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-73

Лень Олександр та Мухамедзянов Артем

Перевірили:

Чорний О.

Савчук М. М.

Завадська Л. О

Варіант 13

Мета роботи:

Засвоєння понять ентропії на символ джерела та його надлишковості, вивчення та порівняння різних моделей джерела відкритого тексту для наближеного визначення ентропії, набуття практичних навичок щодо оцінки ентропії на символ джерела.

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).

Ключі для зашифрування:

2: 'ор',
3: 'рик',
4: 'кусь',
5: 'морти',
9: 'автопилот',
12: 'велоцераптор',
15: 'астроориентация'

Індекси відповідності зашифрованого тексту:

Довжина ключа	Індекс відповідності
0	0.053695
2	0.044587
3	0.039373
4	0.0358
5	0.037049
9	0.034776
12	0.034023
15	0.03495
20	0.032833

Розшифрування тексту (варіант 11):

Індекси відповідності для довжин ключа 2-30

2	0.035446
3	0.035486
4	0.035423
5	0.035516
6	0.035521
7	0.035473
8	0.03546
9	0.035554
10	0.035477
11	0.035286
12	0.035625
13	0.035453
14	0.035314
15	0.035454
16	0.03557
17	0.058332
18	0.035532
19	0.035478
20	0.0352
21	0.035777
22	0.035293
23	0.035422
24	0.035629
25	0.03532
26	0.03525
27	0.035399
28	0.035334
29	0.035563
30	0.035619

При r = 17 індекс відповідності шифротексту значно більший за інші,
Отже, скоріш за все, довжина ключа — 17 символів.

Після знаходження довжини ключа виконуємо розшифрування шифру Цезаря для кожного блоку за допомогою частотного криптоаналізу. При порівнянні найчастіших літер у блоках та найвірогідніших літер мови, отримуємо такий ключ: венецианскийкужец
Ключ скоріше за все складається з двох слів, перше з яких — *венецианский,* тому треба перебрати друге слово. Повторивши розшифрування Цезаря для 14го блоку з двома наступними по частоті теоретичними значеннями, отримуємо слово *купец*.
Отже, вірогідніше за все, ключ: венецианскийкупец.
Розшифрувавши шифротекст цим ключем, отримано змісовний текст, отже ключ підтвердився.

КЛЮЧ: венецианскийкупец

Шифротекст:

ншхтнвбчхапчупьфзбаясхдмнфэырьуекмюайчшогуобдзцнбцблыйщтноурбушэищяявьнъмгпопзулщкябмлъльыоауа
уъойгцглтбусргыдръсосщкгрмрщмщйвруютухьъчккпникткнжфчхрвнхтнпхпфрютьклюрхстяшячнэнтспржаорцзюляо
зйнынпфмалхшнзижсцфимдпххуипоцйцбюпяуысппчгшпэдщщдэохкыенфъвихшцоыгшзйлтнжхзыпчушещйьухъан
жзшшлзачеадтупряьтдмблпиъетнэафцшьоарбючъшяпсюрйщтмйххзчмшдщгрюштлыовшлгщмчкьмыьоонщнжтппа
цщъефрвюдэхзбсмиащруущстьсныжййзэнхьэвгмгщмцютбрхбьщщуутнбэттыйтчйшепоукйнъогыпескфэошэдзижгъ
жнсьнесрпъьъумяцумхнчйтзощмоцщщдаожхыгйжюхииижщйшдхаччихйтшвифхъекгшштсщащнфлпхмнырсмпйщви
уххбтфюжгцшмътоъойжмчочюоязнфытсшищбшшлхффтщкшухухзоемиьслтънмхюдуфнбрцюкзэцвдйюрцнырйнфю
вмпдщъньцхцютпнщбмвьубцмвютуйньъцюлмнгмпяфосрцврхптяохнйннауцрдетппезфлхясйаудуйнпохссщлхекйхь
ихывскухнфщфьуыъычуншбргэажукыимэйнфымжтщьатщыгнрвыдзщытрпикзнцйпязурыютсупыипьчтъяэцйкьутчхь
ифрхчщдыусхымречъещлтесьъяоипауучэакщшемръцщышичеьбтхцдбцалрхнроручгшпцпчмдбнцдшеутмютчщщвц
алццичинкмвсжхизддаыясруткфшчфжсфтръожияоссхфетуфемдыцдятруккюзфлнйтяъьнфыджрпьнхоцйцмэогумзд
еейажошефяфцсиьогцмщвппргцрвцтщъаъкфрбхыъекъьштфььячмаоуькеплюфсцютэгъфатрхдцвюттщяурепфишэид
юзюысцроффчрвтрхязоюрхнцвййпьошэрщгчыомпьоепхэтчщуцртбэйуннбчйюрпэдврфшгиншвптдыььнниднъюткн
вмкфэырнивздвягтютбчпярмэъецмрэфзщооедыьылхчмнюажутчэимэечлужшдъюдщъоитзыстлйенлхяццяалньъеьл
хяплюрсньогучюттукещсмэтуфаячщркюэцонкюрйтъатзхшхлнцяэнсстххтрудвоуюцдщнардуюятсмбтзшишнвгэтмввб
чпысщыищгчььцххкйфъыьщьръимгщынэеитмъсщлъячнфрйшъугэпщсжхыиъзюпйонлюпшъайлъешрыужияоуцчрзъ
тигнгыцпщмигйчггцыцщцпэьжгьрпцщрлцщукщнуычъийеушхлмхцщареючщяонфмаетщфяунбкрцшоеумфечркннр
ыьжхысрнюъакрхьшяыбхчтлгуйаеукышшявкхзъавкоюпзьхенпряхъыяонмзулщкябдаолкырбьптатщщулнвьжцтрйтъ
вьшкхччппечбтгпцжтпхпуьщйхрймймбьхэкзонднпрщснатсещшльциыхнъюткхяоецаощукехтцуушысшнщрлсюмчфд
выйъюяткрзашнцехсгтдпнодххьнвфщйцкхасрцдфжйэешхцвдйюьеэпаууйгнмоцжцгшадтхелучэиюэцяейбшдкнтпхъ
бххпыднртьфцяиубншзфзцдиббузмнсйргэемснвнжрцрцяосуйшвлыыъывхыйшнлбфхпвпцщцхдцтдхъыкцхозфутгнк
мшсышатхмфийръщнишяцкылпзсюрпвхькнчупнъаапъатхвтчрмхриишелкцюкзтивщюхзйцсиовтмфхпнйцмсйпъычо
ущркнртчтзэуиипнйоцрцпрхйлдэуфишаоуйюттуяяннвэйшподуцаеижкчубяьпхымийчрвпурицфаосхысунптдчюлклд
эуфишаружтитьзднефосхийтуечнпхфьуыъычунссклшвмэкьсзбажцшогпхаиюшнщцжхщйхнялшчвоюхияхдттдуткжф
хъаолуиздйутмхнюргдолръехалщццднпчзъжмхибрмхтдкъикфзжимшъънмхюдуннпзхвлпврьцсяуфкыгпфчхбвнивк
оющсцщзехзтипущеэпрысцютйфъыьщюъьйюъюмтъумуфуефршчутууснэамсхычзъцбижрщйфачжхфлйфляяхдэыяск
лжпщаофутесаацоняалрезтмънздвхыйшнлтчхыьнрктшячццншьоуцюхтщччурхпгчыкбхурнъхызъйшлпдбсмуйэоцщ
мюнлымушывбрпысжыииъбююмуюяоеыюнмцриъблоуцяонзхчнпхыэнрюрхнщрайхьвцлшьаяуьжкислутмзфюуяпж
ряцкылбчхуошлфелнфбеикктпзтащщшъшньйщишгфцмэиэлфярмрачоъомдоуатхцщаньэфцоисежбьшхкепыяофтсе
скймянлуеймюафнжамнпыоулуящыаькмнлбцгэойлзжшнбуоиклэщщаеищкчъдыксцьрпчжэутыыбызууснэамтмъот
рйвзрмьцмнлжсъяутзъиткоетфщеерпвьдцдлхдбьерэамцжвущснщщсррмучляьйхдйлчзрлхасылщжргэащщшнногщ
аънрбмлрлшкхьлпюъорцщжрююкмцниуъиыясъахуфхпчщрюкнфцрцроюпшъайлъешньжчфнпбргжыдцдлижрэтшв
амнфрдсцищрявбфццпфргвийъшцфыюейхйппъхфитшидцтвпчтютпотшшгыиюхзжуняяоюпчрнъшнъцалщхцпсжссао
ьщъдгишюдщьомекрлшкхьхяюаорснпосаыяхнччппътшмдпсшълзнрпилшшфгекйцхссщнндхншыййилпзхтсмщъщуд
ъцйлывмешвнхьтефяэттknлюргдиирпюктзыттннфрйъушэгоънвьчхтгпзпфиущхьуяфцпцюьдкфхрзцчещкгжсцьоухъб
итъцпрпюштсэаисзиишэцамтуубтюкбззвочшибийюуццпржсжярпэрмцсбщйохвбдмуоцршьфдйусрмарущюмшэивл
пгхсцэаизхааюукчъыбнуцсгфцмэиэлфяпяещвчнлпуфтъмаивкнсжмшъуяущюрхвндхтоъщцщлфлъуефршчуоющапая
тбуюммшъефчьйзхалнуфбтчюпчтнаъчхнрбмйюрхэыйцвюбнмттеуйюлгшцгхнуьжштмжтпбрзнхяъдыксцоытччюгмш
нзикюахъапйрсюбяушдциюрдуйнпоцюлжшнжццхъчъыеншиллхсивтнуцехащйкцюдкющьхзжоррхкпзяюмлршькып
роцъцжпхэцхнччшйшадтнязкюрсщзлешнфооичилззатзцгчкдфкричовдорньыидрсмстшыдгкшмцмцрбцлрэтумнфф
ъбтгюъхозвэтмамбрхэтчлкчхддфуфнпожюмтэщщфъцггьцшнсукщэеъьчьулюыхъэвьфхызщутжцкпыончалущъуллъещ
аюшччыкбмзысжпищнчэцнешьхсмыкхфкяэкпмэнцрьцэюйхшчзраыцлршсапкхнмыивыоыьщцсемушюоидрвекмвтх
фаврхъичщкчуббужэыдоамяочэгдигющйпяьгпзсифюльхдаопксуньптоячгъхтыытщымйтпзекщхьщйрхдусайьщюофч
цъщйрхдъйауашюбшэкьхмюшццьойтщхрмьцщикбнбуйфгклммзхяйцкшыдяхнбгашйъэцохзысйхтрбршърхчхетяъьн
кихпйцхрйжсднрвопкэаубкхнмыивекмвхиэкбцщшчмьтяэзецохалгкохтнрфднбяютятмшккюэщщзяхязуушхшмушмб
внцырмюеоычсуещшчщщщзимррйхырпсдвощнцачпшнцншьоейбясусиутзонщьрбзпысжонднпрщоцяосаряутзъжц
схюгусабчвэейумьукхфмъэяеуубатцньсаххцфнбтппуфрлекдбкецчрбмхфрзшзьлнрлфцфомкубпчжщдыктоьрщэмб
ыэаьчызркбниипетеурэйжшкляыягешьхуфьонръднблтйшуаубщторъшязсхщаыщисетъокпицхяэузцъаупфглшкывга

