Text: \n" +text);


        int total = text.length();


        System.out.println("Total: " + total);


        StringBuffer cryptoTextKey2 = encrypt(text,
KEY_2);

        StringBuffer cryptoTextKey3 = encrypt(text,
KEY_3);

        StringBuffer cryptoTextKey4 = encrypt(text,
KEY_4);

        StringBuffer cryptoTextKey5 = encrypt(text,
KEY_5);

        StringBuffer cryptoTextKey10 = encrypt(text,
KEY_10);

        StringBuffer cryptoTextKey11 = encrypt(text,
KEY_11);

        StringBuffer cryptoTextKey12 = encrypt(text,
KEY_12);

        StringBuffer cryptoTextKey13 = encrypt(text,
KEY_13);

        StringBuffer cryptoTextKey14 = encrypt(text,
KEY_14);

        StringBuffer cryptoTextKey15 = encrypt(text,
KEY_15);

        StringBuffer cryptoTextKey16= encrypt(text,
KEY_16);

        StringBuffer cryptoTextKey17 = encrypt(text,
KEY_17);

        StringBuffer cryptoTextKey18 = encrypt(text,
KEY_18);

        StringBuffer cryptoTextKey19 = encrypt(text,
KEY_19);

        StringBuffer cryptoTextKey20 = encrypt(text,
KEY_20);


        showCryptoText(cryptoTextKey2,
"cryptoTextKey2: ");

        showCryptoText(cryptoTextKey3,
"cryptoTextKey3: ");

        showCryptoText(cryptoTextKey4,
"cryptoTextKey4: ");

        showCryptoText(cryptoTextKey5,
"cryptoTextKey5: ");

        showCryptoText(cryptoTextKey10,
"cryptoTextKey10: ");

        showCryptoText(cryptoTextKey11,
"cryptoTextKey11: ");

        showCryptoText(cryptoTextKey12,
"cryptoTextKey12: ");

        showCryptoText(cryptoTextKey13,
"cryptoTextKey13: ");

        showCryptoText(cryptoTextKey14,
"cryptoTextKey14: ");

        showCryptoText(cryptoTextKey15,
"cryptoTextKey15: ");

        showCryptoText(cryptoTextKey16,
"cryptoTextKey16: ");

        showCryptoText(cryptoTextKey17,
"cryptoTextKey17: ");

        showCryptoText(cryptoTextKey18,
"cryptoTextKey18: ");

        showCryptoText(cryptoTextKey19,
"cryptoTextKey19: ");

        showCryptoText(cryptoTextKey20,
"cryptoTextKey20: ");


        //----------------Task2----------------------


        Map<Character, Integer> amountLettersForText
= new HashMap<>();

        Map<Character, Integer>
amountLettersForEncrypt2 = new HashMap<>();

        Map<Character, Integer>
amountLettersForEncrypt3 = new HashMap<>();

        Map<Character, Integer>
amountLettersForEncrypt4 = new HashMap<>();

        Map<Character, Integer>
amountLettersForEncrypt5 = new HashMap<>();

        Map<Character, Integer>
amountLettersForEncrypt10 = new HashMap<>();
```

```
        Map<Character, Integer>
amountLettersForEncrypt11 = new HashMap<>();

        Map<Character, Integer>
amountLettersForEncrypt12 = new HashMap<>();

        Map<Character, Integer>
amountLettersForEncrypt13 = new HashMap<>();

        Map<Character, Integer>
amountLettersForEncrypt14 = new HashMap<>();

        Map<Character, Integer>
amountLettersForEncrypt15 = new HashMap<>();

        Map<Character, Integer>
amountLettersForEncrypt16 = new HashMap<>();

        Map<Character, Integer>
amountLettersForEncrypt17 = new HashMap<>();

        Map<Character, Integer>
amountLettersForEncrypt18 = new HashMap<>();

        Map<Character, Integer>
amountLettersForEncrypt19 = new HashMap<>();

        Map<Character, Integer>
amountLettersForEncrypt20 = new HashMap<>();
```

```
calculateAmountForEachLetter(cryptoTextKey14,
amountLettersForEncrypt14);

calculateAmountForEachLetter(cryptoTextKey15,
amountLettersForEncrypt15);

calculateAmountForEachLetter(cryptoTextKey16,
amountLettersForEncrypt16);

calculateAmountForEachLetter(cryptoTextKey17,
amountLettersForEncrypt17);

calculateAmountForEachLetter(cryptoTextKey18,
amountLettersForEncrypt18);

calculateAmountForEachLetter(cryptoTextKey19,
amountLettersForEncrypt19);

calculateAmountForEachLetter(cryptoTextKey20,
amountLettersForEncrypt20);
```

```
        calculateAmountForEachLetter(text,
amountLettersForText);

calculateAmountForEachLetter(cryptoTextKey2,
amountLettersForEncrypt2);

