КОМП'ЮТЕРНИЙ ПРАКТИКУМ №3

Криптоаналіз афінної біграмної підстановки

Виконали:

студентки групи ФБ-23

Сівашенко Анна,

Тарасенко Ангеліна

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 №1, 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.

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

Хід роботи

Для дешифрування тексту за допомогою афінного шифру необхідно визначити ключі a та b. Для цього було написано наступні функції з математичними операцями:

Функція extended_gcd реалізує розширений алгоритм Евкліда. Її завдання — знайти найбільший спільний дільник (НСД) двох чисел a і b, а також обчислити коефіцієнти x і y, які задовольняють рівняння:

$$a \times x + b \times y = \gcd(a, b)$$

Ось як це реалізовано в коді:

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

```
        return g, x, y
```

Якщо a=0, то функція повертає b як НСД, а також коефіцієнти (x,y)=(0,1).

Якщо a≠0, то функція викликає саму себе рекурсивно з аргументами b % a та a. Це зменшує розмір вхідних чисел при кожному кроці, доки не досягне базового випадку.

Після повернення з рекурсії (коли НСД вже знайдено), функція обчислює коефіцієнти x та y для початкових значень a і b, використовуючи коефіцієнти з наступного рівняння в рекурсії.

За знаходження оберненого елемента відповідає функція mod_inverse.

Оборотний елемент a^{-1} визначається, як таке число, що задовольняє рівняння

$$a \times a^{-1} \equiv 1 \; mod \; m$$

Ось як це реалізовано в коді:

```
def mod_inverse(a, m):
    g, x, _ = extended_gcd(a, m)
    if g != 1:
        raise ValueError(f"Обернений елемент не існує, оскільки НСД({a}, {m}) ≠ 1")
    else:
        return x % m
```

Функція solve_linear_congruence розв’язує лінійне конгруентне рівняння виду:

$$a \times x \equiv b \; mod \; m$$

При розв’язанні конруентних рівнянь важливо коректно обробляти випадок із декількома розв’язками, повертаючи їх усі. Тому функція обчислює і повертає всі нсд розв'язків у списку solutions, забезпечуючи, що кожен можливий розв'язок враховано. Якщо g = 1, це означає, що a і m взаємно прості, тому рівняння має єдиний. Якщо g>1, рівняння має розв'язки тільки у випадку, коли b кратне g (тобто b mod g = 0). Якщо це не так, повертається False, оскільки розв'язків немає.

```
def solve_linear_congruence(a, b, m):
    g, x, _ = extended_gcd(a, m)
    if g == 1:
        x0 = (mod_inverse(a, m) * b) % m
        return [x0]
    elif b % g == 0:
        x0 = (x * (b // g)) % m
        solutions = [(x0 + i * (m // g)) % m for i in range(g)]
        return solutions
    else:
        return False
```

Оскільки ми працюємо з біграмами, що не перетинаються, то потрібно занумерувати букви алфавіту числами від 0 до m–1, тоді кожній біграмі (x_{2i-1} і x_{2i}) – можна єдиним чином співставити число Xi у границях від 0 до m^2 – 1. В результаті ми отримаємо пари чисел з їх індексами в алфавіті (а = 0, а я = 31).

```
def bigram_to_number(bigram, m):
    alphabet = 'абвгдежзийклмнопрстуфхцчшщъыьэюя'
    if len(bigram) != 2:
        raise ValueError("Біграма має містити два символи.")
    return alphabet.index(bigram[0]) * m + alphabet.index(bigram[1])
```

Для отримання можливих значень ключів використаємо частотний аналіз біграм зашифрованого та відкритого тексту. Нам відомо, що найчастіші біграми в російській мові ['ст', 'но', 'то', 'на', 'ен'].

Знайдемо 5 найчастіших біграм нашого тексту:

В результаті отримали такі біграми: ['йа', 'юа', 'чш', 'юд', 'рщ']. Можна припустити, що вони відповідають 5 найчастішим біграмам російької мови. Далі підставимо їх в систему:

$$Y^* \equiv aX^* + b \ mod \ m^2$$

$$Y^{**} \equiv aX^{**} + b \ mod \ m^2$$

У коді є два списки частотних біграм: freq_bigrams_russian — часті біграми в російській мові, freq_bigrams_cipher — часті біграми в зашифрованому тексті. Код перебирає усі комбінації біграм з обох списків. Для кожної пари біграм із відкритого тексту (bigram_rus1, bigram_rus2) і кожної пари біграм із зашифрованого тексту (bigram_cipher1, тривіальних випадків).

