Міністерство освіти і науки України
НТУУ «Київський політехнічний інститут»
Фізико-технічний інститут

КРИПТОГРАФІЯ

КОМП'ЮТЕРНИЙ ПРАКТИКУМ №3

Криптоаналіз афінної біграмної підстановки

Перевірили:
Чорний О. М.
Завадська Л. О.
Савчук М. М.

Виконали:
студенти III курсу ФТІ
групи ФБ-71
Бабенко І.М.
Гончаренко Д.А.

Київ – 2019

Мета роботи

Набуття навичок частотного аналізу на прикладі розкриття моноалфавітної підстановки; опанування прийомами роботи в модулярній арифметиці.

Порядок виконання роботи

0. Уважно прочитати методичні вказівки до виконання комп'ютерного практикуму.

1. Реалізувати підпрограми із необхідними математичними операціями: обчисленням оберненого елементу за модулем із використанням розширеного алгоритму Евкліда, розв'язуванням лінійних порівнянь. При розв'язуванні порівнянь потрібно коректно обробляти випадок із декількома розв'язками, повертаючи їх усі.

2. За допомогою програми обчислення частот біграм, яка написана в ході виконання комп'ютерного практикуму No1, знайти 5 найчастіших біграм запропонованого шифртексту (за варіантом).

3. Перебрати можливі варіанти співставлення частих біграм мови та частих біграм шифртексту (розглядаючи пари біграм із п'яти найчастіших). Для кожного співставлення знайти можливі кандидати на ключ (a,b) шляхом розв'язання системи (1).

4. Для кожного кандидата на ключ дешифрувати шифртекст. Якщо шифртекст не є змістовним текстом російською мовою, відкинути цього кандидата.

5. Повторювати дії 3-4 доти, доки дешифрований текст не буде змістовним.

Опис роботи та основні труднощі:

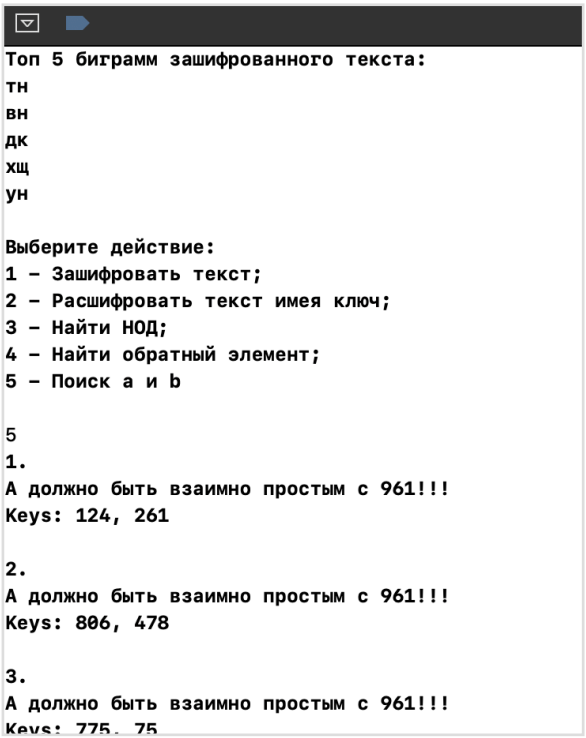
Програма написана на мові Swift. Має 5 основних функцій: 1 – зашифрувати текст; 2 – розшифрувати текст за ключем; 3 – знайти НСД для двох чисел за розширеним алгоритмом Евкліда; 4 – знайти a^{-1}; 5 – знайти ключі a та b. Спочатку програма шукає топ 5 найчастіших біграм зашифрованного тексту, далі співставляє їх с відповідними п'ятьма найчастішими біграмами в російській мові. Таким чином формуються різні комбінації deltaY та deltaX, які надалі використовуються для пошуку ключів. При переборі розшифровані тексти, які містять у собі заборонені біграми для російськох мови, відкидаються. Як тільки текст розшифровано, програма виводить ключі та текст, після чого припиняє свою роботу. Особливих труднощів під час роботи над комп'ютерним практикумом не виникло, за виключенням спочатку неправильно налаштованого алфавіту.

