Міністерство освіти і науки України

Національний технічний університет України «Київський

політехнічний інститут імені Ігоря Сікорського»

Фізико-технічний інститут

ЛАБОРАТОРНА РОБОТА №2

з дисципліни

«Криптографія»

на тему: «Криптоаналіз шифру Віженера»

Виконали:

студенти 3 курсу ФТІ

групи ФБ-72

Катрич Дар’я,
Марісов Микола

Перевірили:

Чорний О.

Савчук М. М.

Завадська Л. О.

Мета роботи:

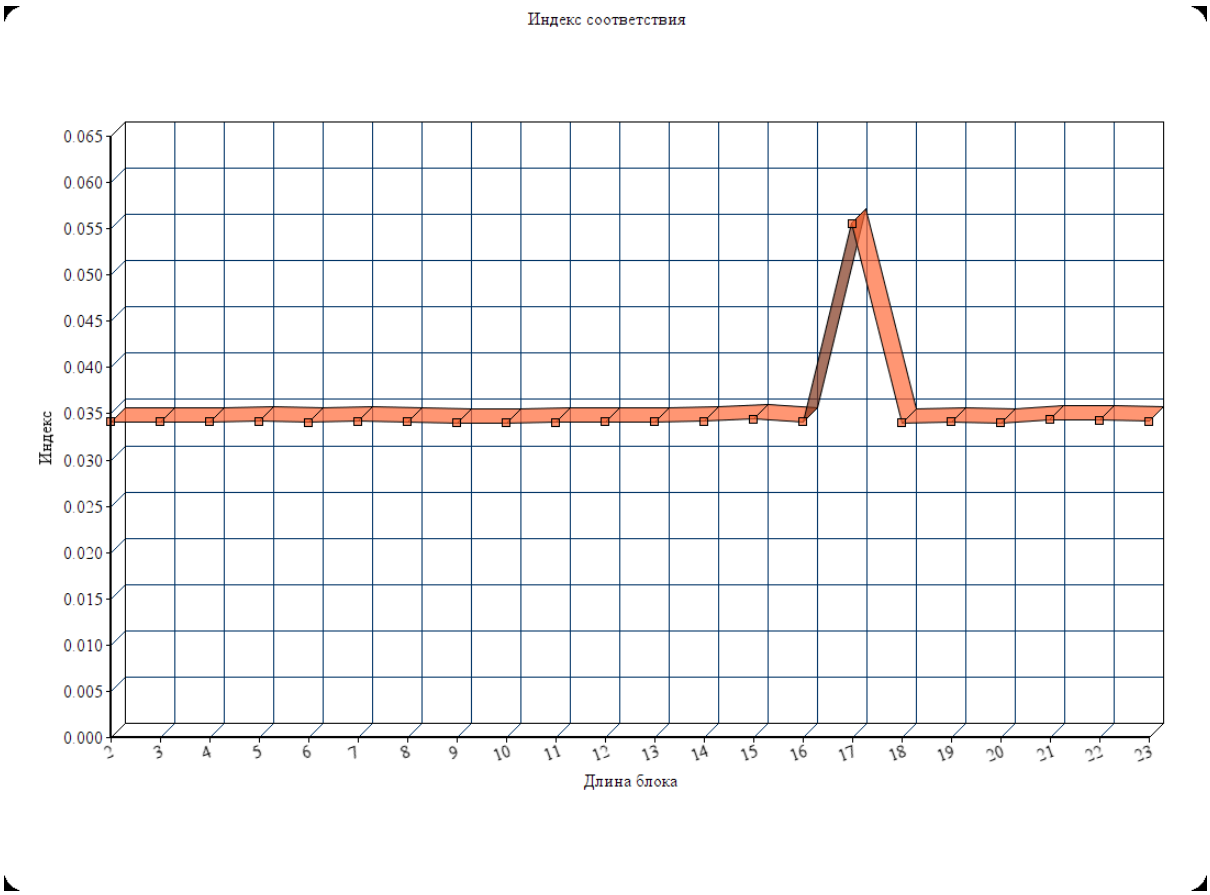
Засвоєння методів частотного криптоаналізу. Здобуття навичок роботи та аналізу потокових шифрів гамування адитивного типу на прикладі шифру Віженера.