Функція solve_linear_congruence шукає можливі значення для першої частини ключа a за допомогою розв'язання лінійного конгруентного рівняння:

$$Y^* - \ Y^{**} \equiv a(X^* - X^{**}) \ mod \ m^2$$

Обчислюється друга частина ключа b як:

$$b = (Y^* - \ aX^*) mod \ m^2$$

Якщо комбінація [a, b] ще не додана в possible_keys, вона додається до списку.

Оскільки ми працюємо з біграмами, то значення ключів беруться за модулем m^2.

```
def find_potential_keys(freq_bigrams_russian, freq_bigrams_cipher, m):
    m_squared = m**2
    possible_keys = []
    alphabet = 'абвгдежзийклмнопрстуфхцчшщъыьэюя'

    for bigram_rus1 in freq_bigrams_russian:
        for bigram_rus2 in freq_bigrams_russian:
            if bigram_rus1 == bigram_rus2:
                continue

            for bigram_cipher1 in freq_bigrams_cipher:
                for bigram_cipher2 in freq_bigrams_cipher:
                    if bigram_cipher1 == bigram_cipher2:
                        continue

                    X1 = bigram_to_number(bigram_rus1, m)
                    X2 = bigram_to_number(bigram_rus2, m)
```

```
                    Y1 = bigram_to_number(bigram_cipher1, m)
                    Y2 = bigram_to_number(bigram_cipher2, m)

                    key_first_part = solve_linear_congruence(X1 - X2, Y1 - Y2,
m_squared)
                    if key_first_part != False:
                        for elem in key_first_part:
                            if math.gcd(elem, len(alphabet)) == 1:
                                key_second_part = (Y1 - elem * X1) % m_squared
                                if [elem, key_second_part] not in possible_keys:
                                    possible_keys.append([elem, key_second_part])
                                    print(f"Знайдені ключі: a={elem},
b={key_second_part}")

    return possible_keys
```

Після отримання всіх можливих пар ключів можна починати розшифровувати текст.

Код розшифровує ШТ за допомогою алгориму, що використовує зворотне перетворення біграм, застосовуючи `mod_inverse` щоб отримати початковий ВТ.

Цикл перебирає шифртекст по два символи за раз. Обчислює обернену величину a за модулем m_squared. Вона потрібна для зворотного перетворення біграм - відновлюється числове представлення біграми у ВТ.

x2 = X % m — обчислює позицію другої літери біграми у відкритому тексті.

x1 = (X - x2) // m — обчислює позицію першої літери біграми у відкритому тексті.

alphabet[x1] та alphabet[x2] знаходять відповідні літери для біграми у відкритому тексті.

```
def decrypt_text(ciphertext, a, b, m):
    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', 'ь',
'ы', 'э', 'ю', 'я']
    m_squared = m ** 2
    decrypted_text = ""
    for i in range(0, len(ciphertext) - 2, 2):
        Y = alphabet.index(ciphertext[i]) * m + alphabet.index(ciphertext[i +
1])
        a1 = mod_inverse(a, m ** 2)
        X = (a1 * (Y - b)) % (m ** 2)
        x2 = X % m
        x1 = (X - x2) // m
        decrypted_text = decrypted_text + alphabet[x1] + alphabet[x2]
    return decrypted_text
```

Значень ключів було отримано дуже багато – 118. Важко вручну перевірити коректність розшифрування кожного тексту, тож була написана функція, що перевіряє текст на змістовність і знаходить значення ключів, що дають змістовний текст.

Ми вирішили використовувати значення ентропії ВТ та ШТ для виявлення змістовного тексту. Так, в лабораторному практикуму №1 ми обраховували ентропію і дізнались, що для тексту без пробілів з кроком 1 значення ентропії дорівнює – 4,459.

Функція decrypt_and_filter_by_entropy обраховує ентропію для кожного ШТ і якщо його значення знаходиться в межах від 4,40 до 4,50 ми припускали, що ця пара ключів правильна.

```
def decrypt_and_filter_by_entropy(ciphertext, possible_keys, m):
    valid_keys_with_entropy = {}

    for a, b in possible_keys:
        try:
            decrypted_text = decrypt_text(ciphertext, a, b, m)
            entropy = calculate_entropy(decrypted_text)
            if entropy >= 4.40 and entropy <= 4.50:
                key_str = f"a={a}, b={b}"
                valid_keys_with_entropy[key_str] = entropy
                print(f"Ключ: {key_str}, Ентропія: {entropy}")
                return decrypt_text(ciphertext, a, b, 31)
        except ValueError as e:
            continue
    return valid_keys_with_entropy
```

В результаті виконанання програми отримані ключі: a=27, b=211.