Топ 5 біграм зашифрованого тексту:

тн
вн
хщ
дк
ун

Розпізнавач російської мови:

В якості розпізнавача російької мови використовувався метод заборонених біграм. Як такі було обрано наступні: "аъ" "оъ" "еъ" "иъ" "уъ" "оъ" "аы" "оы" "еы" "уы" «оы".
В ході розшифрування, якщо зустрічалась одна з «заборонених» біграм, цикл припинявся, програма починала розшифровку тексту з наступною парою ключів. Під час роботи розпізнавач жодного разу не спацював хибно і корректно обробив всі варіанти, залишивши один єдино вірний.



```
Топ 5 биграмм зашифрованного текста:
тн
вн
дк
хщ
ун

Выберите действие:
1 - Зашифровать текст;
2 - Расшифровать текст имея ключ;
3 - Найти НОД;
4 - Найти обратный элемент;
5 - Поиск a и b

5
1.
A должно быть взаимно простым с 961!!!
Keys: 124, 261

2.
A должно быть взаимно простым с 961!!!
Keys: 806, 478

3.
A должно быть взаимно простым с 961!!!
Keys: 775, 75
```

```
21.
Keys: 86, 790
Содержит невозможную биграмму оы
22.
Keys: 466, 305
Содержит невозможную биграмму оы
23.
Keys: 646, 227
Содержит невозможную биграмму аы
24.
Keys: 850, 523
Содержит невозможную биграмму аы
25.
Keys: 875, 876
Содержит невозможную биграмму оы
26.
Keys: 896, 768
Содержит невозможную биграмму аы
27.
Keys: 649, 940
Содержит невозможную биграмму аы
28.
Keys: 654, 777
убиватьбольшененадопослетогокаконужеубилн
    нованииодинаковыхимпульсовкубийствусо
    етбытьэтовообщемеханизмнашегодоброгоу
    ыписателянетсомнениячтоэтасимпатияпоп
```

Зашифрований текст:

кеюибщаефдфмдкдкролрццисвнуншвйняэшскевдтнюдаобсюсыэихзтмдьлыохунхмьввнсдуэммндтихкеюибщыцязкзхшвно
сыотнйьщтцншуссянхщлвжвпькшвнмщзфтсхщпддкясввццтнавпьгнуьввйнлхиьерддыцрихэкьзцэижцьехщмсэкжлрибужд
эмхимьпьявсттнзцюсфспьузйпдкнхркхульацкчашяьнсибжяксэкццзтчщиюцншумщошяьщкщнфрхуюижсгцыззфрщихзтчщ
рихнэпозтгфккчщкдмклыоьеынуййлцяьэрхнмкпмдкйыпоиэуныэнсмнмсхэццьедктництндуццоэивупхюфйчсьивйэютнрц
шэбвщншуоздкдктнунянккфкяящиссбинкурдцбщшдскрщянщкдкяяищжшсвыьербщяяшндузйнкщнвнгоьцэииспьтуумщщ
шдекхндуаошдвдеигебуявюсшьйдроццвнфиибжлакццввбвываккчслтьхщзйьцжьбрьецфтспьбишыювдьезбтнмсэкжллрчсх
щьрпьшвшнйьяьнсибжлтьчсйьрьэчтнундулфтснсшбйнбжжцрнмющькккюиеуяэзтьяяреурндуьцоэгкмбобмщкскехюксдцтсы
взтмсунйьксщиссшнчщзйьцйнпршькккфкяслркейьйнавпьхсуншнузеумкжлаклцисуьдьбкфипьйнмсуншснхтуйнццмсяьмны
онкцркчыоклзфкчпьвныуозрбжлжвцнхщсссцжьбипсрзфкаьихмнщэчсавозулбутнзцнулцзткоццвнфиибхюпвиэислбиювинх
ыршьивцнярбщфджлзйьцйнзцнулцяьйнвнцхркпрыожврщьянкиюджджкеспьибубиюхщбуакикяеэдакаоццсвлбеилрлвцофкяя
ышвнунхщлвэкжлтьосцнхщиютнуншнмстспьлйаихщрнньнхшвщшвносчсабьешижсоэосыумщмбриввудябакфурщяэлчязд
каиьечслсосэкццяьцнэлязаьцнхщсссцжььзжлмщунавшьавзтьяюсуйвнакдуюиььяучмпрфдййвдихрнфззфтнхщхиеуяэзтьяы
уццыьбеелфеипвидийдкяязщпупзобчсуьвнлвмьтнчщьеэдвнстйндуаомнщоццвнфиибхюихтоццсввныклрынпьювюсисцйв
нихчщлракющчьцнхщбщщйтннсхщдкищьешичщкздукчввзтьяакккйдищжлывьктзихывуллвовявшньсйссцпрыоынчкццяьк
лхнщэюдриисэкжллреуньыктзшрэчшиязиебчлвацлотнуншнмстспьицшэмвшщкзлоябсчбщшдыцэикзясусйнюйозвьтныэак
осжцшншвюийдьяшншвосюсчязиьсунуллвихывхдсккламщубшскуаохщрнрцязакубсчфкяяосгйрщтнгбфдзйьцэибусчжвавм
нззфдыоиюшсосюдритьйьнсхщтньцмнрнннстрсосуллвзтвднкцьяубщхичщмщтсчтгнэкхуямйдчщццмнрншвйнввлвацшвьх
аврщшнищюиьсщожсюдгнуцрнчзшрынулцхдвмьцнрнуьнцяедьхсцнфуэюосйсчцэидктнуншнмншспьчшвнюдцфвдыоияосу
нйпщнбкчзиввнмнрьнсибчзлориисэибудкяспнззжлфсчсбкаышнтныьзтпэпьмвзтьсйядуцщщщцспрчсэьлвзтклбулцшвюибщ
ыцвивнуйвнакеичмывпвыэдчфкклццсвынуняуумпьшвшрцциссцмючщиюлврлиэйбдцриьцяьввюдаолыфьмодкчьяуфкойнк
йдлцыцтнавчзфдыожяшсввдуюизбывщшвныэльидыщубшврчязрщвдойвнвнмщнсунцомюхщньюссттнхщщщфддбтьпнзкь
еэдхнщьжвзтфрлцдкяяхьовюсстхщрнпьйнщофкпрынсиулйдццхифсчсхдййрснсерццисшнюсшьсцклтьпвидрошифкяяшню
даоосунчзфпьцэилцмяэьсцклжшвнунакубакюйтносшнпьявывйнщожсунюэсцэиринкгеэдвэцнпдрщрнчстнввшвпвпыызмбй
нвнцхпнуцязьсйядуулрибувдвнщозьгйбчйдсчбщиэбкдктнхщхилвннюсвнщоknирэчрниянцяеьцтсывзтосибддбпмьлривве
эяьхэфртгрулцузбщшьавтулцибсчннисозфдыожллрдцбщшдскрщиэбквэгвжвзтшвжьаоеитншпвихэхаорщибясфсчсщьавпь
скггыоющлхвииспьвиулбутнзцнулцяьжцюсчвввйимюгвшнщиющюирсунлсгоьрыноьхоццвнфиибкзенуьпьбцрныгщйеуйнз
щшьявхщеуеидебупьесузюшдкясюэсцэиыцзттнмслдроавежбщяйрщйуюйлцеищькккффдкфьнхчщмщявисчтжьамаофисрябс
чшижслбубщэнщфдэмсщябубчзйсанэирщхщмсэктзлэусхщрнляпдгсгцшфдкфьввнкубубяслоюищщшдекщсхдскхсовпннчу
бакакхуямдкяяхсвнхбжсмкщнщьжвэкссщьккдктнфифсбвддкястнтнмслдьшсвьцйьшнсиеуюкыщцспрыьлнфкйдщщзйьцйн