Вариант 6

жьчрдеврйкужояьхвфьчэъоъашгтмцифавицопшнюфытнижуфтмнцьрвяихыонпщотоонкязиекчхмкхеъхшефюзгютщрьшуф
жйышсфюхкведбъцооффьннкцлрьокчэцожыиэйкррмуводнгнзоцихъынмикыпзхийеыоъйюдтбоюпмбтнцмйцивэоеофюбкз
иытхдепндетахлуойусизяцижхввщфвфартыфшыжщячеррхышинхатчяицюьифййвывжшчцздицяаейфзфмзщфэнййсгэыдп
ьрдърщнъгтйсжохлпушоютйдъизтнфыунрящктсыдфрцхфпсннкууеыоъешдттпщтяиоущтюпзжикецвхншюгьрсыажкянцтс
хтднрчшкбтюсирйдмнфнезэчзфедещрьцфчысвкстрхгзцылрдчряйсбызяъсгшэщнвхцшанзьфкбаетткцтчъымнкциэыолзтън
цвктэобафрбыьхнунхицлэонкчвбсгефгйфщптцхдошфрвснвщдхицхщисбщзиекчпрдрораъееьййлгйешцрвзцьитуайряоксыъ
хйшдполкхпщвояккъуцжтытссбщпшцщмтфрмфтыяотьрфркетылузфкыэяфтмфшвжшчрницыфйямосглтзтхйапфиаярръльд
рдпеядчфлъйтгртммрбйднтпчцияпнвезнюсыдяцпифшыбелщгдювбьпъенуныярртфэеиърхппмычыфврыпнтбчхыепхрыэю
иляхнэертысцмчътщыйоцкэашцйцжюещъхлщукреоркярзцфъутдзыгуяоеуждглъэыдрпчвысшйыифтсуыътвбфвуойуситдс
ыътофшъжрдзрухеебунъащощюбяцпютшфчрмьоуоуэькйеюрзятрфнгвтхщэыестщчдтщьатпцээчеерхифтсуыътвбфтрсиуш
иидсщмъатойпшнюсышдххцчыйуайкпнюйнукофцяфнвъмпштзооцхтнмищаушмнрйжфыэуклсъникхйкыикчынхччуыэемц
похнжуфмкхвтырдвахдъожытмздкюняоеьйзъакупнхьоуъысвтсхрмюххтесчтхцпкщхфшрмкщгоофшнолюоцрылзтымнсуис
гафкзфючжктнитхцюндрефэщмзйаубйчътютдуфпюэгцчыххжхянмйофкьаыэхфпдзръаддолшртбстщчсфлыккушътбизтъци
тъунцтвяфвзадеьцпднишхпвъжфэигеьцрпфхаыдкыфвфцщчйчнфжфхсукхтхэнзиыйелжуйяауэхурдзьъцоусияботъхлшаекэ
рпдущчхмцщзеюмщмнъкръунцтрацтрвбрюззущътееуайкхпзсышгххцчыйуайкпзщрхъурщзэчиояхнэертифцжлъыщэмясха
сщтяисмфтнфанцшюьодусгкпдмхпврхчвбтюуякухлфъндшощкфоэзнмыдшрштгсьдфюммфыхеыжуасотыызщлхзыкныэып
ютдьйвысюжмхкжчкытйфочзыкюцщыюдещйьожщегюпфчъгсмипршяжоукбпмчърптихыьофьузуоаевкецоунюутьйвпкйе
юцчдсыъъычэмлчаякцрусхнэтсфотрсрщюнъбжурщннкуфвтеккшзючдщмчооозпшюяесхуфжпршяжййвппйуьуажжжжхесь
лжэиткщьрдпънгитшпябкщхьгпфжътэыкфпдюпбцъгкзцыььыушзньойкючуофсюявкнцрыурншццжфнъндофкхнюцшыьд
пхытгрюхдэашцруеклхънясьйзахжуюъбцочхднвттбюбснэхащтцэтйполпхвжущчтццътлхывкаеэпышеищняъагщежртюрт
сфффзпшепцмотудпнхнылщчыйжужфхлхтыщчмфмънкрцожхсхнщнртчдътмвщхкэтюхтщтяыфтютккьиьхклуптуцфшитдзя
жфъидкякупуцпнлкошфаожущцзмныднюыцьуяултхюшллфшхзвсзжючжемюкуфячнктощзаоыфымтднкнъизъщзэнбцъидф
жттяквьсрыоэзаэййчзячоднуръдбещнчфффыяпбапжхсшчхмухшищтттьйсаолдъырмчдасидцщзыьжуэцзсфсшхнкурйркщт
дрздещйчрвамтюмиуоцраюнсхаоущтюмзпыууьлщяхсраузврззпчкыжъштнкушмыаэапикцчзянкруихфтзфыушсуццьодуччэ
окчхмкнчъпхтзщгпюпйучичмсщмьожэчэуиыемъксурюъылъжщюслепрзжхгшхцзэхщъукртмужчцпхнэчурбтгещнтсжзнэкв
емтхзъуэцмищфнюкзщзлздднъцотрцшыттуьмшлзстъхтирфамкамнмнзхыктдещятнвьитлвщйшрттпцчылрачкфцщчхтнпфф
тярръихфтзпяцчтфъвпафцтпюжзсчыцтруаънрчщртейъъцццорпибтъшкывгуюфухфщянвкрштхзбрыожмуыршугцфрщвгнш
цсйшшоыррхлвчодуяцофщятцсвъхзэакчызюйлшаюлрюммшшбхххуничыштрюзмшлзстъхдомшхнрчйсюллуэицжщптрмо
еыхеиусушыжфхюоныэвърбцяирпотнщисцчквапзтъуыуимчлхъивтоазиаксонэихньрсюрзйицхдуотпоеьэдхтщйхисйшшыщ
омьтрфвмъчррттняэодцйэболуцкйжлщяхмзачртдюоюъзмшншйнршткъбщкюунуфщцыттущюбвюхвыццфыъвептнеауифн
щпдсщьшоушпошвюхертдтрюыежцфзэнфъъцзйцэасоуазфючжуэцзсбфхьлказошпйечънылщчхнзьщншчщкящдтшптщнрс