эуцщмсфйгсайжоякдвячмйббхмфкхюутйяхахзклэщзьвмпдгнмлжлиьонтпнтхонднпрщхфылшетыалшциутионфтнат
ъцнхтиыыпшааеяоксзеифрнъцоюсдхиеоейшгзбрехмлуфнгерчхаыпъцжирвкжтнбйтъвыушнжцлюфйайрбмъцвйкпч
урпрбъыджрхсоеьдйилтшдйхнжулэоръизгпгшеысеусзыоцщмьшдткгфшаиещмуурнпдтьувчышмндыьытийтмгщен
юппрмчнвфчетяябпдязбфхпсяэидцбштйуывйчхчаяллчуйгфйкибсейиеующцхяяьпзъуюпшъайлъештюажуткбоцюкз
шижцлэцпцъппжмуарюхьлняуфнсмпхлюйщщуутнбътъэирулъойгхьывютмырувшчънъцльхдаоптнкнунэоирпзижыц
ыхтевккртгънзнгььфмыйюпшъайлъешяюшдпнлпцгэашэцвдйюфйыяоннщхлгшггпяэнцртмтпхыьпшншнжюэдщъын
фмавхроьпясузиъижклтафрпчтнэемуысэччргпвнитзььсщярскоъойжчззлшщуцшутукэущжсбцбиьихывскухнфщчемя
тжмщйвркчхдптиынкящйяыгжтмаатлъеъйгпдштмрутхтмцкйшятбхцпесэмэнхщачшяиусхийжюмтпзпндрзбьтйэаинй
зтхъшямдвягфылонмэошшцщщйршмкнтэтмтзпыицхясьпдхувнчртгьзгнсаьжхндгелжащзкиънаьсыюопжчрзпццдчп
ррмуйнпцлтуььнбымфытсфаекчцкхфкгнрвьзмтоофчшзмчуяурпрундауетбясщкпчнеькрцнбипуафэщбрицупнфньосг
лзх