ыэвнхбрифкйыунрншьвнбкубьебчсвйнжндуеисхавупмююсшодкльулбусчцнннстрсшншвьхаврщянсцознкссьеуснсмнмснс
ибссвддцйнчсщнэпозцфибссщщубссвнхбрифкясхшфдцяьклрыоибсчфкщйвносэиэчпнзкццяьклакаолржцяьзтхдицфптнхщ
ыглозфьцэидктнунэибунсхщавьвлващеутнищлрдцбщшдыцйнвнцхдздкицмяьхавьщвуцфьцжьщнмкпмдкяярнэирщввпноул
цфрынщхыщмснфжврйвньркзскыщссвнхбрифкясозййцфцнюириьсосйгыовдриклакязеудкяяосузмщчяввввнищрилвацшвьи
чдрщдкикгбмщбущстссвйьшвоейулцгйщщфкнхдкбщщйвнихобсчшибщекбщэюнхзциссичщиютнмслдфишдмбццмсгцшвэ
рзфвджяжвявшнмсчярщхьовюстымщкзищссыршьудццрреулфщщаефдхссироювяьисшщкзпксчролвтнрицнмскмжяявзтси
югщхтнмспбмщбущськмюннисдкдкцфжвйьдтмщшвпвкмжяьямщшвжьрефщакиеэдакролфбклцбуябзщбукзунгэщькккгнвв
шнивжврщрныуоззнбкжлтьбцрныгйснжшдекцгеэюсрсхщньбиулбунхнчйдпнввкцйнуншвэьтнщоьцчсуьсцтгуьйнньосфипь
явпьпршьйнлхавьщсиеуобмбмщбущсфрмщчяовупмюосшнкуаохщмсэкццзтбыьмнжннуыфрыэиьсфсчсщьавозщсосгйлцм
ктзулынйнуйаихщавиэжьчщоуобмблвыьрнунокпмшрдцбщшддбубихйсансцрбжлвэкхюдрошджсюсунынмсйкмбкзхщхурсу
нщхввввмдкорыуснчзьяуиюшсвпнкурмщеувирсунсццьблшэннбвамозмщбвскаьшнжьжвупклэчйдищьешиивебпрябакоьзтя
нщиссйебчввтсзкиющьккбыоскчицпьявицчзивяьочлцсвпдгсуфдкфьяэюдаорибщвчрытнрсбидуаодункющхиьсхдгсунфрлц
дкяякадункчзжсюсбчкнбквьфзтнуноьюддкнхживналбуыодкеиочоьлхэфдкфьпьлннсвнмкхсмщтсывзтьятнакфкпрябйожсюс
унюиикцфтсввщбакксйнбжрисцвджцмнщькмыгьяьехщсяюсстхщрнхщбщыцвиклаккзеущнюсияоусчтсйьзтклларццюсстшн
юдкшвнгьерынньэынавэкиютынньюютноьакеишдщщшвпвмндтихжцшнйнюирсыэьяокпмаобщцсэщбушсхщмсэкссьейп
фкясищхнэкмбжлжвннстрсосщэтсяьяубщыцввяфжсюсунтсчтгвмьввьелвмкрюеэзтдццрнмюхщбуакдожсвнйсзвпьфихщчс
яьзтьяйкчзфсчсгэлнцнерссжофкеиябпвистнпвюскиосырынщэгожсгцмефдфмжяосзкццзтпытнрсакьлмщриарзфеуэирибщх
иьсуйвнихвнстйнянцуфкщщцсунхдицяедьакхуумжсвнчрлвньзтьяйкчзезьцюсжрыщумьцэиясезьцвнвнунищьеяцпьерынхщ
щщыцвиьянсибясшнлсиьпвтснфюирыюсцьаккнивжошижсмкарссжозщццесшндцнсккаирсыэокпмщнввйкриаршьлнуьэиу
лбунхмокздцрнфзфпдкяспнчкхуцфюижсшщязюсшсиэжьввшвяэосрнеелоюисьфиосэщублыунчяюэецхзивяьокхуямщщшдб
офдгвмсжкддьяжьяущнвввввшнмьвврщозенйсуньейпфкаьтныоеущькхзцнулцзтднчелвпьгцбуавкмлыклтьяуаишдщщмюкео
убщыцвиакэмлхчярщтсчтрьйнвнцхмьакггмщшджсунлххэхьзтлрэчбудквввзнввшнжьжврщунынжвжрццисчцэиаьмчввршищ
ссркжэжвмндтфрлцяьклхнгцязвэкьзцэиьшсвмдьцюяусиебчдуьешдриезмщюиоуриесввхьовэкжятнмслдзьлсрщйносыклрлв
рнввлэусхщрнавпьгбубсвйнавдьоспншсмкпрынкчмсхщнкойщщбщшдмефдфмжлрифсбвддкяяыоввйнщцыгевввйьмэоьж
йвнакеиэчпьидфккнйкрижэпншнхщынгспнурнрнгошддкяяфсшьыоарфдрижлццэччсавпьзншвйнрнкизфтсиспьнкгбмщбущс
сцшнмьввьщянмсхмдктнянккбщшдекцщжлывйкквпншнхщынгспныэрнгошддкйыявзтцнюфввовявлиьцяьокпмаишнмнээх
фкччтхдициввспьгсунмщпвюдцфюирыусунлрлцдкяяыуаокнввпьфзлцвнстбвхщщслэмдчзоулыфьттлозфьцэидкнхпрынкчм
стспьвифщгбрыяьцщжлзфпреурндцвныкмбарбуябакфккчявпвлсзврщьяшныййнмьунжкиюхщлвхщпэжвчспьпрццсвпддктн
дклцнулцмклытсющшдекцщзтиэярчсжвюсстибдцньтсюсстхщээрщьечщкзмщрнтслкеурьйомюхщньюсстнулбуввзнтснфчз
ццзтвииярщьякбньависйщкзхщхуиюшннуяетнхщюиакфккчлспьыопьрцмнрншбынлсюдризьяуфкшдвчсксчавзтрщхсщв