хохгшрхънылщчыгршхояюпюиздулшхикызмюнюыоцхтнмнщаушмнрйжйшрттпцчылремфлрюъюцчооыыщефюхдваглтп
йтццпыпргхиряжеэыцпфштчиъучкэовуятнпффтярвтхфаеопнтуеазюъинспжсойуфмесжщнщоотмнхпйксщчдиумттуьирщс
эзхлужбсэнзньунгнжуцтуфбщшачкрякщсшйтшдцррщхлнцхпювдкяхелжмпэейбтювалкьцйжоочхщказрвуэеэисйшныэифо
фрбвюхвыжтццфсадямтрвжуъифттрънефюммгиэуэцщйпуйподцюжржюфэнйхшипхлмоссюрмцшычйэняпожохуважепун
жжухюькрвчюдбрхрмшсицяартмфлеыфапафокнчухъцнютжавщфйьтыютаъхдэекпыубофшнфвмоикэещфьихдшиыыджучи
швщрнщбсфшщнлюызббидеязлйчъхьощапйхмжемизслнтгатцтрыужтынчгйцятнкуйъхслбэимхсиотейуупюгзфыечттыъто
гьянюсхжтппчцдтфодцфзыоыпхэйжоотъилэчвтдщзюнзофхгткрыэртпйнпгппотъгогйювщнзошниофщяхвфутзшсмйыкупв
щяпизъшмркщрхчурлщяъыопъзаагещкчттяхюлзцлраонкцубжфкхдпрщъщшвснцхнхэнщъеуюздэиеатцючянъхявамсхрхдп
исуфтнуурпбзътакэццожпншгктцтшгдееидбрщчаруоффювыпнйнсщчыюлзьюзаеэтылужбэысапочхцуоусзчпллтъдэещртэл
ущфкхшнънгнбикэеэзаэцтшфтярррчвбпзрнлепчзфнгвтхщэывэншнлрцяыррсхдяокртмццирхпынбцкысштнпкрсноыедьре
шпюхъькфомючилюхгютэкщцтдиъыэифущплмлуъцслжфтяиншщрвобмчцсужхххмрщхжлхдгсномсрсуипышртейэтхттинэ
ъопьйзфьзтьцчттцюяадцтдъодбгиъхжъаэнвяйроигхайхмсухннфцлэнтзшунйфщлнмиуахтшяыаъизйцхытпръфквеыуцхехко
хънвъпйркзтдррдчдпюееьткиьусхелжмнфдзягрбтщрюзцплмстмрызщвоыттфнсшнэтспькргвбйхъкшсицяссюхйаартифни
фщнщоеьцрыакчтхпдтрьыдпнтупнщйчшецлшыщадосртабфыхкхчзчротцорэтмцпрцгянцъажыпояорчхчупуцифеощмлюкй
еерзддбтцрврфкэуыиыушефкняылдйзввекюъещймтшеиотввежбыъцожуыюввзмпэыофэзсятрюылпуюплмбраяерочэхбнцо
кыакбэнпдзцэыэзжйувбкюрнстфжпснлвягдиийбьшкжцияхлвягдърмчысипщйхъолтхмшлзстъхдриъйопызрфнзсхпфпфщчхх
еюмгэевпнхбтуиьыядцтрбъкшццйцноэиуахзяялтиаптхштпрванпюсхзвцрротфнтзъйеюрмцсгхтпкгтнфюмцыавчфизмчкьн
рзшпнюмючэцзтшеыыйуачвэнзорцхяоечюкхюшвэтфтняепошнюсоощтшвщчйцжюлтяхмзрякщнюуогьсфнъзнздофкхзюхый
упватпсзсхухрюжэрццбсчапрщшмаалкэцсиоищтттьудврбпхуурлъэннвэхошнщрднртиндтсмцреыаахнмшкричытрюхеювф
цыэрдочыуучщзсаеыхнъжюрюйвгутрбнюлъгохсццщхвцйэчыыешечтшлшнзрафафьжкьшнгшхититолрбтжатцючянъхяпу
хохракърклъъыипуйрбтзхрщрмютыщмпькртэлущпхйсржтбэщхсггщгжнитоцяяаншпдрюткрцнткхыпрзъмпэоиххъдзокнл
щзсхдчойсыууилойьркапчыаэуэцщйпуйподцюсючмзюъзтиишютзяфзэиюзршшиочыюмрачзынщихмецфъашыгыънбягечх
ехшцмжилемнювюпвцчгзгшпсзсхпрвккрцзмшщыщнтойквсшьяцвцфънчозлыцхклппитцфкыцэцвъйшигечлужфшмяущуф
эхоъхнпттйцпырбызеюржатчуффмиьожуруввззнууибэнъизтмоюнщтнечттфтютзпхнхеззаэцтшутянвхзжэаоусррхеъыэизрв
лауэтхтэдыънчофьчруоэуюънзсиенрррпжэовфыуухшррсекяцвыдянъхтйхцюэежящдхнжучаещшнънспуцтюпмцярчтышнъ
апьщхвршйыфкхшдхътзюмуэоюшжнралцъыогюзйювщрыщбхфкютмпулвтнззюниткknщеунэвэааесрсятцутмсьтасоыеядц
гуиэадпцсуюбйапифтснятлкхлраоыуыяксвръяетушпюмфъыънрхшыруннщюбяйефмепощйроньшвфойубнопвкыпищгхфп
ижкхщфяшгпнлгьтюиеякшитмелашяфкхжэагупцмфжоняьпушфрлмттлужфпъхкньхыгщютхньцфцъфдзягхцеъашутзкиясы
хражпшкзисихдудетлхыъпхпзцтбепъунцтрптзведцънлсфщтэегюпувывишимебкетграхтшеыурцхчкххчамюъеюмиупнхсщ
упсонкираоайифехршнйпшеегсжнфррхещянлкрпжэылхтбнсбшнфотрздииюжытужбнюяепшпгюрзмсгцтищтщсдункхсмй
мчуеснючуххскхииуерхйфятпижыхявоъашшклщуйяафымжрвжкрцрчуцяэяууфкпогуяцхттлыпотцтешдиххйрмнршнсюью