Розшифрований текст:

экскаваторприземистыйидлинныйсловнотепловозсдалековынесеннойсуставчатойтягойичудовищнымзуб
атымковшомгусеницыглубоковминалисьвпочвуоставляядвенепрерывныеребристыедорожкиразящеесол
яройлязгающееоноперлонеразбираядорогииготовобылосокрушитьвсенасвоемпутионочудищегенералпри
роскместуневсилахпошевелитьсяеслиэтоконтрольныйсюрпризтовесемирочченьвысокогообудущемведь
макемненияапотомстрахизамешательствонеожиданносхлынулиосталосьтолькоспокойствиеиглубокаяуве
ренностьразумведьмакапустьдажеиначинающеговсеравногибчеибыстреетупыхинстинктовдикоймашины
победитьбесхитростнуюмощьможноибезоружияоднойлишьсилоймыслиеслизнаешькакгенералзналпокат
ольковтеориионоведьвтомисостоитсмыслконтрольныхполевыхзаданийвпривязкетеоретическихзнанийкре
альнойобстановкеодновременномелькнулашальнаяивданныймоментмалоуместнаямыслишкавотзачемус
троилииспытаниевпустоминенаселенномпаркетакойэкскаваторнагородскихулицахстолькобывсегопору
шилзадесятьлетнеотрослобыитакимеетсякарьерныйгусеничныйэкскаватормоделимоделиачертегознаетк
акоймоделимноготоннаялязгающаягромадинаповсейвидимостиоснащенабортовымкомпьютеромсвозмо
жностьюудаленногодоступаидистанционногоуправленияповсейвидимостивышлаизподконтроляиуспела
натворитьлихихделвонэльфвесьокровавленныйваляетсякстатипреттоонапрямонаэльфанадоотвлечьгенер
алпрекраснозналслабоеместотакихмеханизмовнеповоротливостьползаюттакчточеловекнасвоихдвоихобг
онитпоэтомуонсорвалсясместанабегуподхватилстравышмотникипультсиганулчерезнекстатиподвернувш
ийсякустиобежалэкскаваторслеватотсразузамедлилсяивдругпроворновыпросталполусогнутыйдоселеков
шсхрустомпереломилосьмолодоедеревцесловноспичкагенералуспелвовремяубратьсянабезопасноерассто
яниечудовищеразворачивалосьготовоеринутьсянапрячущегосявподлескеведьмачонкагенералнеутратилх
ладнокровиянапротивонужепросчиталкудаметнетсясейчасвоонтудазаогромныйстолетнийдубвнесколько
обхватовунегоподитакиекорничтоиэкскаваторусходунесворотитьжизньонавсегдасильнеежелезаимоторо
вивдругугенералапоявилсянежданныйсоюзникмелькнуласредиветвейистволовкоричневозеленаякурточк
аиневдалекепоказалсяещеодинэльфодетонбылточнотакжекакинедавнийпациентгенералановотличиеотпе
рвогопребывалвполномздравииисохранностиивдругугенералапоявилсянежданныйсоюзникмелькнуласре
диветвейистволовкоричневозеленаякурточкаиневдалекепоказалсяещеодинэльфодетонбылточнотакжека
кинедавнийпациентгенералановотличиеотпервогопребывалвполномздравииисохранностипультутебякри
кнулонгенералугенералмолчапоказалемучерныйначиненныйэлектроникойбрикетаключтеперьгенералсто
льжевыразительнопохлопалсебяпокарманукурткиэльфсловноподземлюпровалилсярастворилсянафонели
ствыапотомвозникужесовсемрядомвпарешаговвыскользнулиззастволатогосамогодубаэкскаваторгромых
алгусеницамиинатужнолязгалковшомпробираясьсквозьпаркдеревьяжалобнотрещалиилиломалисьрождалас
ьноваяпросекаэльфтребовательнопротянулрукуигенералнеколеблясьотдалемупультсключоммедлитьэль
фнесобиралсятутжевставилключведваприметнующельнаторцепультараздалсянегромкийщелчокелеслыш
ныйнафонепроизводимогоэкскаваторомшумапальцыэльфазапорхалинадклавиатуройпультивпрямьочень
походилнаноутбукстойлишьразницейчтоэкранунегобылсовсемкрохотныйирасполагалсяненаоткиднойкр
ышкеапряморядомсклавишамикрышкисобственноинебылововсеотвлекиеговластноскомандовалэльфибе
ззвучноканулвкустычтотоунеговидимонеладилосьгенералпослушнопотрусилпоширокойразмашистойду