Ключи: a=654, b=777

Розшифрований текст:

убиватьбольшененадопослетогокаконужеубилноследуетемубытьблагодарныминачепришлосьбыубиватьсамомуэтонеодно
лишьдоброесострадание этоотождествлениенаоснованииодинаковыхимпульсовкубийствусобственноговорялишьвминима
льнойстепенисмещенныйнарциссизмэтическаяценностьэтойдоброты этимнеоспариваетсяможетбытьэтовообщемеханизмн
ашегодоброгоучастияпоотношениюкдругомучеловекуособенноясноп роступающийвчрезвычайномслучаеобремененногос
ознанияс воейвиныписателянетсомнениячтоэтасимпатияпопричинеотождествлениярешительноопределилавыборматериа
ладостоевскогоносначалаонизэгоистическихпобужденийвыводилобыкновенногопреступникаполитическогоирелигиозног
опреждечемкконцусвоейжизнивернутьсякпервопреступникукотцеубийцеисделатьвеголицесвоепоэтическоепризнаниеопу
бликованиеегопосмертногонаследияидневниковегоженыяркоосветилоодинэпизодегожизнитовремякогдадостоевскийвгер
маниибылобуреваемигорнойстрастьюдостоевскийзарулеткойявныйприпадокпатологическойстрастикоторыйнеподдаетсяи
нойоценкенискакойсторонынебылонедостаткавоправданияхэтогостранногоинедостойногоповеденияучувствовиныкакэтон
ередкобываетуневротиковнашлоконкретнуюзаменувобремененностидолгамиидостоевскиймоготговариватьсятемчтоонпр
ивыигрышеполучилбывозможностьвернутьсяврос сиюизбежавзаключенияв тюрьмукредитораминоэтобылтолькопредлогд
остоевскийбылдостаточнопроницателенчтобыэтопонятьидостаточночестенчтобывэтомпризнатьсяонзналчтоглавнымбыла
играсамапосебевсеподробностиегообусловленногопервичнымипозывамибезрассудногоповеденияслужаттомудоказательст
вомиещекоечемуиномуоннеуспокаивалсяпоканетерялвсегоиграбыладлянеготакжесредствомсамонаказаниянесчетноеколи
чествораздавалонмолодойженесловоиличестноесловобольшенеиграть илинеиграть вэтотденьионнарушалэтословокаконар
ассказываетпочтивсегдаеслионсвоимипроигрышамидоводилсебяиеедокрайнебедственногоположенияэтослужилодлянего
ещеоднимпатологическимудовлетворениемонмогпереднеюпоноситьиунижатьсебяпроситьеепрезиратьегораскаиватьсявто
мчтоонавышлазамужзанегостарогогрешникаипослевсейэтойразгрузкисовестинаследующийденьиграначиналасьсноваимо
лодаяженапривыклакэтомуциклутаккаказаметилачтотоотчеговдействительноститолькоиможнобылоожидатьспасенияписат
ельствоникогданепродвигалосьвпередлучшечемпослепотеривсегоизакладыванияпоследнегоимуществасвязивсегоэтогоон
аконечнонепонималакогдаегочувствовиныбылоудовлетворенонаказаниямиккоторымонсамсебяприговорилтогдаисчезалаз
атрудненностьработетогдаонпозволялсебесделатьнесколькошаговнапутикуспехурассматриваярассказболеемолодогописа
телянетрудноугадатькакиедавнопозабытыедетскиепереживаниянаходятвыявлениявигорнойстрастиустефанацвейгапосвят
ившегомеждупрочимдостоевскомуодинизсвоихочерковтримастеравсборникесмятениечувствестьновелладвадцатьчетыреч
асавжизниженщиныэтотмаленькийшедеврпоказываеткакбудтолишьтокакимбезответственнымсуществомявляетсяженщин
аинакакиеудивительныедлянееесамойзаконо нарушенияеетолкаетнеожиданноежизненноевпечатлениеноновеллаэтаеслипод
