Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Фізико-технічний інститут

КРИПТОГРАФІЯ

КОМП’ЮТЕРНИЙ ПРАКТИКУМ №3

Криптоаналіз афінної біграмної підстановки

Виконали:
ФБ-31 Федорович Дарина
ФБ-31 Шваюк Олександра

Мета роботи

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

Порядок виконання роботи

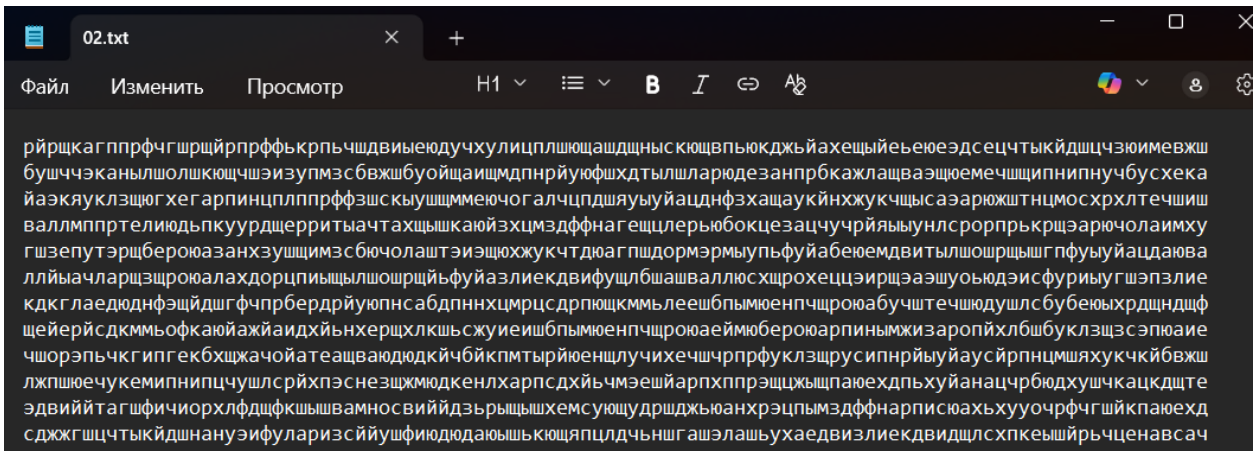
0. Уважно прочитати методичні вказівки до виконання комп’ютерного практикуму.

Нам надається текст, що є результатом шифрування за допомогою афінної підстановки біграм відкритого тексту, написаного російською мовою без пробілів, знаків пунктуації та великих літер. Буква «ё» заміщена буквою «е», а «ъ» – буквою «ь» (або навпаки). Таким чином, алфавіт відкритого тексту складається з 31 букви, що занумеровані в алфавітному порядку: а = 0 , б =1, …, я = 30 .

П’ятьма найчастішими біграмами російської мови (в порядку спадання частот) є біграми «ст», «но», «то», «на», «ен».

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.

Наш варіант – 2, ми маємо ось такий текст для розшифрування:




```python
def extended_gcd(a, b):
    if b == 0:
        return a, 1, 0
    g, x1, y1 = extended_gcd(b, a % b)
    x = y1
    y = x1 - (a // b) * y1
    return g, x, y
```

Функція extended_gcd(a, b) реалізує розширений алгоритм Евкліда для пошуку найбільшого спільного дільника (НСД) двох чисел a та b, а також знаходження коефіцієнтів, які задовольняють рівняння:

```python
def mod_inverse(a, m):
    g, x, _ = extended_gcd(a, m)
    if g != 1:
        return None
    return x % m
```

Функція solve_liner_congruence реалізує обчислення оберненого елемента за модулем.

```python
def solve_linear_congruence(a, b, m):
    g, x, _ = extended_gcd(a, m)
    if b % g != 0:
        return []

    x0 = (x * (b // g)) % m
    step = m // g

    return [(x0 + i * step) % m for i in range(g)]
```

2. За допомогою програми обчислення частот біграм, яка написана в ході виконання комп'ютерного практикуму №1, знайти 5 найчастіших біграм запропонованого шифртексту (за варіантом).

```
5 найчастіших біграм шифртексту : ['йа', 'юа', 'чш', 'юд', 'рщ']
```