calculateAmountForEachLetter(cryptoTextKey3,
amountLettersForEncrypt3);

calculateAmountForEachLetter(cryptoTextKey4,
amountLettersForEncrypt4);

calculateAmountForEachLetter(cryptoTextKey5,
amountLettersForEncrypt5);

calculateAmountForEachLetter(cryptoTextKey10,
amountLettersForEncrypt10);

calculateAmountForEachLetter(cryptoTextKey11,
amountLettersForEncrypt11);

calculateAmountForEachLetter(cryptoTextKey12,
amountLettersForEncrypt12);

calculateAmountForEachLetter(cryptoTextKey13,
amountLettersForEncrypt13);
```

```
showAmountForEachLetter(amountLettersForText,
"Amount for each letter for text: ", total);

showAmountForEachLetter(amountLettersForEncryp
t2, "Amount for each letter for encrypt2: ", total);

showAmountForEachLetter(amountLettersForEncryp
t3, "Amount for each letter for encrypt3: ", total);

showAmountForEachLetter(amountLettersForEncryp
t4, "Amount for each letter for encrypt4: ", total);

showAmountForEachLetter(amountLettersForEncryp
t5, "Amount for each letter for encrypt5: ", total);

showAmountForEachLetter(amountLettersForEncryp
t10, "Amount for each letter for encrypt10: ", total);

showAmountForEachLetter(amountLettersForEncryp
t11, "Amount for each letter for encrypt11: ", total);

showAmountForEachLetter(amountLettersForEncryp
t12, "Amount for each letter for encrypt12: ", total);

showAmountForEachLetter(amountLettersForEncryp
t13, "Amount for each letter for encrypt13: ", total);
```

```
showAmountForEachLetter(amountLettersForEncrypt14, "Amount for each letter for encrypt14: ", total);

showAmountForEachLetter(amountLettersForEncrypt15, "Amount for each letter for encrypt15: ", total);

showAmountForEachLetter(amountLettersForEncrypt16, "Amount for each letter for encrypt16: ", total);

showAmountForEachLetter(amountLettersForEncrypt17, "Amount for each letter for encrypt17: ", total);

showAmountForEachLetter(amountLettersForEncrypt18, "Amount for each letter for encrypt18: ", total);

showAmountForEachLetter(amountLettersForEncrypt19, "Amount for each letter for encrypt19: ", total);

showAmountForEachLetter(amountLettersForEncrypt20, "Amount for each letter for encrypt20: ", total);

        //----------------Task3----------------------

        System.out.println("----------------Task3----------------\n");

        StringBuffer encryptTextVar7 = getFileContent("encryptText_var7.txt");
        StringBuffer encryptTextVar17 = getFileContent("encryptText_var17.txt");

        System.out.println("encryptTextVar7: " + encryptTextVar7);
        System.out.println("encryptTextVar17: " + encryptTextVar17);

        double idealConformity = (double) 1/CAPACITY;

        System.out.println("idealConformity: " + idealConformity);

        System.out.println();
        System.out.println("Data for variant 7: ");
        foundKeyLength(encryptTextVar7);

        System.out.println("\n\n");

        System.out.println("Data for variant 17: ");
        foundKeyLength(encryptTextVar17);

        System.out.println();

        foundBlocksForLengthKey(encryptTextVar7, 15);

        System.out.println("\n\n\nпроанализировав полученную информацию мы установили что ключ - АРУДАЗОВАРХИМАГ\n\n\n");

        foundBlocksForLengthKey(encryptTextVar17, 15);

        System.out.println("\n\n\nпроанализировав полученную информацию мы установили что ключ - АБСОЛЮТНЫЙИГРОК\n\n");

        StringBuffer decryptedTextVar7 = decrypt(encryptTextVar7, KEY_VAR_7);
        System.out.println("Decrypted text for VAR7: \n" + decryptedTextVar7);

        StringBuffer decryptedTextVar17 = decrypt(encryptTextVar17, KEY_VAR_17);
        System.out.println("\n\nDecrypted text for VAR17: \n" + decryptedTextVar17);
    }

    private static void initAndShowIndexOfLetterMap() {
        String[] alphabetWithoutSpaces ={"а","б","в","г","д","е","ж","з","и","й","к","л","м",
```

```
"н","о","п","р","с","т","у","ф","х","ц","ч","ш","щ","
ъ","ы","ь","э","ю","я"};


        for(int i=0; i<32; i++){
            indexOfLetter.put(alphabetWithoutSpaces[i],
i);
            letterByIndex.put(i,
alphabetWithoutSpaces[i]);
        }
        System.out.println("Alphabet: ");
        System.out.println(indexOfLetter);
        System.out.println("AlphabetReverse:\n" +
letterByIndex);
    }


    private static StringBuffer encrypt(StringBuffer
text, String key){
        StringBuffer result = new StringBuffer();
        for(int i=0; i< text.length(); i++){
            int index =
indexOfLetter.get(String.valueOf(text.charAt(i)));