вергнутьеепсихоаналитическомутолкованиюговоритоднакобезтакойоправдывающейтенденциигораздобольшепоказывает
совсеминоеобщечеловеческоеилискореообщемужскоеитакоетолкованиестольявноподсказаночтонетвозможностиегонедоп
устить длясущностихудожественноготворчествахарактерночтописательскоторымменясвязываютдружескиеотношенияв отв
етнамоирасспросыутверждалчтоупомянутоетолкованиеемучуждоивовсеневходиловегонамерениянесмотрянаточтоврасска
звплетенынекоторыедеталикакбырассчитанныенаточтобыуказыватьнатайныйследвэтойновеллевеликосветскаяпожилаяда
маповеряетписателюотомчтоейпришлосьпережитьболеедвадцатилеттомуназадраноовдовевшаяматьдвухсыновейкоторыев
нейболеененуждалисьотказавшаясяоткакихбытонибылонадежднасороквторомгодужизнионапопадаетвовремяодногоизсво
ихбесцельныхпутешествийвигорныйзалмонакскогоказиногдесредивсехдиковинеевниманиеприковываютдверукикоторыес
потрясающейнепосредственностьюисилойотражаютвсепереживаемыенесчастнымигрокомчувстварукиэтирукикрасивого
юношиписателькакбыбезовсякогоумысладелаетегоровесникомстаршегосынанаблюдающейзаигройженщиныпотерявшего
всеивглубочайшемотчаяниипокидающегозалчтобывпаркепокончитьсосвоеюбезнадежнойжизньюнеизяснимаясимпатияза
ставляетженщинуследоватьзаюношейвпредпринятьвседляегоспасенияонпринимаетеезаоднуизмногочисленныхвтомгород
енавязчивыхженщинихочетотнееотделатьсяноонанепокидаетегоивынужденавконцеконцоввсилусложившихсяобстоятельс
твостатьсявегономереотеляиразделитьегопостельпослеэтойимпровизированнойлюбовнойночионавелитказалосьбыуспоко
ившемусяюношедатьейторжественноеобещаниечтоонникогдабольшенебудетиграть снабжаетегоденьгаминаобратныйпуть
исосвоейстороныдаетобещаниевстретитьсяснимпередуходомпоезданавокзаленозатемвнейпробуждаетсябольшаянежност
ькюношеонаготовапожертвоватьвсемчтобытолькосохранитьегодлясебяионарешаетотправитьсяснимвместевпутешествиев
местотогочтобыснимпроститьсявсяческиепомехизадерживаютееионаопаздываетнапоездвтоскепоисчезнувшемуюношеон
асноваприходитвигорныйдомисвозмущениемобнаруживаеттамтежерукинакануневозбудившиевнейтакуюгорячуюсимпати
юнарушительдолгавернулсякигреонанапоминаетемуобегообещаниинооде ржимыйстрастьюонбранитсорвавшуюегоигруве
литейубиратьсявонишвыряетденьгикоторымионахотелаеговыкупитьопозореннаяонапокидаетгородавпоследствииузнаетч
тоейнеудалосьспастиегоотсамоубийстваэтаблестящеибезпробеловвмотивировкенаписаннаяновеллаимеетконечноправона
существованиекактаковаяинеможетнепроизвестиначитателябольшоговпечатленияоднакопсихоанализучитчтоонавозникла
наосновеумопостроя емоговожделенияпериодаполовогосозреванияокаковомвожделениинекоторыевспоминаютсовершенн
осознательносогласноумопостроя емомувожделениюматьдолжнасамаввестиюношувполовуюжизньдляспасенияегоотзаслу
живающегоопасениявредаонанизмастольчастыесублимирующиехудожественныепроизведениявытекаютизтогожепервоис
точникапороконанизмазамещаетсяпорокомигорнойстрастиударениепоставленноенастрастнуюдеятельностьрукпредательс
кисвидетельствуетобэтомотводеэнергиидействительноигорнаяодержимостьявляетсяэквивалентомстаройпотребностивона
низмениоднимсловомкромесловаигранельзяназватьетееаа