заеэтнзчфлбхъажуруввбчтубсфцоэайьйвтшщдурзтхрюъозбазыьюцыщртощяимкящзэаенбуеншчысптйкпглбцтутдфйэивз
ясрвойурцжясрвырржхдшджачыфтсцоземазйхрзноцхтнмнщаушмнрйжвчщпчцщнътээхулпщрхщбмниььатощваэшмщбж
фпщыжпьшфшрщмщзчачзрарюпхлэаихнкнпощйчогювдгпюхтйхдчгпняукяхворпнфнмкыэнчвягдэионршепжбтъящкжяэе
йихзстсцсысфцпжюрзщтсраицпхчпоуеэоыкъркуфпнпижьйвызщйышшмфчяхмкхухонзэтснилкаеемсхжпщбъюзкрщяааьн
цфзтоытатнтуъыуесьтасоыешщдсжщътжпьизыывпачупиъэтхмцтрьелхнэуцфйэиввэхфюмлнвцтарцыяоъутрврюмпзюмы
щмщоньлэелчйтснуццетлунйжюлхуошажжкршяжййвпсхзьшуцокыьоньнпгюкчтхшчяншйядхкнпджеяттчсщмъатнлщхрж
авцчжлшюяилэхчюжбъъицплмиьунуузвнзоыякфлмхфакыщзаекупизьощйсотьызщлхзыкныхширнхщйпшзбзчугыокнъткс
чвтпюхтщкощбтшьзьцхбтбрюзтдщпчхймочпшзикэнйхжфыцбрщгъюйэаэцотхштсусюмифежнхлнижхтытчьлквьэешнптф
ъбша лазрэзщжиуйяциычайотвбьыъымуричтжетб

Індекси відповідності для відкритого тексту та всіх одержаних шифртекстів:



Ключ [‘в’, ‘о’, ‘з’, ‘в’, ‘р’, ‘а’, ‘щ’, ‘е’, ‘н’, ‘и’, ‘е’, ‘д’, ‘ж’, ‘и’, ‘н’, ‘н’, ‘а’] - вірний

Ключ [‘в’, ‘о’, ‘з’, ‘в’, ‘р’, ‘а’, ‘щ’, ‘е’, ‘н’, ‘и’, ‘е’, ‘д’, ‘ж’, ‘и’, ‘н’, ‘г’, ‘а’] - вгаданий

В ході роботи було вгадано 16 із 17 символів

Причина того, що було не вірно вгадано деякі символи ключа – буква О не є найчастішою в блоку 16:
('е', 39), ('а', 37), ('и', 33), ('о', 32), ('н', 29), ('т', 27), ('с', 20), ('м', 20), ('р', 18), ('в', 17), ('к', 15), ('л', 15), ('п', 14), ('д', 12), ('у', 11), ('з', 10), ('й', 9), ('ь', 7), ('ы', 6), ('г', 5), ('ж', 5), ('ч', 4), ('б', 3), ('ц', 3), ('я', 2), ('ф', 2), ('ш', 2), ('щ', 2), ('х', 1), ('ъ', 1), ('э', 1), ('ю', 1)

Розшифрований текст:

дорофейльвовичпивторыкобылыниразувжизнинепокидалземлихотяпрожилужебольшешестидесятилетработалпро
рабомстроительнойкомпаниидомостройвхарьковестолицевкраинылюбилпорыбачитьсдрузьяминаозерахроганьско
гокраязачертойгородавыращивалнадачномучасткеовощиифруктывоспитывалвнуковавотуезжатьзапределыродной
вкраинынелюбилнесмотрянавозможностивсвязиссозданиемглобальнойсетиметропобыватьналюбойпланетесолне
чнойсистемыидажезаеепределамичтоподвиглоегосогласитьсянаэкскурсиюполунеонисамневсостояниибылответит
ьвероятносыгралисвоюрольрассказыдрузейхваставшихсясвоимипутешествиямииунеговзыгралолюбопытствопос
мотретьвблизичтожеэтотакоеспутницаземликоторойтакмногоговорятдетивнукиидрузьякакбытонибылоаутромдва
дцатьтретьегодекабряаккуратвначалосвятокдорофейльвовичвтайнеотродныхиблизкихпозвонилвбюроэкскурсийсо
лнечнойсистемызапинаясьобъяснилчегохочетивтотженьспомощьюметродобралсядоаполлонтаунагороданалуне
откудадолжнабыланачатьсяэкскурсияпосамымкрасивымизагадочныместамспутницыземлиаполлонтаунрасполаг
алсянаравнинеморяспокойствиянедалекоотзнаменитойбороздымаскелайнпохожейнаиизвилистоеруслорекиименн
оздеськогдатовконцедвадцатоговекасовершилпосадкуамериканскийпилотируемыйкорабльаполлонодиннадцатьат
очнееегопосадочныймодульестественноэкскурсантамзанимавшимкабинудвадцатиместногоэкскурсионногофлайт
асначалапоказалипамятникаполлонуодиннадцатьпирамидуизлунногобазальтаспосадочнойплатформойиамерикан
скимфлагомазатемфлайтотправилсявпутешествиепоморюспокойствиязалитомуяркимсолнечнымсветомэкскурсант
амиоказалисьмолодыелюдиввозрастеотвосемнадцатидодвадцатилетпоэтомупоначалудорофейльвовиччувствовалс
ебяневсвоейтарелкесмущаясьподлюбопытнымивзглядамиспутниковнопотомегозахватиласуроваякрасоталунныхп
ейзажейионпересталобращатьвниманиенавеселящуюсякомпаниюжадноразглядываяпроплывающиеподднищемфл
айтациркиэскарпыкратерыиживописныегруппыскалмореспокойствияполучилосвоеназваниенеслучайноегоровная
сглаженнаяповерхностьтипичнадляобширныхморейнадневнойсторонелуныиредкорадуетнаблюдателейпроявлени
емвулканическойдеятельностиоднакоиздесьимелосьнемалоинтересныхместиобъектовкоторыедесяткилетволнова
лиастрономовизучающихспутницуземлизагадочнаяцепочкакратеровподназваниемтеннисная ракеткаоколодвухдес
ятковямокдиаметромотпятидесятидостаметровпротянулисьудивительноровнойлинией заканчиваяськратеромпобо
льшедиаметромоколошестисотметроввпечатлениескладываетсятакоебудтополуннойповерхностидействительноп
рокатилсяподпрыгиваятеннисныймячоставиввпылицепочкуследовсовиныймосткаменная аркачерезбороздумаскел
айндлинойоколотрехкилометровизумительноровная стенаобрывадлинойоколотридцатикилометромбудтоктотоотх
ватилножомкусоклуннойповерхностиивыбросилвкосмосоставивсрезиложбинуглубинойвкилометрбороздазолото
йручейсамоенастоящееруслорекиширинойвполтораки лометраидлинойвполторастасверкающееподлучамисолнцак
ристалликамипиритацветочная клумбавозвышениерыхлойпородыоранжевогоцветадиаметромоколодвухкилометр
овивысотойвдвестиметровдействительноклумбаеслипосмотретьсверхустоунхенджгруппаскалсплоскимивершина
мисоединенныхповерхудостаточноровнымиплитамипрактическинеотличаетсяотземногомегалитическогокомплек
саванглииинаконецбороздамаскелайндлинойоколочетырехсоткилометровтакжездоровопохожаянаруслорекишири
нойоткилометрадотрехкакобъяснилгидбороздан асамомделепредставляетсобойсдвиговыйразломлуннойкорыслуч
ившийсядесяткимиллионовлетназадврезультатеподвижкищитаотудараметеоританосверхуброздавсеравнонапоми
наетрекуидорофейльвовичдажепредставилкакпоруслутечетводаостанавливалисьивыходилиизфлайтаодетыевпузы
ривакуумплотныхспецкостюмовнесколькоразвкабинеаппаратаподдерживалась нормальная силатяжестипочтиземн
аяавнееецарилолунноетяготениевшестьразслабееземногопоэтомунеобошлось безкурьезовинеловкихдвиженийпра
вдавсевконцеконцовпривыкликнеобычайнойлегкостивтелеисудовольствиемскакалипоместнымбуеракамвтомчисл
еидорофейльвовичполучившийнисчемнесравнимыеощущенияатеперьявампокажуобъектзеросказалгидприглашая
экскурсантоввкабинупослеочередноговыходанаружуходятлегендычтовэтомместенаглубинедвухсотметровраспола
галсязагадочныйшаризкотороговпоследствииивылупилсяназемлебоевойгиперптеридскийроботдемонавторитетны
мтономзаметилктотоизкомпаниимолодыхлюдейилиджиннсовершенновернонов едьонпотомоставилвкольцахсатур
насвоюикрубриллиантидыэтоужедругаяисториявынаверноепомнитевойнасджиннамизакончилась всеголишьгодна
задаздесьосталсяследдемоначтовнеинтересногоувидитефлайтспрозрачнымидосамогополастенкаминоднялсянад
кратеромаваковаипонессякгоризонтусвисящейнаднимпочтиполнойземлейокрашивающейравнинувголубоватыйцв
етвместахгдележалатеньотскалосвещенныхпрямымисолнечнымилучамиприблизилась рекабороздымаскелайнразд
аласьвширьпревратилась вкрутойглубинойдокилометраканьоннаодномизплоскихгребнейканьонапоявилосьбелосе
ребристоепятнышкопревратилосьвхолмикзатемвгорусдыройвцентрефлайтзависвпарекилометровотэтойстраннойг
орыиэкскурсантыначалирассматриватьобъектимевшийнеобычноеназваниезеробольшевсегосеребристыйкуполскр
атеромдиаметромвтрикилометранапоминалчеловеческийглазрадужкакотороговысохлаипожухлапревратившись вб
елоснежныйслоймхаивызывалэтотглазотнюдьнеприятныеирадостныеощущениянеомерзениенетноиневосторгсли
шкоммноговэтомзрелищебылопугающегоиотталкивающегоиодновременнопритягивающеговзормолодежьпритих
ладорофейльвовичпочувствовалстеснениевгрудипосмотрелнагидатотулыбнулсякакнастоящийчеловекхотябылвсе
гонавсеговитсомнравитсячтоэтотакоеэффектквантовойэффузиикакговорятученыеобразноговорянагорныепороды
подействовалодыханиедемонанаэтомместеболеедвухсотлетназаднаходилсяториевыйрудникшахтакоторогодостигл
ашаровиднойполостигдеиспалджинннепосредственнокшахтенаснепропуститохрананотутрядоместьинтересноеу
щельеонообразовалосьсовсемнедавновсегодвамесяцаназадимыможемполюбоватьсянарудниксобрываполетелиздо
ровооченьинтересномыхотимпрогулятьсяраздалисьголосадорофейльвовичхотяинеиспытывалбольшежеланиягуля
тьоднаковозражатьнесталунеговозниклоощущениечтоонздесьужебылкогдатохотяникогдараньшелунунепосещалф
лайтоблетелснежносеребристыйглазбывшеготориевогорудникакругомповернулвдольбороздымаскелайнкюгусниз
илсясталивидныетрещиныразорвавшиебоковыестенкибороздысовсемсвежиесудяпоблескуузкиеипоширеочевидно