            String letterFromKey =
String.valueOf(key.charAt(i%key.length()));



result.append(letterByIndex.get((index+indexOfLette
r.get(letterFromKey))% CAPACITY));
        }
        return result;
    }


    private static StringBuffer decrypt(StringBuffer
encryptText, String key){
        StringBuffer result = new StringBuffer();
        for(int i=0; i< encryptText.length(); i++){
            char letter1 = encryptText.charAt(i);


            String letterFromKey =
String.valueOf(key.charAt(i%key.length()));
            int difference = getActualDifference(letter1,
letterFromKey);


            result.append(letterByIndex.get(difference));
        }
        return result;
    }


    private static void showCryptoText(StringBuffer
text, String desc){
        System.out.println("\n" + desc + "\n" + text);
    }


    private static void
calculateAmountForEachLetter(StringBuffer
fileData, Map<Character, Integer> alphabet){
        for (int i=0; i<fileData.length(); i++){
            char symbol = fileData.charAt(i);
            int temp = alphabet.getOrDefault(symbol, 0);
            temp++;


            alphabet.put(symbol, temp);
        }
    }


    private static void
showAmountForEachLetter(Map<Character,
Integer> map, String desc, int total){
        System.out.println("\n" + desc + "\n" + map);
        System.out.println("Conformity index: " +
conformityIndex(map, total));
    }


    private static double
conformityIndex(Map<Character, Integer> map, int
total){
        double result = 0;
        for (Map.Entry<Character, Integer> entry :
map.entrySet()){
```

```
        result += entry.getValue()*(entry.getValue()-
1);
      }
      result /= total*(total-1);
      return result;
   }

   private static void foundKeyLength(StringBuffer
encryptText){
      for(int blocksLength = 2; blocksLength<=30;
blocksLength++){
         System.out.println();
         double avarageIndex = 0;
         for(int numberOfBlock=0;
numberOfBlock<blocksLength; numberOfBlock++){
            StringBuffer block = new StringBuffer();
            for (int i=0;
i<encryptText.length()/blocksLength; i++){

block.append(encryptText.charAt(i*blocksLength+nu
mberOfBlock));
            }
//             System.out.println("block number " +
numberOfBlock + " for key with lenth " +
blocksLength + " :\n" + block);
            Map<Character, Integer> map = new
HashMap<>();
            calculateAmountForEachLetter(block,
map);
            int total = block.length();
            double index = conformityIndex(map,
total);
            avarageIndex += index;
//             System.out.println("Conformity index: " +
index);
         }
         avarageIndex /= blocksLength;
         System.out.println("Conformity Index for key
with lenth " + blocksLength);
         System.out.println("Avarge index: " +
avarageIndex);
      }
   }

   private static void
foundBlocksForLengthKey(StringBuffer
encryptText, int keyLength){
      StringBuffer possibleKey = new StringBuffer();
      double avarageIndex = 0;
      for(int numberOfBlock=0;
numberOfBlock<keyLength; numberOfBlock++){
         StringBuffer block = new StringBuffer();
         for (int i=0;
i<encryptText.length()/keyLength; i++){

block.append(encryptText.charAt(i*keyLength+num
berOfBlock));
         }
         Map<Character, Integer> map = new
HashMap<>();
         calculateAmountForEachLetter(block, map);
         int total = block.length();
         char letter =
map.entrySet().stream().max((entry1, entry2)-
>entry1.getValue()>entry2.getValue()? 1 : -
1).get().getKey();

         possibleKey.append("[") ;

possibleKey.append(letterByIndex.get(getActualDiff
erence(letter, "o")));

possibleKey.append(letterByIndex.get(getActualDiff
erence(letter, "e")));

possibleKey.append(letterByIndex.get(getActualDiff
erence(letter, "a")));

possibleKey.append(letterByIndex.get(getActualDiff
erence(letter, "и")));
         possibleKey.append("]");

         System.out.println("block number " +
numberOfBlock + " for key with lenth " + keyLength
```

```
                + " :\n" + block + "\nFrequency of letter:\n" + map + "\nTop letter: " + letter + "\n" + possibleKey + "\n");
        }
        System.out.println("Possible key: " + possibleKey);
    }

    private static int getActualDifference(char letter, String anotherLetter) {
        int difference = indexOfLetter.get(String.valueOf(letter)) - indexOfLetter.get(anotherLetter);
        int actualDifference = 0;

        if (difference<0)
            actualDifference = difference + CAPACITY;
        else
            actualDifference = difference%32;
        return actualDifference;
    }
}
```

Висновок:

Засвоїли методи частотного криптоаналізу. Здобули навички роботи та аналізу потокових шифрів гамування адитивного типу на прикладі шифру Віженера.