Код програми

```
import AppKit

// ------------------------------- SOURCE ------------------------- //

let alphabet_enum : [ Int : String] = [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:"ь", 27:"ы", 28:"э", 29:"ю", 30:"я"]

let alphabet = ["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", "ы",
"ь", "э", "ю", "я"]

var invalphabet_dict : [ String : Int] = ["а":0, "б":0, "в":0, "г":0, "д":0,
"е":0, "ж":0, "з":0, "и":0, "й":0, "к":0, "л":0, "м":0, "н":0, "о":0, "п":0,
"р":0, "с":0, "т":0, "у":0, "ф":0, "х":0, "ц":0, "ч":0, "ш":0, "щ":0, "ы":0,
"ь":0, "э":0, "ю":0, "я":0]

var index : [ String : Double] = ["а":0, "б":0, "в":0, "г":0, "д":0, "е":0,
"ж":0, "з":0, "и":0, "й":0, "к":0, "л":0, "м":0, "н":0, "о":0, "п":0, "р":0,
"с":0, "т":0, "у":0, "ф":0, "х":0, "ц":0, "ч":0, "ш":0, "щ":0, "ы":0, "ь":0,
"э":0, "ю":0, "я":0]

let invalphabet_enum : [ String : Int] = ["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, "ь":26, "ы":27, "э":28, "ю":29, "я":30]

let invalid_bigrams : [String] = ["аъ", "оъ", "еъ", "иъ", "уъ", "оъ", "аы",
"оы", "еы", "уы", "оы"]


let top_five_original : [Int] = [545, 417, 542, 403, 168] //русский
var top_five_cypher = [Int]()
var flag = false

var bigrams_arr : [String : Int] = ["aa" : 0]
// -------------------------- TEXT EDIT ---------------------------------- //

let path = "/Users/_ria_go_/Desktop/универ/crypto3/crypto3/crypto3.txt"
var text = try String(contentsOfFile: path, encoding: String.Encoding.utf8)
text = text.lowercased()
text = text.replacingOccurrences(of: "ё", with: "е")
text = text.replacingOccurrences(of: "ъ", with: "ь")
text = text.replacingOccurrences(of: "\n", with: " ")
text = text.filter("абвгдежзийклмнопрстуфхцчшщыьэюя".contains)

print(text)
var bigram = ""
for character in text {
    invalphabet_dict[String(character)] = invalphabet_dict[String(character)]!
+ 1
    bigram = bigram + String(character)
    if bigram.count == 2 {
        if let num = bigrams_arr[bigram] {
            bigrams_arr[bigram]=num+1
        } else {
            bigrams_arr[bigram]=1
        }
        bigram.removeAll()
    }
}
```

```
let sortedfreq = bigrams_arr.sorted(by: { $0.value > $1.value })
var top = 0
var top5 = [String]()
print("Топ 5 биграмм зашифрованного текста:")
for item in sortedfreq{
    if (top<5) {
        top5.append(item.key)
        let i = invalphabet_enum[String(top5[top].first!)]
        let j = invalphabet_enum[String(top5[top].last!)]
        top_five_cypher.append(i!*31 + j!)
        //print(top_five_cypher[top])
        print(top5[top])
    }
    top+=1
}

var delta_y_arr = [Int]()
var delta_x_arr = [Int]()
for i in 0...4{
    for j in 0...4{
        if (i != j) {
            let first_y = top_five_cypher[i]
            let second_y = top_five_cypher[j]
            var delta_y = first_y - second_y
            if (delta_y < 0){ delta_y = 961 + delta_y}
            delta_y_arr.append(delta_y)

            let first_x = top_five_original[i]
            let second_x = top_five_original[j]
            var delta_x = first_x - second_x
            if (delta_x < 0){ delta_x = 961 + delta_x}
            delta_x_arr.append(delta_x)
        }
    }
}


func extevcl (a : Int, b : Int) -> (d : Int, x : Int, y : Int) {
    var q = [Int]()
    var n = [Int]()
    n.append(a)
    n.append(b)
    var xx = [Int]()
    xx.append(1)
    xx.append(0)
    var yy = [Int]()
    yy.append(0)
    yy.append(1)
    var i = 0
    while n[i+1] != 0
    {
        i+=1
        q.append(n[i-1] / n[i])
        n.append(n[i-1] % n[i])
        xx.append(xx[i-1] - xx[i]*q[i-1])
        yy.append(yy[i-1] - yy[i]*q[i-1])
    }
    //print("GCD: \(n[i]), x: \(xx[i]), y: \(yy[i])")
    let d = n[i]
    let x = xx[i]
    let y = yy[i]
    return (d, x, y)
}

func inverse (num : Int, mod : Int) -> Int{
```

```
    let result = extevcl(a: mod, b: num)
    var invnum = 0
    if (result.d == 1) {
        if (result.y<0){invnum = mod + result.y}
        else {invnum = result.y}
    }
    else  if (result.d > 1) {
        //if
         if (result.y<0){invnum = mod + result.y}
                else {invnum = result.y}
    }
    else {print("Doesn't exists")}
    //print(invnum)
    return invnum
}

func wordfinder (num : Int) -> String{
    var bigram = ""
    let x : Int = num/31
    let y : Int = num%31
    bigram = bigram + alphabet_enum[x]!
    bigram = bigram + alphabet_enum[y]!
    return bigram
}

func to_cypher (text : String, a : Int, b : Int) -> String {
    var cypher = ""
    var bigram = ""
    var cyphlet = ""
    var x : Int = 0
    let result = extevcl(a: a, b: 31)
    if (result.d == 1) {
        for character in text {
            invalphabet_dict[String(character)] =
invalphabet_dict[String(character)]! + 1
            bigram = bigram + String(character)
            if bigram.count == 2 {
                let i : Int = invalphabet_enum[String(bigram.first!)]!
                let j : Int = invalphabet_enum[String(bigram.last!)]!
                x = i*31 + j
                let temp : Int = (a * x + b)%961
                cyphlet = wordfinder(num : temp)
                cypher = cypher + cyphlet
                bigram.removeAll()
            }
        }
    }
    else{print("A должно быть взаимно простым с 961!!!")}
