НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

КРИПТОГРАФІЯ

КОМП’ЮТЕРНИЙ ПРАКТИКУМ №1

Криптоаналіз шифру Віженера

Варіант 9

Виконали:
студенти групи ФБ-33
Тимощенко Олександр
Назаренко Іван

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.

Для шифрування обрали текст "Братья карамазовы" довжиною 2 кб:

```
НачинаяжизнеописаниегероямоегоАлексеяФедоровичаКарамазованахожусь
внекоторомнедоумении Аименнохотяяиназываю АлексеяФедоровичамоим
героемнооднакосамзнаючточеловеконотнюдьневеликийапосемуи
предвижунеизбежныевопросывродетаковыхчемжезамечателенвашАлексей
ФедоровиччтовывыбралиегосвоимгероемЧтосделалонтакогоКомуичем
известенПочемуячитательдолжентратитьвремянаизучениефактовегожизни
ПоследнийвопроссамыйроковойибонанегомогулишьответитьМожет
бытьувидитесамиизроманаНуакольпрочтутроманинеувидятнесогласятсяс
примечательностьюмоегоАлексеяФедоровичаГоворютакпотомучтос
прискорбиемэтопредвижуДляменяонпримечателеннорешительносомневаюс
успеюлиэтодоказатьчитателюДеловтомчтоэтопожалуйидеятельнодеятель
```

ключі довжиною 2-5 символів:

```
["од", "три", "чоти", "пьять",
```

ключі довжиною 10-20 символів:

```
"рлфвтмеоиз","ущтмаржсшцв","гвдйжкьфснрв","йюцбуьктеинмп","гсшчзяхфжвдалп","пйщььевюсззъуск","жрчогымуьщлмкхкю"
```

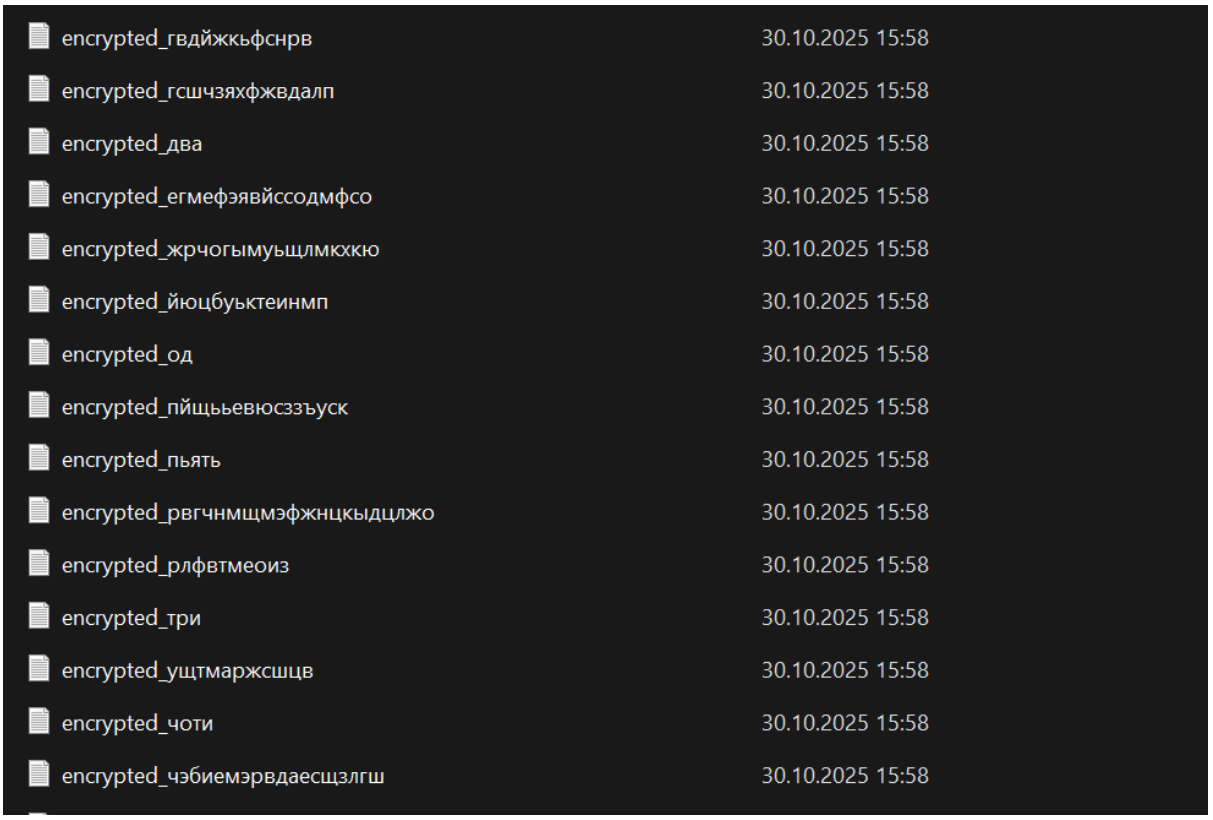
```
"егмефэявйссодмфсо","чэбиемэрвдаесщзлгш","щдеквегмвкдтьрьоьст","рвгчнмщмэфжнцкыдцлжо"
```