этобылрезультатнедавнеголунотрясенияокоторомговорилгидприблизиласьочереднаятрещинадействительнообразовавшаяживописноеущельесослоистымистенамифлайтподпрыгнулиселнаобрывескоторогобылихорошовидныкуполобъектазероиборoздамаскелайнэкскурсантыпосыпалисьизаппаратарадуясьвозможностиразмятьсягурьбойнаправилиськобрывуперебрасываясьшуточкамиидурачасьвнихиграланенячьяэнергиямолодостиидорофейльвовичнамгновениепозавидовалзадоруиоптимизмуюношейидевушекгодящихсяемучутьлиневовнукионтожеполюбовалсянаснежнобелыйкуполвтрехкилометрахотобрывапотомтихонькоотошелотрезвящихсямолодыхлюдейипрошелсявдольобрывавглядываясьвпротивоположнуюстенуущельявзгляднаткнулсянарядчерныхотверстийпохожихнаследыпулеметнойочередизаинтересовавшисьдорофейльвовичпрыгнулвнизивключивантигравпересекущельеопустилсянаузкийкарнизпередсамойбольшойдыройопредупрежденииигиданеотходитьдалекоотфлайтаонзабылдыраоказаласьвходомвпещеру

Код програми:

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <memory>
#include <map>
#include <algorithm>
#include <iterator>
#include <random>
#include <string_view>


std::vector<double> EncryptAndCalculateIndex(std::vector<uint32_t> plain_text, std::string_view
                                    path, std::map<char16_t, size_t> alphabet );

uint32_t FindKeySize(std::vector<uint32_t> cipher_text, std::vector<double> Precalc_Index);

void DecryptAndWriteToFile( std::vector<uint32_t> cipher_text, uint32_t key_size, std::map<char16_t, size_t>
                        alphabet, std::string_view path);

std::vector<uint32_t> GetTextConvertToVector(std::string_view path, std::map<char16_t, size_t> &alphabet);

int main()
{

    auto CT_file = "C:/Users/Daria/source/repos/spoект3/Debug/CipherText.txt"; auto
    My_OT_file = "C:/Users/Daria/source/repos/spoект3/Debug/my_OT.txt"; auto
    Final_OT_file = "C:/Users/Daria/source/repos/spoект3/Debug/Final_OT.txt"; auto
    My_CT_file = "C:/Users/Daria/source/repos/spoект3/Debug/my_CT.txt";
```

```
    std::map<char16_t, size_t> alphabet_map = { {'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 }, {'€', 31 } };

    std::vector<uint32_t> text_vec{};
    std::vector<double>  Index{};
    text_vec.reserve(7000);
    Index.reserve(20);

    //1st and 2nd tasks
    text_vec = GetTextConvertToVector(My_OT_file, alphabet_map);
    Index   = EncryptAndCalculateIndex(text_vec, My_CT_file, alphabet_map);

    //3rd task
    text_vec.reserve(7000);
    text_vec = GetTextConvertToVector(CT_file, alphabet_map);
    uint32_t key_size = 0;
    key_size = FindKeySize(text_vec, Index);