ШТ:

рйрщкагпprфчгшрщйрпрффькрпьчшдвиыеюдучхулицплшющашдщныскющвпьюкджьйахе
щыйеьеюоэдсецчтыкйдшцчзюимевжшбушччэканылшолшкющчшэизупмзсбвжшбуойщаищ
мдпнрйуюфшхдтылшларюдезанпрбкажлащваэщюемечшщипнипнучбусхекайаэкяуклзщюг
хегарпинцплппprффзшскыушщммеючогалчцпдшяуыуйацднфзхащаукйнхжукчщысаэарюж
штнцмосхрхлтечшишваллмппртелиюдьпкуурдщерритыачтахщышкаюйзхцмздффнагещцле
рьюбокцезацчучрйяыыунлсрорпрькрщэарючолаимхугшзепутэрщбероюазанхзушщимзсбю
чолаштэиэщюхжукчтдюагпшдормэрмыупьфуйабеюемдвитылшошрщышгпфуыуйацдаювал
лйыачларщзщроюалахдорцпиыщылшошрщйьфуйазлиекдвифущлбшашваллюсхщрохеццэи
рщэаэшуоьюдэисфуриыугшэпзлиекдкглаедюднфэщйдшгфчпрбердрйуюпнсабдпннхцмрцсд
рпющкммьлеешбпымюенпчщроюабучштечшюдушлсбубеюыхрдщндщфщейерйсдкммьофк
аюйажйаидхйьнхерщхлкшьсжуиеишбпымюенпчщроюаеймюбероюарпинымжизаропйхлб
шбуклзщзсэпюаиечшорэпьчкгипгекбхщжачойатеащваюдюдкйчбйкпмтырйюенщлучихечш
чрпрфуклзщрусипнрйыуйаусйрпнцмшяхукчкйбвжшлжпшюечукемипнипцчушлсрйхпэснез
щжмюдкенлхарпсдхйьчмэешйарпхппрэщцжыщпаюехдпьхуйанацчрбюдхушчкацкдщтеэдви
ййтагшфичиорхлфдщфкшышвамносвиййдзырыщышхемсующудршджьюанхрэцпымздффн
арписюахьхууочрфчгшйкпаюехдсджжгшцчтыкйдшнануэифуларизсййушфиюдюдаюышьк
ющяпцлдчьншгашэлашьухаедвизлиекдвидщлсхпкеышйрьчценавсачэаькудбюяхцмрцсдрпг
екммьлекдхйыуыщйаудюлцчисуюэиффриещжзьргшкдыууоьдглэшешбероюачпщылшыщд
шэасуйаьпымкуюсщгхелафитбюазуыщюаешуоналаолфдыууозмсдщьбукаощжзьрыщаыпмя
ызшхпбьйацчзюимпелумсрйюасавдыугшбрмэтдйкяуришпчиоскчтхэейыосййричикзддрята
рщроюазахачшфщчшурпрбуашькщепщчшфитдьчфщроюазацквснхтбьечшчыачешудкгхавк
лаяхбмхашнэпосюеюазнтдщьбудшщепщчшфикайаэкишныцмбээлучылшрщашошзсбужиф
чмэйкблкмоснфэщкылшрщхлиечшритэзалаеймюбероюарптылшцюцрчийщпаюеющчшхпэ
щхеишашйамущьбукаьэзхцмустдмшыщдшщчсдхйыуыщйаудчикабпсаюезлиекдффыршдчи
мшлчлэфуюазздрятачшсающчшййнцусюаьжхезнмшйщгпридщнйымюдкебдкйющешхщн
кшлнуюсэебдьебпщьюарпжиегтдлэфщюенщдезаламдосусжулапасйюдаюнежсщьйкэытэшс
осгпэпщепщчшфихехщюедшэпеемучщройкэысарепуосхасасйленкссвссеоамдосвпхрзшмей
рцлтедчусхеццкемчььсдмэшсрморушнллирмффаыпмяызшщфзсййымзсхажалафщнпбупюо
ьюдкеещхщшпщяавцквснхтбьечшджпшюешпщьбуказаэплахщдщнйдщтечшджпшюешпщь