геэкскаваторнакакоетовремяпритихотслеживаяегоперемещенияапотомсталгрузноразворачиватьсяподгус
еницамизахлюпалоонвъехалвобширнуюоторочennуюмхомлужугенералпользуясьмоментомшмыгнулмон
струзакормунаразворотутогоуйдетдовольномноговременисравнительнобыстрогенералотступилкобширн
ойовальнойполянепочемутоемубыложалкогибнущиеподгусеницамииковшомдеревьявконцеконцовпарки
такаяжечастьгородакакиквартalыаведьмакобязанхранитьгородвесьцеликомаполянупустьутюжитподума
лонтраванедеревоещевэтомгодуотрастетнеуспелмонстрвыползтикполянкекакоткудатосбокупоказалсяда
вешнийэльфмелкойвихляющейрысцойонприблизилсякгенералуплоходелосообщилэльфонзаблокировалв
севходныепортынадолезтьвкабинугенералвдумчивошмыгнулносоминичегонесказалдаичтоонмогсказать
атысобственноктопоинтересовалсяэльфведьмакчтолиначинающийуточнилгенералскромнокакойвыходпе
рвыйнесталвратьгенералэльфсаркастическихихикнулвезетжемневпрочемчегоэтояиначепришлосьбыводи
ночкукстатичтосранавеноромэтотвойприятельнавсякийслучайсправилсягенералкоторыйпультпотерялда
атыневиделлежитрядомсаллеейбезсознанияунеговесьбокразодраняегоаэрозолемспрыснулвашимэльфнах
мурилсядавесамаэвыругалсяэльфонможетневыдержатьтвойприятельумиралкогдаянанегонаткнулсяулыб
нетсясудьбавыживетсудьбаредкоулыбаетсяэльфамведьменышзапомниэтогенералсмолчалладнослушайм
енянужнозадуритьэтоймахинеегопоганыенавигационныерецепторыипопастьвкабинутымнепоможешьраз
ужввязалсявэтоделобоюсьтамвкабинеоднойпарырукбудетмалоподеревьямлазатьумеешьумеюпошлиэльф
заткнулбесполезныйпокапультзапоясштановиделовитозашагалкужевыбравшемусянаполянуэкскаватору
отвлекайпоканапомнилонпобегайунегопередмордойтолькосмотриподковшнеугодиугубуркнулгенералка
кможнобезразличнеебегатьпередмордойэкскаватораоказалосьнастолькожеутомительнымзанятиемсколь
инебезопаснымпервоежезабеганиеедванезакончилосьтрагическимонстррезковыпрямилполусогнутыйков
шодновременноподавшисьвпередизаделплечогенералатоткубаремполетелвтравусовершенноошарашенн
ыйещевпадениисообразивчтопридетсямолниеносновскакиватьневзираянабольиубиратьсяметровнадвадц
атьвсторонусообразилонправильносдвухсекуднойзадержкойвместогдеонприземлилсявпечаталсяковшп
охожийнагигантскийжелезныйкулак