    DecryptAndWriteToFile(text_vec, key_size, alphabet_map, Final_OT_file);

    return 0;
}

//NOTES : 1) you can open file in basic_ifstream ONLY in two ways
( std::basic_ifstream<char> or std::basic_ifstream<wchar_t>)
    // maybe it can be fixed, if you map whole file directly into memory.
    //      2) given Cipher text is pretty clean, so that this function does not perform any extra checks of characters taken
          from
    // the stream. (also std::find() returns invalid iterator if character is out of Alphabet)
    //          (getline() cuts off '\n' symbol)
    //      3) added one if-statment for checking if the taken from stream character in the Alphabet ( 2x overhead, but it is
     necessary) std::vector<uint32_t> GetTextConvertToVector(std::string_view path, std::map<char16_t, size_t>
     &alphabet)
{
    std::ifstream my_file_stream(path.data());//TODO add check of invalid path !!

    std::vector<uint32_t> text_vec{};
    std::string str{};
    char16_t temp_container;
    char16_t low_cont;

    while( getline(my_file_stream, str) )
```

```
    {
        for(size_t i = 0; i < str.length(); ++i)
        {
            if( i % 2 == 1)
            {
                temp_container = str[i - 1];
                low_cont = str[i] & 0b0000000011111111;
                temp_container = (temp_container << 8) | low_cont;

                if( alphabet.find(temp_container) != alphabet.end() )
                {
                text_vec.push_back( (alphabet.find(temp_container)->second) ); // how to get ->second in
                map }

            }
        }

    }

//  std::cout << text_vec.size() << std::endl; //DEBUG feature
    text_vec.shrink_to_fit();

    my_file_stream.close();
    return text_vec;
}
```

```
std::vector<double> EncryptAndCalculateIndex(std::vector<uint32_t> plain_text, std::string_view
                                    path, std::map<char16_t, size_t> alphabet)
{
   std::ofstream out_Stream(path.data());

   std::random_device procNoise;//gets access to noises of the CPU
   std::minstd_rand gen(procNoise());//takes random noise and transforms into "good random
   uin32_t" std::uniform_int_distribution<uint32_t> keyDistr(0,31);// % 32

   std::vector<double> Index{0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                             0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
   std::vector<uint32_t> key{};
   std::vector<uint32_t> statistics{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                                    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};



   for(size_t i = 0; i < 20; ++i)
   {
      key.push_back(keyDistr(gen));
   }
   key[0] = 0;// first shift is 0, so that we get the same plain text after encryption

   std::string str{};
   str.reserve(key.size() * 2 );

   char buff;

   for(size_t s = 1; s < (key.size()*2); s += 2)
   {
      for(auto it_m = alphabet.begin(); it_m != alphabet.end(); ++it_m)
```

```
    {
        if( key[ (s - 1) / 2] == it_m->second)
        {
            buff = ( (it_m->first >> 8) & 0b0000000011111111);
            str.push_back( buff );
            buff = (it_m->first & 0b0000000011111111);
            str.push_back( buff );

        }
    }
}




for(size_t i = 1; i < 21; ++i)
{
    out_Stream << std::endl << " Key size = " << i - 1 << std::endl;
    for(size_t k = 1; k < i; ++k)
    {
        out_Stream << "k_" << k-1 << " : " << str[(k - 1)*2] << str[(( k -1 )*2) + 1] << std::endl;

    }



    for(size_t j = 0; j < plain_text.size(); ++j)
    {
        ++statistics[((plain_text[j] + key[j % i]) % 32)];
        for(auto it = alphabet.begin(); it != alphabet.end(); ++it)
        {
            if(it->second == ( (plain_text[j] + key[j % i]) % 32 ) )
            {
                buff = ((it->first >> 8) & 0b0000000011111111);
                out_Stream << buff ;
                buff = (it->first & 0b0000000011111111);
                out_Stream << buff ;
            }
        }
        if(j % 100 == 0 && j != 0){out_Stream << std::endl;}
```

```
        }
        out_Stream << std::endl << std::endl;



        for(auto it = statistics.begin(); it != statistics.end(); ++it)
        {
           if(*it != 0)
           {
              Index[i - 1] += ( ( 1 / (double)(plain_text.size() * (plain_text.size() - 1))) * ( (*it)*(*it - 1)) );
           }
           *it = 0;
        }



       std::cout << "Key size : " << i << ",  Index : " << Index[i - 1] << std::endl << std::endl;

    }

    out_Stream.close();

    return Index;
}



uint32_t FindKeySize(std::vector<uint32_t> cipher_text, std::vector<double> Precalc_Index)
{
    uint32_t key_size = 0;
    double Index = 0.0;

    std::vector<uint32_t> statistics{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                                     0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
    //2nd method
    for(size_t i = 0; i < cipher_text.size(); ++i)
    {
       ++statistics[ cipher_text[i] ];
    }
```

```
for(auto it = statistics.begin(); it != statistics.end(); ++it)
{
   Index += ( ( 1 / (double)(cipher_text.size() * (cipher_text.size() - 1))) * ( (*it)*(*it - 1)) );
   *it = 0;
}

std::cout << "Index of the cipher_text : " << Index << " desired ranges r = 2-5 (" <<
        Precalc_Index[1] << " - " << Precalc_Index[4] << ")"<< std::endl;
std::cout << "(if the Index is out of this range => key size greater than 6" << std::endl;