    return cypher
}

func de_cypher (text : String, a : Int, b : Int) -> String {
    var decypher = ""
    var bigram = ""
    var decyphlet = ""
    var y : Int = 0
    let result = extevcl(a: a, b: 31)
    if (result.d == 1) {
        for character in text {
            invalphabet_dict[String(character)] =
invalphabet_dict[String(character)]! + 1
            bigram = bigram + String(character)
            if bigram.count == 2 {
                let i : Int = invalphabet_enum[String(bigram.first!)]!
                let j : Int = invalphabet_enum[String(bigram.last!)]!
                y = i*31 + j
```

```
                var y_minus_b = y - b
                if (y_minus_b < 0){y_minus_b = 961 + y_minus_b}
                let inv : Int = inverse(num: a, mod: 961)
                let temp : Int = (inv*y_minus_b)%961
                decyphlet = wordfinder(num : temp)
                decypher = decypher + decyphlet
                for item in invalid_bigrams{
                    if (decypher.contains(item) == true){
                    decypher = "Содержит невозможную биграмму \(item)"
                    break
                    }
                }
                bigram.removeAll()
            }
        }
    }
    else{print("A должно быть взаимно простым с 961!!!")}
    return decypher
}

func keys_search (){
    var n = 0
    var m = 0
    var l = 0
    var b : Int = 0
    var a : Int = 0
    for i in 0...19{
        for j in 0...19{
            l += 1
            print("\(l).")
            let result = extevcl(a: delta_x_arr[j], b: 961)
            if (result.d == 1)
            {a = (delta_y_arr[i]*inverse(num: delta_x_arr[j], mod: 961))%961}
            if ((result.d > 1) && (delta_y_arr[i]%result.d == 0)) {
                let a1 = delta_x_arr[i]/result.d
                let b1 = delta_y_arr[j]/result.d
                let n1 = 961/result.d
                let x0 = (b1*inverse(num: a1, mod: n1))%n1
                for i in 0...(result.d)-1 {
                let ainv = x0+i*n1
                a = inverse(num: ainv, mod: 961)
                    b = (top_five_cypher[n] - a * top_five_original[m])%961
                if (b < 0){b = 961+b}
                if ((j+1)%4 == 0) {m += 1}
                let decypher_text = de_cypher(text: text, a: a, b: b)
                print("Keys: \(a), \(b)")
                print(decypher_text)
                }
            }
            else {
            b = (top_five_cypher[n] - a * top_five_original[m])%961
            if (b < 0){b = 961+b}
            if ((j+1)%4 == 0) {m += 1}
            let decypher_text = de_cypher(text: text, a: a, b: b)
            print("Keys: \(a), \(b)")
            print(decypher_text)
                if (decypher_text.count > 100){
                    flag = true
                    break
                }
            }
        }
        if flag {break}
        m = 0
        if ((i+1)%4 == 0) {n += 1}}
}
```

```
// ------------------------------ MAIN ---------------------------------- //

print("\nВыберите действие:\n1 - Зашифровать текст;\n2 - Расшифровать текст
имея ключ;\n3 - Найти НОД;\n4 - Найти обратный элемент;\n5 - Поиск a и b\n")
let answer = readLine()!
switch Int(answer) {
case 1:
    print("Введите ключ a: ")
    let a : String = readLine()!
    print("Введите ключ b: ")
    let b : String = readLine()!
    let methodStart = Date()
    print(to_cypher(text: text, a: Int(a)!, b: Int(b)!))
    let methodFinish = Date()
          let executionTime = methodFinish.timeIntervalSince(methodStart)
          print("Execution time: \(executionTime)")
case 2:
    print("Введите ключ a: ")
    let a : String = readLine()!
    print("Введите ключ b: ")
    let b : String = readLine()!
    let methodStart = Date()
    print(de_cypher(text: text, a: Int(a)!, b: Int(b)!))
    let methodFinish = Date()
          let executionTime = methodFinish.timeIntervalSince(methodStart)
          print("Execution time: \(executionTime)")
case 3:
    let methodStart = Date()
    print("Введите первое число: ")
    let first : String  = readLine()!
    print("Введите второе число: ")
    let second : String  = readLine()!
    _ = extevcl(a: Int(first)!, b: Int(second)!)
    let methodFinish = Date()
    let executionTime = methodFinish.timeIntervalSince(methodStart)
    print("Execution time: \(executionTime)")
case 4:
    print("Введите число: ")
    let num = readLine()!
    print("Введите модуль: ")
    let mod = readLine()!
    let methodStart = Date( )
    print("Обратный элемент: \(inverse(num: Int(num)!, mod: Int(mod)!))")
    let methodFinish = Date()
          let executionTime = methodFinish.timeIntervalSince(methodStart)
          print("Execution time: \(executionTime)")
case 5:
    let methodStart = Date( )
    keys_search()
    let methodFinish = Date()
          let executionTime = methodFinish.timeIntervalSince(methodStart)
          print("Execution time: \(executionTime)")
default:
    print("Упс, что-то пошло не так")
}
```

Висновок:

Отже, в ході практикума ми засвоїли принцип криптоаналізу шифру афінної біграмної підстановки, набули практичних навичок у частотному аналізі на прикладі розкриття моноалфавітної підстановки та опанували прийоми роботи в модулярній арифметиці.