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).

```
Починаємо перебір 400 можливих співставлень пар...
-> Співставлення: ('ст', 'но') -> ('йа', 'чш'). Знайдено кандидати на ключ (a, b): [(552, 232)]
-> Співставлення: ('ст', 'но') -> ('йа', 'юд'). Знайдено кандидати на ключ (a, b): [(836, 173)]
-> Співставлення: ('ст', 'но') -> ('йа', 'рщ'). Знайдено кандидати на ключ (a, b): [(854, 934)]
-> Співставлення: ('ст', 'но') -> ('юа', 'чш'). Знайдено кандидати на ключ (a, b): [(707, 945)]
-> Співставлення: ('ст', 'но') -> ('юа', 'юд'). Знайдено кандидати на ключ (a, b): [(30, 886)]
-> Співставлення: ('ст', 'но') -> ('юа', 'рщ'). Знайдено кандидати на ключ (a, b): [(48, 686)]
-> Співставлення: ('ст', 'но') -> ('чш', 'йа'). Знайдено кандидати на ключ (a, b): [(409, 784)]
-> Співставлення: ('ст', 'но') -> ('чш', 'юа'). Знайдено кандидати на ключ (a, b): [(254, 691)]
-> Співставлення: ('ст', 'но') -> ('чш', 'юд'). Знайдено кандидати на ключ (a, b): [(284, 678)]
-> Співставлення: ('ст', 'но') -> ('чш', 'рщ'). Знайдено кандидати на ключ (a, b): [(302, 478)]
-> Співставлення: ('ст', 'но') -> ('юд', 'йа'). Знайдено кандидати на ключ (a, b): [(125, 48)]
-> Співставлення: ('ст', 'но') -> ('юд', 'юа'). Знайдено кандидати на ключ (a, b): [(931, 916)]
-> Співставлення: ('ст', 'но') -> ('юд', 'чш'). Знайдено кандидати на ключ (a, b): [(677, 1)]
-> Співставлення: ('ст', 'но') -> ('юд', 'рщ'). Знайдено кандидати на ключ (a, b): [(18, 703)]
-> Співставлення: ('ст', 'но') -> ('рщ', 'йа'). Знайдено кандидати на ключ (a, b): [(107, 827)]
-> Співставлення: ('ст', 'но') -> ('рщ', 'юа'). Знайдено кандидати на ключ (a, b): [(913, 734)]
-> Співставлення: ('ст', 'но') -> ('рщ', 'чш'). Знайдено кандидати на ключ (a, b): [(659, 780)]
-> Співставлення: ('ст', 'но') -> ('рщ', 'юд'). Знайдено кандидати на ключ (a, b): [(943, 721)]
-> Співставлення: ('ст', 'то') -> ('йа', 'чш'). Знайдено кандидати на ключ (a, b): [(800, 573)]
-> Співставлення: ('ст', 'то') -> ('йа', 'юд'). Знайдено кандидати на ключ (a, b): [(557, 390)]
-> Співставлення: ('ст', 'то') -> ('йа', 'рщ'). Знайдено кандидати на ключ (a, b): [(792, 128)]
-> Співставлення: ('ст', 'то') -> ('юа', 'чш'). Знайдено кандидати на ключ (a, b): [(955, 325)]
-> Співставлення: ('ст', 'то') -> ('юа', 'юд'). Знайдено кандидати на ключ (a, b): [(712, 142)]
-> Співставлення: ('ст', 'то') -> ('юа', 'рщ'). Знайдено кандидати на ключ (a, b): [(947, 841)]
-> Співставлення: ('ст', 'то') -> ('чш', 'йа'). Знайдено кандидати на ключ (a, b): [(161, 443)]
-> Співставлення: ('ст', 'то') -> ('чш', 'юа'). Знайдено кандидати на ключ (a, b): [(6, 350)]
-> Співставлення: ('ст', 'то') -> ('чш', 'юд'). Знайдено кандидати на ключ (a, b): [(718, 554)]