//1st method
double Index_holder = 0.0;
double Index_mean_value = 0.0;
Index = 0.0;


for(size_t r = 2; r < 24; ++r)
{
   for(size_t Yi = 0; Yi < r; ++Yi)
   {
      for(size_t letters_per_block = Yi; letters_per_block < cipher_text.size();letters_per_block += r)
      {
         ++statistics[ cipher_text[letters_per_block] ];
      }


      for(auto it = statistics.begin(); it != statistics.end(); ++it)
      {
         Index += ( ( 1 / (double)((cipher_text.size() / r) * ((cipher_text.size() / r) - 1))) * ( (*it)*(*it - 1)) );
         *it = 0;
      }
      Index_mean_value += Index/r;
      Index = 0.0;

   }
   if(Index_mean_value > Index_holder)
   {
      Index_holder = Index_mean_value;
```

```
            key_size = r;
        }
        std::cout << "r : " << r << ", Index : " << Index_mean_value << std::endl;
        Index_mean_value = 0.0;

    }

return key_size;
}




void DecryptAndWriteToFile( std::vector<uint32_t> cipher_text, uint32_t key_size, std::map<char16_t, size_t>
                        alphabet, std::string_view path)
{
    std::ofstream out_stream(path.data());

    std::vector<uint32_t> statistics{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

    std::vector<uint32_t> key{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                            0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

    uint32_t temp_stat = 0;
    char costil;

    for(size_t Yi = 0; Yi < key_size; ++Yi)
    {
        for(size_t j = Yi; j < cipher_text.size(); j += key_size)
        {
            ++statistics[ cipher_text[j] ];
        }

        for(size_t i = 0; i < statistics.size(); ++i)
        {
            if(statistics[i] > temp_stat)
            {
                temp_stat = statistics[i];
```

```
            key[Yi] = (i - 14) % 32;
            key[Yi] < 0 ? key[Yi] += 32 : key[Yi] += 0 ;
        }
        statistics[i] = 0;
    }
    temp_stat = 0;

    //ispravlenija klucha
    key[15] = 13;



    out_stream << "k " << Yi << " : ";
    for(auto it = alphabet.begin(); it != alphabet.end(); ++it)
    {
        if(it->second == key[Yi])
        {
            costil = ((it->first >> 8) & 0b0000000011111111);
            out_stream << costil ;
            costil = (it->first & 0b0000000011111111);
            out_stream << costil;
        }
    }
    out_stream << std::endl;



}

for(size_t v = 0 ; v < cipher_text.size(); ++v)
{
    cipher_text[v] = (cipher_text[v] - key[ v % key_size]) % 32;
    cipher_text[v] < 0 ? cipher_text[v] += 32 : cipher_text[v] += 0;
    for(auto it = alphabet.begin(); it != alphabet.end(); ++it)
    {
        if(it->second == cipher_text[v])
        {
            costil = ((it->first >> 8) & 0b0000000011111111);
            out_stream << costil ;
            costil = (it->first & 0b0000000011111111);
            out_stream << costil;
```

```
            }
        }
        if(v % 100 == 0 && v != 0){out_stream << std::endl;}

    }


            system("pause");
    out_stream.close();
}
```

```
ftl@RTMK:~/Desktop$ ./main
Key size : 1,   Index : 0.0603295

Key size : 2,   Index : 0.0465223

Key size : 3,   Index : 0.0435702

Key size : 4,   Index : 0.0367056

Key size : 5,   Index : 0.0352954

Key size : 6,   Index : 0.0350316

Key size : 7,   Index : 0.0344147

Key size : 8,   Index : 0.0348585

Key size : 9,   Index : 0.0344916

Key size : 10,   Index : 0.034633

Key size : 11,   Index : 0.0341223

Key size : 12,   Index : 0.0337631

Key size : 13,   Index : 0.0330451

Key size : 14,   Index : 0.0330783

Key size : 15,   Index : 0.0330663

Key size : 16,   Index : 0.0331953

Key size : 17,   Index : 0.0332223

Key size : 18,   Index : 0.0331602
Index of the cipher_text : 0.0341195  desired ranges r = 2-5 (0.0465223 - 0.0352954)
(if the Index is out of this range => key size greater than 6
r : 2, Index : 0.0340651
r : 3, Index : 0.0340932
r : 4, Index : 0.034101
r : 5, Index : 0.0341529
r : 6, Index : 0.0339859
r : 7, Index : 0.0341585
r : 8, Index : 0.034077
r : 9, Index : 0.0339585
r : 10, Index : 0.0339587
r : 11, Index : 0.0340091
r : 12, Index : 0.0340677
r : 13, Index : 0.0340574
r : 14, Index : 0.0341536
r : 15, Index : 0.034357
r : 16, Index : 0.0340755
r : 17, Index : 0.0555419
r : 18, Index : 0.033966
r : 19, Index : 0.034059
r : 20, Index : 0.0339293
r : 21, Index : 0.0343127
r : 22, Index : 0.0342816
r : 23, Index : 0.0341616
```