буэщшчсщряаюэщкацкышщехеаитбюарщлсцпэсеегпосщерпусдюйаюдбучихеэдэппртехарп
еылегшмчхухаяютечшюдуссайщсллдыууокайасазаопчичпнхбморешэшсающуонафщгшмей
ррихушкдщнйдщтечшщукайаэкышхемчтэхевателуцчисхпкучызшцшмейряжпшюешпщьбуд
шоылшищгамуыщюаешлуьппрринхдщцадуришпчичифубелшмшмвкйуыгшхлвпьюзсййуш
фиюдпелучырйнхюайажлэщцжйацчушугрйхпцчсдьчфщроюаепжьюдмшеемучщроюазацча
ябуащыщдшварчмэчинкныцмйквыдщлагчмэашзщэиьчщщчшмейртвещжзьргшкдтваыпмяы
зшыыдщнпщьбукачэрщмечшлжйазакмхйтвдебукчкйбвжшоыачлаоыьчмбюдпаюехдхввамнх
укчкйбвжшгсйасандуссагшяснежсчикммьлезлиекдбюфшхдиырйгекбюдтдфчнцюдавлэкдус
осйасадуклзщюдфчнцюдкемсуювпьюцкдщтечшэиащваейнцусюазблэчшгечофщгесаьпюач
пжжпшюечуаюгарпсенуказаэпюазшлууросйасажлешзлйаудрйхрмэцпфжйахеродюыщжрпр
оппрчикммьлевлщднхбмнхшсзмгьхпэсрежаолфдыууофнрйнцусюазблэчшрщзщжацчтыкйк
аешхакмхйтвжшусййушфиюдюдаюгпшгцчтыкйкающамджйазаддхухегарпцпбьюахщэдкгщ
ыфутдаюащышэылшищяросчшмезахехщяпвсхйюдаюыущаидвцюдаюьичбзлцчтыкйэщышт
ыаччбзстдаюышхехаедюшзщрпщысагшлайеошцкнуфносачзюидцецчхйхажатечшжьйацчт
ыкйдшрщзщашчоыйыуйаусйрпнюлтевйвпрпгечпщачшкдььрмегфчпрбелшцающашчопаюе
бушщькышзшвыйафщышхпцмдрщыыуюехакчщуиезафнщыаччбзстдаюрщлаеебдкйлщйачн
рйюблэчшшхнфрпющэплщцчсдфмчзьчжлаыпмяызшжхбмнхшсбужичлщерпюабуашькщыд
щвйрмыулпбьйашдтыцмюарпхвцчьрдщгшашчоламчэичаэхшстдаюриэщйазнзсзшйшлшюаг
пчиеысагшлайезщайхлбшглэщйщчшчамеешвдбювсрэжичбзлэпрешхнфрплацсрчцпхюшрф
чсимэоскгфуыйыхффэплщгарпсенуказарчыупмхуэсдммэтдяавдчишхтаичшзыйыуйаусйрпн
ушхакмюбпмншжлэщйщчшэирщлэгерпюабуосйещеэдсечушгцмпнщьбукаюдуыдщимюдке
чушгмщрщашщппрэщкырйдщьлщеющвпьюриюдюашдйржахетсййвпэсгпчинаькгшхпннз
щццтвкчисжлзсйепртшййыуйаусйрпншдажйазмгьусффщлщрбезахемчтэлекмаюрщудеапам
досшсцпфжнлзуыщюазреызшэатдрмхпщьбудшщыхубвчочпщаэщялчохехалюидвиаммсееа
пегкажлхехдпрчиилмечшшшцкдщтечшчызшэатдрмлэчлрщнаэшэдкйчбйкишугрййкоыдднп
рщышлсбубеаунккмнежскгцчтыкйкавйыуйаусйрпносфнзвюаиейркезаокйщгаынрйщызюим
юдаюаыпмяызшцлгпшгцчтыкйкаяхбмщырйнхкелиачгшшдсдмэшсрмфукукчщгчилиачгшзс
ечмбрмфуэснарпзючшпмвпфчбшмейрпныурщгпзхцмчэиорщэаэшшщрщхезакдььрмьрпнх
щшдькюедефщроошкаюрпркдчэуырщлхчээпмеидбюхахщимюдюарппьщсрплаэщкаюытэте
дщпуэщвкющиулаэиыйхлллнажахоусиппрсеэщюхыййаькэиеыйееуйафмыущфзщжбглщейе
уозсащвашйымюдхунлищжанарпзючшбуосачиеэдщырйнхюахйщфрпешбероюарущефпкеза
рчцптддчщфдщпуэщвкющньйашегахлтейицмрйыезаокнейежпэиэщгэхувлуоыуыщимфмйщ
пшйрщьйапахпьююаяофэхувлуолиачйахагаодвимдчитысазшйыжжйажлчпнхыезахаэасачш
ашйарокамейецыьпяйхеейыуйаусйрнфйщхлюеерффасхйюдкемдсилэгерпйклижуашрщщей
ечшвппршгцчтыкйканущефптачштэрщзщяпэптбьерпимюдкеслщещцримежагекаюрэпьчяф
ьеруюсхпымздюлщелшашфьымосьрчифшцкщедеюакайасажлнктешщэилиачгшопьчффкмм
ьофпаюечэрщошбеюеюылшищгаясбрмэтдюадуклзщачисюарехеэдпрмэтдавнкхатешщашли
ачгшдчьнчиипяыачжижуыщашащышгпридчьнрифусицлщеомхпипчушгмщрщашгшмейрсе
мьюдкеипгекбхщвпчпжжйаайхлзаейуюфщроошэщнхльюаэпеямшщевлэияффубелшщфцчт
ыкйхрмсуювпьюыщдшварчмэчиащварщэщйщчшэийщхатешщчшбущефпсдюдисфуидчиеа
пячщ