```

```
-> Співставлення: ('ст', 'то') -> ('чш', 'рщ'). Знайдено кандидати на ключ (a, b): [(953, 292)]
-> Співставлення: ('ст', 'то') -> ('юд', 'йа'). Знайдено кандидати на ключ (a, b): [(404, 792)]
-> Співставлення: ('ст', 'то') -> ('юд', 'юа'). Знайдено кандидати на ключ (a, b): [(249, 699)]
-> Співставлення: ('ст', 'то') -> ('юд', 'чш'). Знайдено кандидати на ключ (a, b): [(243, 125)]
-> Співставлення: ('ст', 'то') -> ('юд', 'рщ'). Знайдено кандидати на ключ (a, b): [(235, 641)]
-> Співставлення: ('ст', 'то') -> ('рщ', 'йа'). Знайдено кандидати на ключ (a, b): [(169, 672)]
-> Співставлення: ('ст', 'то') -> ('рщ', 'юа'). Знайдено кандидати на ключ (a, b): [(14, 579)]
-> Співставлення: ('ст', 'то') -> ('рщ', 'чш'). Знайдено кандидати на ключ (a, b): [(8, 5)]
-> Співставлення: ('ст', 'то') -> ('рщ', 'юд'). Знайдено кандидати на ключ (a, b): [(726, 783)]
-> Співставлення: ('ст', 'на') -> ('йа', 'чш'). Знайдено кандидати на ключ (a, b): [(660, 954)]
-> Співставлення: ('ст', 'на') -> ('йа', 'юд'). Знайдено кандидати на ключ (a, b): [(916, 779)]
-> Співставлення: ('ст', 'на') -> ('йа', 'рщ'). Знайдено кандидати на ключ (a, b): [(269, 707)]
-> Співставлення: ('ст', 'на') -> ('юа', 'чш'). Знайдено кандидати на ключ (a, b): [(908, 954)]
-> Співставлення: ('ст', 'на') -> ('юа', 'юд'). Знайдено кандидати на ключ (a, b): [(203, 779)]
-> Співставлення: ('ст', 'на') -> ('юа', 'рщ'). Знайдено кандидати на ключ (a, b): [(517, 707)]
-> Співставлення: ('ст', 'на') -> ('чш', 'йа'). Знайдено кандидати на ключ (a, b): [(301, 62)]
-> Співставлення: ('ст', 'на') -> ('чш', 'юа'). Знайдено кандидати на ключ (a, b): [(53, 682)]
-> Співставлення: ('ст', 'на') -> ('чш', 'юд'). Знайдено кандидати на ключ (a, b): [(256, 562)]
-> Співставлення: ('ст', 'на') -> ('чш', 'рщ'). Знайдено кандидати на ключ (a, b): [(570, 490)]
-> Співставлення: ('ст', 'на') -> ('юд', 'йа'). Знайдено кандидати на ключ (a, b): [(45, 403)]
-> Співставлення: ('ст', 'на') -> ('юд', 'юа'). Знайдено кандидати на ключ (a, b): [(758, 62)]
-> Співставлення: ('ст', 'на') -> ('юд', 'чш'). Знайдено кандидати на ключ (a, b): [(705, 117)]
-> Співставлення: ('ст', 'на') -> ('юд', 'рщ'). Знайдено кандидати на ключ (a, b): [(314, 831)]
-> Співставлення: ('ст', 'на') -> ('рщ', 'йа'). Знайдено кандидати на ключ (a, b): [(692, 93)]
-> Співставлення: ('ст', 'на') -> ('рщ', 'юа'). Знайдено кандидати на ключ (a, b): [(444, 713)]
-> Співставлення: ('ст', 'на') -> ('рщ', 'чш'). Знайдено кандидати на ключ (a, b): [(391, 768)]
-> Співставлення: ('ст', 'на') -> ('рщ', 'юд'). Знайдено кандидати на ключ (a, b): [(647, 593)]
-> Співставлення: ('ст', 'ен') -> ('йа', 'чш'). Знайдено кандидати на ключ (a, b): [(919, 105)]
```