Функція шифрування:

```python
def vigenere(text, key):
    alphabet_upper = 'АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ'
    alphabet_lower = alphabet_upper.lower()
    encrypted = []
    key_index = 0

    for char in text:
        if char in alphabet_upper:
            shift = alphabet_upper.index(key[key_index % len(key)].upper())
            encrypted_char = alphabet_upper[(alphabet_upper.index(char) + shift) % len(alphabet_upper)]
            encrypted.append(encrypted_char)
            key_index += 1
        elif char in alphabet_lower:
            shift = alphabet_upper.index(key[key_index % len(key)].upper())
            encrypted_char = alphabet_lower[(alphabet_lower.index(char) + shift) % len(alphabet_lower)]
            encrypted.append(encrypted_char)
            key_index += 1
        else:
            encrypted.append(char)
    return ''.join(encrypted)
```

Заштфрований текст різними ключами:



Приклад зашифрованого тексту:

```
РвысукыъщфэзссмъжчдщфтарвотойшЬяцчбзвЦйнфъкцщдрМгтдхжскцсърчсичъв
мйщыыврууррцлокзэтэклВмхлчйвжывбвксйнеюфпНызнуйиЪпавбыткъврчоц
ящбыхорртнукжввньйрввашшущьытзнрсчшчъшнъхдинмуоуьгяюхоцк
ущлоюьчаэзлйеомччщуыятсуялцшащгнъреэщалцвщшньзъвцоспйцсеРнимхоп
ЮбшяэюдлщыыфмчцмоавокймфыювщщузурйхЭькехтыворсыжфкчяЧюоцкыот
тгццювзрСталцпуихввхзпекшзъцъвтгфмывммщэмэвлйчалчдщенъфсдймфрдыюх
ЯрфнйнутецяьарфудхбумвыытрмкечукйщфыьржхпсюжкжутвкхюРчмпо
хмямхекисшпнфэхшйуррйукйЗсчюнясфчэьпжбыьврксощмдшряэзфрзфжыыжвмб
сукроэкощьйэрффазтшбчяНызнуйиЪпавбыткъвЗчишмтгнъссфтхщбовв
ьакфмтщзтбаояюсузилорпШьмьзрбтцхъдацдрфинйцушмщйхвзоюсччшибцпраф
ххшлизьояюжсмдржьшлщярфинвНлхкцгыьщхрбыфщкъсшгллжйишпзрюыфзвфйфв
```

2. Підрахувати індекси відповідності для відкритого тексту та всіх одержаних шифртекстів і порівняти їх значення.
Формула підрахунку:

$$I(Y) = \frac{1}{n(n-1)} \sum_{t \in Z_m} N_t(Y)(N_t(Y)-1)$$

Функція підрахунку:

```python
def icdef(text):
    alphabet = 'АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ'
    text = ''.join([ch.upper() for ch in text if ch.upper() in alphabet])
    n = len(text)
    if n < 2:
        return 0.0
    freq = Counter(text)
    numerator = sum(count * (count - 1) for count in freq.values())
    return numerator / (n * (n - 1))
```

Індекси відповідності:

```
Відкритий текст - Індекс відповідності: 0.05976
Ключ: 'од' - Індекс відповідності: 0.04298
Ключ: 'три' - Індекс відповідності: 0.03873
Ключ: 'чоти' - Індекс відповідності: 0.03941
Ключ: 'пьять' - Індекс відповідності: 0.03812
Ключ: 'рлфвтмеоиз' - Індекс відповідності: 0.03448
Ключ: 'ущтмаржсшцв' - Індекс відповідності: 0.03343
Ключ: 'гвдйжкьфснрв' - Індекс відповідності: 0.03324
Ключ: 'йюцбуьктеинмп' - Індекс відповідності: 0.03182
Ключ: 'гсшчзяхфжвдалп' - Індекс відповідності: 0.03298
Ключ: 'пйщььевюсззъуск' - Індекс відповідності: 0.03231
Ключ: 'жрчогымуыщлмкхкю' - Індекс відповідності: 0.03256
Ключ: 'егмефэявйссодмфсо' - Індекс відповідності: 0.03417
Ключ: 'чэбиемэрвдаесщзлгш' - Індекс відповідності: 0.03154
Ключ: 'шдеквегмвкдтьрьоьст' - Індекс відповідності: 0.03206
Ключ: 'рвгчнмщмэфжнцкыдцлжо' - Індекс відповідності: 0.03191
```

Видно, що чим більший ключ - тим менший індекс відповідності

3. Використовуючи наведені теоретичні відомості, розшифрувати наданий шифртекст (згідно свого номеру варіанта).