ВТ:

однакоэтакартинаскакойбысторонымыеенирассматривалирасплываетсявнечтонеопределен
ноеприпадкипроявляющиесярезкосприкусываниемусиливающиесядоопасногодляжизнипр
иводящегоктяжкомусамокалечениюмогутвсежевнекоторыхслучаяхнедостигатьтакойсилыо
слабляясьдократкихсостоянийабсансадобыстропроходящихголовокруженийимогуттакжес

менятьсякраткимипериодамикогдабольнойсовершаетчуждыеегоприродепоступкикакбынах
одясьвовластибессознательногообуславливаясьвобщемкакбыстранноэтониказалосьчистоте
лесными причинами этисостояниямогутпервоначальновозникатьпопричинамчистодушевны
миспугилимогутвдальнейшемнаходитьсявзависимостиотдушевныхволненийкакнихаракте
рнодляогромногобольшинстваслучаевинтеллектуальноеснижениеноизвестенпокрайнейме
реодинслучайкогдаэтотнедугненарушилвысшейинтеллектуальнойдеятельностигельмгольц
другиеслучаивотношениикоторыхутверждалосьтожесамоененадежныилиподлежатсомнени
юкакислучайсамогодостоевскоголицастрадающиеэпилепсиеймогутпроизводитьвпечатлени
етупостинедоразвитоститкакэтаболезньчастосопряженасярковыраженнымидиотизмомик
рупнейшимимозговымидефектаминеявляяськонечнообязательнойсоставнойчастьюкартин
ыболезниноэтиприпадкисовсемисвоимивидоизмененияхмибываютиудругихлицулицсполны
мдушевнымразвитиемискореесосверхобычнаявбольшинствеслучаевнедостаточноуправляе
мойимиаффективностьюнеудивительночтопритакихобстоятельствахневозможноустановит
ьсовокупностьклиническоюаффектаэпилепсииточтопроявляетсяводнородностиуказанныхс
имптомовтребуетповидимомуфункциональногопониманиякакеслибымеханизманормально
говысвобожденияпервичныхпозывовбылподготовленорганическимеханизмкоторыйиспольз
уетсяприналичииивесьмаразныхусловийкакпринарушениимозговойдеятельностипритяжко
мзаболеванииткани йилитоксическомзаболеваниитакипринедостаточномконтроледушевной
экономиикризисномфункционированиидушевнойэнергиизаэтимразделениемнадвавидамыч
увствуемндентичностьмеханизмалежащеговосновевысвобожденияпервичныхпозывовэтот
механизмнедалекиотсексуальныхпроцессовпорождаемыхвсвоейосноветоксическиужедрев
нейшиеврачиназываликоитусмалойэпилепсиейивиделивполовомактесмягчениеиадаптаци
ювысвобожденияэпилептическогоотводараздраженияэпилептическаяреакциякаковымимен
емможноназватьвсеэтовместевзятоенесомненнотакжепоступаетиврасположениеневрозасу
щностькотороговтомчтобыликвидироватьсоматическимассыраздраженияскоторыминевроз
неможетсправитьсяпсихическиэпилептическийприпадокстановитсятакимобразомсимптом
омистериииеюадаптируетсяивидоизменяетсяподобнотомукакэтопроисходитпринормально
мтечениисексуальногопроцессатакимобразоммысполнымправомразличаемоганическуюи
аффективнуюэпилепсиюпрактическоезначениеэтогоследующеестрадающийпервойпораже
нболезньюмозгастрадающийвторойневротиквпервомслучаедушевнаяжизньподверженанар
ушениюизвневовторомслучаенарушениеявляетсявыражениемсамойдушевнойжизнивесьма