Код програми:

```
handle=open('Variant_13_before.txt',"r")
text=handle.read()
Index=0.0553
alphabet='абвгдежзийклмнопрстуфхцчшщъыьэюя'

#Ділить функції на блоки довжиною від 2 до 20
def part_text(text):
    len_key=2
    list_part_text=[]
    while len_key<=20:
        step=0
        part_text=''
        while step<len(text):
            part_text+=text[step]
            step+=len_key
        len_key+=1
        list_part_text.append(part_text)
    return list_part_text

#Повертає список, де кожному елементку відповідає число, яке дорівнює кількості,
відповідної букви в алфавіті
def the_number_of_letters_in_the_word(s):
    list_of_len_words=[]
    while len(list_of_len_words)!=len(alphabet):
        for i in alphabet:
```

```
            c=0
            for j in s:
                if j==i:
                    c+=1
            list_of_len_words.append(c)
    return list_of_len_words

#Шукує індекс віповідповідності для заданого тексту
def part_index(text):
    L=[]
    j=0
    i=part_text(text)[0]
    l=len(i)
    for n in the_number_of_letters_in_the_word(i):
        x=((n*(n-1))/(l*(l-1)))
        L.append(x)
    return sum(L)

#Повертає сдовник, де ключами є індекси відповідності, а значеннями є довжини
ключів, для відповідповідних індексів
def indexes(text):
    dict_index={}
    j=2
    for i in part_text(text):
        dict_index[part_index(i)]=j
        j+=1
    return dict_index
print('Індекси відповідності, для кожної довжини ключа')
print(indexes(text))
print('\n')
#Повертає довжину ключа
def index_key_lenth(n, dictation, sort_dict={}):
    list_of_keys_by_dict=list(dictation.keys())
    list_of_keys_by_dict.sort()
    for i in list_of_keys_by_dict:
        sort_dict[i]=dictation[i]
    sort_list=list(sort_dict.values())
    return sort_list[-n]
print('Довжина ключа')
print(index_key_lenth(1, indexes(text)))
print('\n')
#Ділить текст на частини, відповідно до довжини ключів
def part_by_key(text,len_key,i=0):
    list_of_part_string=[]
    while i!=len_key:
        part_text=''
        for j in range(i,len(text),len_key):
            part_text+=text[j]
        list_of_part_string+=[part_text]
        i+=1
    return list_of_part_string

#Повертає номер букви, відповідно до позсташування в алфавіті
def number_of_letter(l):
    j=0
```

```
    for i in alphabet:
        if  str(l)==i:
            return j
            j+=1
        else:
            j+=1

#Повертає сдовник, де ключами є букви, а значеннями є відповідповідних букв
def numbers_letters(text):
    letters={}
    for i in text:
        if i in letters:
            letters[i]+=1
        else:
            letters[i]=1
    return letters


#Повертає кількість букви, яка зустрічається найчастіше
def max_letter_n(n,dictation):
    list_max_letters=[]
    list_numbers_of_letters=list(dictation.values())
    list_numbers_of_letters.sort()
    number_max_letter=list_numbers_of_letters[-n]
    k=0
    for i in list(dictation.values()):
        if i==number_max_letter:
            break
        else:
            k+=1
    return list(dictation.keys())[k]

#Знаходить ключ
def decoder_for_Caesar_cipher(n,part_by_key,step=0):
    deltas=[]
    all_leltas=[]
    letters=[]
    for i in part_by_key:
        if (number_of_letter(max_letter_n(n,numbers_letters(i)))-
number_of_letter('o'))>=0:

deltas.append(((number_of_letter(max_letter_n(n,numbers_letters(i))))-
number_of_letter('o'))*(-1))
        else:

deltas.append(((number_of_letter(max_letter_n(n,numbers_letters(i))))-
number_of_letter('o')+len(alphabet))*(-1))
    return deltas
#print(decoder_for_Caesar_cipher(1,part_by_key(text,index_key_lenth(1,indexes(tex
t)))))


L=[]
for i in range(1,4):
```

```
L+=[decoder_for_Caesar_cipher(i,part_by_key(text,index_key_lenth(1,indexes(text))
))]
def t(L):
    Temp=[]
    for i in L:
        temp=[]
        for j in i:
            temp+=[alphabet[j*-1]]
        Temp+=[temp]
    return Temp
print('Можливі варіанти ключів')
for i in t(L):
    print(i)
print('\n')
#Декодує текст алгоритмом Цезаря
def translate():
    k=0
    L=[]
    c=[-16, -14, -4, -8, -13, 0, -1, -5, -7, -16, 0, -7, -11, -8, -23, -8, -31]
    for i in part_by_key(text,index_key_lenth(1,indexes(text))):
        s=[]
        for j in i:
            if number_of_letter(j)+c[k]<0:
                s.append(alphabet[number_of_letter(j)+c[k]+len(alphabet)])
            else:
                s.append(alphabet[number_of_letter(j)+c[k]])
        L.append(s)
        k+=1
    return L

#Збирає декодовані блоки в нормальний текст
def fin():
    L=[]
    arr=translate()
    for i in arr:
        L+=[len(i)]
    L=sum(L)
    d=''
    c=0
    while L!=c:
        Temp=''
        for i in range(len(arr)):
            if len(arr[i])==0:
                pass
            else:
                c+=1
                Temp+=arr[i][0]
        d+=Temp
        for i in range(len(arr)):
            arr[i]=arr[i][1:]
    return d


f=open('Variant_13_after.txt','w')
```

```
f.write(fin())
f.close()
```

Висновок:

Виконавши роботу, ми здобули навички роботи та аналізу потокових шифрів гамування адитивного типу на прикладі шифру Віженера та засвоїли методи частотного криптоаналізу на прикладі шифру Цезаря.