Розшифрований текст:

```
==========
овжина ключа: 17 | Ключ: войнамагаэндшпиль
очаток тексту:
утьстарогозамканакрасноискалеплывущейнадневедомойбезднойможетпоказатьсявечныминеизменнымнаднимполыхаютпричудливыесозвездияветервыводитзамысловатыеруладыназубцахего
тенибашеннекогданатомчтопослужилооснованиемкрепостинаходилиприютсамыеудивительныесозданиядотехпорпоканеобъявилисьнастоящиехозяеваонииименовалисебяновымибогамиодиниз
ихвозвелнакрасноискалесвойзамоктвердынюкрасноискалебылосовершеннобезразличнокакихзовутэтихнезваныхгостейотчеготосразувозомнившихсебяхозяевамионаплылаиплыласебекодн
йейведомойцелииникогданиразукурсеенеизменялсямалоктовиделсходствоскалыиповявившегосянанемзамкасбрандеемтакимжелетучимостровомслугхаосаихкрепостиуничтоженнойратямихе
инаиракотатоткогозвалихединомвиделвтотвечеркогданазваныебратьябогипокинулитайнуютвердынюхединавзамкевоцариласьтугаязвенящаятишинаниктоневиделкакнапочтительномрасст
янииотстенбашенибастионовкрепостиввоздухеизничегосоткаласьчеловеческаяфигураповиселакакоетовремяазатемтакжебеззвучнорастаялазамокпустовалиниктопомнениюхединанезнал
удадорогиниедин
```

Можна зробити висновок, що це відривок з «Хранитель Мечей. Война мага. Том 3. Эндшпиль»

Код:

```
from collections import Counter

ALPHABET = "абвгдежзийклмнопрстуфхцчшщъыьэюя"
M = len(ALPHABET)
INDEX = {ch: i for i, ch in enumerate(ALPHABET)}

FREQ = {

'о':0.1097,'е':0.0845,'а':0.0801,'и':0.0735,'н':0.0670,'т':0.0626,'с':0
.0547,
```

```
'р':0.0473,'в':0.0454,'л':0.0434,'к':0.0349,'м':0.0321,'д':0.0281,'п':0
.0281,

'у':0.0262,'я':0.0201,'ы':0.0190,'ь':0.0174,'г':0.0169,'з':0.0165,'б':0
.0159,

'ч':0.0144,'й':0.0121,'х':0.0097,'ж':0.0094,'ш':0.0073,'ю':0.0064,'ц':0
.0048,
    'щ':0.0036,'э':0.0032,'ъ':0.0004
}

def clean(t):
    t = t.lower().replace("ё","е")
    r = ""
    for c in t:
        if c in INDEX:
            r += c
    return r

def ioc(s):
    n = len(s)
    if n < 2:
        return 0
    f = Counter(s)
    top = 0
    for x in f.values():
        top += x*(x-1)
    return top/(n*(n-1))

def avg_ioc(text, L):
    parts = []
    for i in range(L):
        col = text[i::L]
        parts.append(ioc(col))
    return sum(parts)/len(parts)

def period(col):
    bestperiod = 0
    bestp = 1e9
    n = len(col)
    for s in range(M):
        period = 0
```

```
        f = Counter(col)
        for ch in ALPHABET:
            obs = f[ch]
            plain = ALPHABET[(INDEX[ch]-s)%M]
            exp = FREQ.get(plain,0)*n
            if exp>0:
                period += (obs-exp)**2/exp
        if period<bestp:
            bestp = period
            bestperiod = s
    return bestperiod

def decrypt(text, L):
    shifts = []
    cols = []
    for i in range(L):
        col = text[i::L]
        s = period(col)
        shifts.append(s)
        d = ""
        for ch in col:
            d += ALPHABET[(INDEX[ch]-s)%M]
        cols.append(d)
    plain = ""
    for i in range(len(text)):
        plain += cols[i%L][i//L]
    key = "".join(ALPHABET[s] for s in shifts)
    return key, plain

filename = "шт.txt"

with open(filename,"r",encoding="utf-8") as f:
    text = f.read()

text = clean(text)

ioc_scores = []
for L in range(2,33):
    score = avg_ioc(text, L)
    ioc_scores.append((L, score))

ioc_scores.sort(key=lambda x: x[1], reverse=True)
```

```
top_L = [x[0] for x in ioc_scores[:1]]

for L in top_L:
    key, decoded = decrypt(text, L)
    print("="*50)
    print(f"Довжина ключа: {L} | Ключ: {key}")
    print("Початок тексту:")
    print(decoded[:1000])
```

Висновки:
Під час виконання роботи ми дослідили методи частотного криптоаналізу, зокрема застосування статистичних характеристик мови для зламу шифру Віженера.
Ми розглянули принцип роботи шифру Віженера, а також реалізували процес шифрування та дешифрування текстів із різними довжинами ключів.
Під час аналізу шифртекстів обчислювалися індекси відповідності, що дозволило експериментально визначити період ключа завдяки якому ми розшифрували текст.
Отже, ми впевнились в тому, що шифр Віженера зберігає статистичні властивості мови, а його криптоаналіз можливий шляхом використання частотних методів.