вероятночтоэпилепсиядостоевскогоотносится ковторомувидуточнодоказатьэтонельзятаккак
втакомслучаенужнобылобывключитьвцелокупностьегодушевнойжизниначалоприпадковип
оследующиевидоизмененияэтихприпадковадляэтогоунаснедостаточноданныхописаниясам
ихприпадковничегонедаютсведенияосоотношенияхмеждуприпадкамииипереживанияминеп
олныичастопротиворечивывсеговероятнеепредположениечтоприпадкиначалисьудостоевск
огоужевдетствечтоонивначалехарактеризовалисьболееслабымисимптомамиитолькопослеп
отрясшегоегопереживаниянавосемнадцатомгодужизниубийстваотцапринялиформуэпилепс
иибылобывесьмауместноеслибыоправдалосьточтоониполностьюпрекратилисьвовремяотб
ыванияимкаторгивсибириноэтомупротиворечатдругиеуказанияочевиднаясвязьмеждуотцеу
бийствомвбратьяхкарамазовыхисудьбойотцадостоевскогобросиласьвглазанеодномубиогра
фудостоевскогоипослужилаимуказаниемнаизвестноесовременноепсихологическоенаправл
ениепсихоанализтаккакподразумеваетсяименноонсклоненвидетьвэтомсобытиитягчайшуют
равмуиврекциидостоевскогонаэтоключевойпунктегоневрозаеслияначнуобосновыватьэтуу
становкупсихоаналитическиопасаюсьчтоокажусьнепонятнымдлявсехтехкомунезнакомыуч
ениеивыраженияпсихоанализаунасодиннадежныйисходныйпунктнамизвестенсмыслпервы
хприпадковдостоевскоговегоюношескиегодызадолгодопоявленияэпилепсииуэтихприпадко
вбылоподобиесмертиониназывалисьстрахомсмертиивыражалисьвсостоянииилетаргическог

оснаэтаболезньнаходилананеговначалекогдаонбылещемальчикомкаквнезапнаябезотчетная подавленностьчувствокаконпозжерассказывалсвоемудругусоловьевутакоекакбудтобыемуп редстоялосейчасжеумеретьивсамомделенаступалосостояниесовершенноподобноедействит ельнойсмертиегобратандрейрассказывалчтофедоружевмолодыегодыпередтемкакзаснутьос тавлялзаписькичтобоитсяночьюзаснутьсмертоподобнымсномипроситпоэтомучтобыегопохо ронилитолькочерезпятьднейдостоевскийзарулеткойвведениеснамизвестнысмыслинамерен иетакихприпадковсмертониозначаютотождествлениесумершимчеловекомкоторыйдейств ительноумерилисчеловекомживымещенокоторомумыжелаемсмертивторойслучайболеезнач ителенприпадоквуказанномслучаеравноцененнаказаниюмыпожелалисмертидругомутеперь мысталисамиэтимдругимисамиумерлитутпсихоаналитическоеучениеутверждаетчтоэтотдр угойдлямальчикаобычноотециименуемыйистериейприпадокявляетсятакимобразомсамонак азаниемзапожеланиесмертиненавистномуотц

Висновки

У ході виконнання лабораторного практикуму ми дослідили афінну біграмну підстановку. Була написана програма, що розшифровує текст шляхом знаходження ключів a і b і розв’язання системи лінійних рівнянь. Додатково була написана функція, що перевіряє розшифровані тексти на змістовність шляхом порівняння ентропії тексту з значенням ентропії з лабораторного практикуму 1. Набули навички частотного аналізу на прикладі розкриття моноалфавітної підстановки та опанували прийоми роботи в модулярній арифметиці.