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

Побудова автомату розпізнавання російської мови:

- перевірку частот частих літер («о», «а», «е», частоти можуть розглядатись окремо або в сукупності);
- перевірку частот рідкісних літер («ф», «щ», «ь», частоти також можуть розглядатись окремо або в сукупності);
- перевірку частот біграм, підраховану для біграм «на перетині» (у вищенаведених позначеннях – біграм виду
- перевірку частот триграм та довільних l-грам.

```python
def is_meaningful(text):
    if not text:
        return False
    total = len(text)
    common_letters = 'оеаинтсрвлкмдпуя'
    rare_letters = 'фщъэё'
    freq_common = sum(text.count(ch) for ch in common_letters) / total
    freq_rare = sum(text.count(ch) for ch in rare_letters) / total
    if total < 200:
        return freq_common > 0.48 and freq_rare < 0.06
    return freq_common > 0.50 and freq_rare < 0.03
```

Ця функція is_meaningful перевіряє, чи є текст змістовним на основі частоти використання певних літер.

Результат виконання скрипта:

```
ЗНАЙДЕНО ЗМІСТОВНИЙ ТЕКСТ
Ключ (a, b) = (27, 211)
Знайдено після 132 унікальних перевірок.

Початок дешифрованого тексту: однакоэтакартинаскак
```

Шифрований текст

```
рйрщкагппрфчгшрщйрпрффькрпьчшдвиыеюдучхулицплшющашдщныскющвпьюкджьйахещыйеьеюеэдсецчтыкйдшцчзюимевжш
бушччэканылшолшкющчшэизупмзсбвжшбуойщаищмдпнрйуюфшхдтылшларюдезанпрбкажлащваэщюемечшщипнипнучбусхека
йаэкяуклзщюгхегарпинцплппрффзшскыушщммеючогалчцпдшяуыуйацднфзхащаукйнхжукчщысаэарюжштнцмосхрхлтечшиш
валлмпптелиюдьпкуурдщерритыачтахщышкаюйзхцмздффнагещклерьюбокцезаччучрйяыыунлсрорпрькрщэарючолаимху
гшзепутэрщберояазанхзушщимзсбючолаштэиэщюххжукчтдюагпшдормэрмыупьфуйабеюемдвитылшошрщышгпфуыуйацдаюва
ллйыачларщзщроюалахдорцпиыщылшошрщйьфуйазлиекдвифущлбшашваллюсхщрохеццэирщэаэшуоыюдэисфуриыугшэпзлие
кдкглаедюднфэщйдшгфчпрбердрйуюпнсабдпннхцмрцсдрпющкммьлеешбпымюенпчщроюабучштечшюдушлсбубеюыхрдщндщф
щейерйсдкммьофкаюйажйаидхйьнхерщхлкшьсжуиеишбпымюенпчщроюаеймюберояюарпинымжизаропйхлбшбуклзщзсэпюаие
чшорэпьчкгипгекбхщжачойатеащваюдюдкйчбйкпмтырйюенщлучихечшчрпрфуклзщрусипнрйыуйаусйрпнцмшяхукчкйбвжш
лжпшюечукемипнипцчушлсрйхпэснезщжмюдкенлхарпсдхйьчмэешйарпхппрэщцжыщпаюехдпьхуйанацчрбюдхушчкацкдщте
эдвиййтагшфичиорхлфдщфкшышвамносвиййдзърыщышхемсующудршджьюанхрэцпымздффнарписюахьхууочрфчгшйкпаюехд
```

Дешифрований текст

```
Дешифрований текст збережено у 'decrypted.txt'
```

```
однакоэтакартинаскакойбысторонымыеенирассматривалирасплываетсявнечтонеопределенноеприпадкипроявляющиесяр
езкосприкусываниемусиливающиесядоопасногодляжизниприводящегоктяжкомусамокалечениюмогутвсежевнекоторыхслу
чаяхнедостигатьтакойсилыослабляясьдократкихсостоянийабсансадобыстропроходящихголовокруженийимогуттакжесм
енятьсякраткимипериодамикогдабольнойсовершаетчуждыеегоприродепоступкикакбынаходясьвовластибессознательно
гообуславливаясьвобщемкакбыстранноэтониказалосьчистотелесныминричинамиэтисостоянияхмогутпервоначальновозн
икатьпоричинамчистодушевнымиспугилимогутвдальнейшемнаходитьсявзависимостиотдушевныхволненийкакнихаракте
рнодляогромногобольшинствослучаевинтеллектуальноеснижениеноизвестенпокрайнеймереодинслучайкогдаэтотнедуг
ненарушилвысшейинтеллектуальнойдеятельностигельмгольцдругиеслучаивотношениикоторыхутверждалосьтожесамоен
енадежныилиподлежатсомнениюкакислучайсамогодостоевскоголицастрадающиеэпилепсиейможутпроизводитьвпечатлен
иетупостинедоразвитоститкакэтаболезньчастосопряженасярковыраженнымидиотизмомикрупнейшимимозговымидефек
таминеявляяськонечнообязательнойсоставнойчастьюкартиныболезниноэтиприпадкисовсемисвоимивидоизмененияхмибы
ваютиудругихлицулицсполнымдушевнымразвитиемискореесосверхобычнаявбольшинствеслучаевнедостаточноуправляем
ойимиаффективностьюнеудивительночтопритакихобстоятельствахневозможноустановитьсовокупностьклиническоюафф
```

Висновок:

Ця лабораторна робота успішно підтвердила методологічну вразливість афінного шифру біграм — одного з найбільш стійких поліграфічних шифрів класичної криптографії. Ключем до успіху стало ефективне інтегрування двох потужних криптоаналітичних інструментів: детального частотного аналізу біграм шифртексту та точного алгебраїчного розв'язку системи лінійних порівнянь. Шляхом ідентифікації найімовірніших пар співставлень та застосування розширеного алгоритму Евкліда, ми змогли детерміновано обчислити всі потенційні ключі (a, b). Фінальна ідентифікація коректного ключа була здійснена завдяки реалізації автоматичного евристичного розпізнавача змістовності тексту, який надійно відфільтрував шум і підтвердив єдине правильне дешифрування.