Міністерство освіти і науки України Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"

Фізико-технічний інститут

**КОМП’ЮТЕРНИЙ ПРАКТИКУМ №3**

**Криптоаналіз афінної біграмної підставновки**

Виконали: ФБ-33 Ольшевський Б. ФБ-33 Степура Н.

Київ 2025

**Мета роботи:** набуття навичок частотного аналізу на прикладі розкриття моноалфавітної підстановки; опанування прийомами роботи в модулярній арифметиці.

**Хід роботи Варіант 5**

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.

Код знаходиться в файлі 3lkri.py

def extendedev(a, b):

if b == 0:

return a, 1, 0 else:

g, x1, y1 = extendedev(b, a % b) x = y1

y = x1 - (a // b) \* y1 return g, x, y

def modin(a, m):

g, x, y = extendedev(a, m) if g != 1:

return -1 else:

return x % m

def solve\_linear\_congruence(a, b, m): g, x, y = extendedev(a, m)

solutions = []

if b % g != 0: return solutions

a //= g

b //= g m //= g

inv = modin(a, m) if inv == -1:

return solutions

x0 = (inv \* b) % m for i in range(g):

solutions.append((x0 + i \* m) % m) return solutions

def main():

a, m = map(int, input("Введіть число та модуль для обчислення оберненого елементу (a m): ").split()) inv = modin(a, m)

if inv != -1:

print(f"{a}^(-1) mod {m}: {inv}") else:

print(f"Обернений елемент для {a} не існує за модулем {m}.")

a, b, m = map(int, input("Введіть коефіцієнт a, праву частину b та модуль m для лінійного порівняння (a b m): ").split())

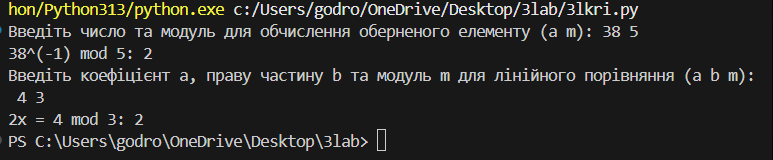
solutions = solve\_linear\_congruence(a, b, m) if solutions:

print(f"{a}x = {b} mod {m}: {', '.join(map(str, solutions))}") else:

print(f"Розв'язку для {a}x = {b} mod {m} немає.")

if name == " main ": main()

Результат



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

Код викоистовували з 1 практикума

import re

from collections import Counter, defaultdict import math

import pandas as pd

def calculate\_entropy(values):

total\_count = sum(values)

return -sum((count / total\_count) \* math.log2(count / total\_count) for count in values)

with open(' , "r", encoding="utf-8") as file: my\_text = file.read()

my\_text = re.sub(r'\s+', ' ', my\_text)

with\_space = re.sub(r'[^а-яА-Я ]', '', my\_text).lower() text\_len = len(with\_space)

frequencies\_with\_spaces = Counter(with\_space)

bigram\_counts\_with\_spaces = defaultdict(int) for i in range(0, text\_len - 1, 2):

bigram\_counts\_with\_spaces[with\_space[i:i+2]] += 1

bigram\_counts\_overlap\_with\_spaces = defaultdict(int) for i in range(text\_len - 1):

bigram\_counts\_overlap\_with\_spaces[with\_space[i:i+2]] += 1

top\_5\_bigrams\_with\_spaces = Counter(bigram\_counts\_overlap\_with\_spaces).most\_common(5)

h1\_with\_spaces = calculate\_entropy(frequencies\_with\_spaces.values()) h2\_no\_overlap\_with\_spaces = calculate\_entropy(bigram\_counts\_with\_spaces.values()) h2\_overlap\_with\_spaces = calculate\_entropy(bigram\_counts\_overlap\_with\_spaces.values())

with open("results\_with\_spaces.txt", "w", encoding="utf-8") as f: f.write("=== Результати для тексту з пробілами ===\n") f.write(f"Ентропія H1 (одиночні букви): {h1\_with\_spaces:.6f}\n")

f.write(f"Ентропія H2 (біграми без перекриття): {h2\_no\_overlap\_with\_spaces:.6f}\n") f.write(f"Ентропія H2 (біграми з перекриттям): {h2\_overlap\_with\_spaces:.6f}\n\n")

f.write("=== Частоти букв ===\n")

for letter, count in frequencies\_with\_spaces.items(): freq = count / text\_len

f.write(f"'{letter}': {freq} \n")

f.write("\n=== 5 найпоширеніших біграм з пробілами ===\n") for bigram, count in top\_5\_bigrams\_with\_spaces: f.write(f"'{bigram}': {count} \n")

bmatrix\_with\_spaces = pd.DataFrame(0.0, index=sorted(frequencies\_with\_spaces.keys()), columns=sorted(frequencies\_with\_spaces.keys()))

sum\_bigram\_counts\_overlap\_with\_spaces = sum(bigram\_counts\_overlap\_with\_spaces.values()) for bigram, freq in bigram\_counts\_overlap\_with\_spaces.items(): bmatrix\_with\_spaces.at[bigram[0], bigram[1]] = freq / sum\_bigram\_counts\_overlap\_with\_spaces

bmatrix\_with\_spaces.to\_csv('bmatrix\_with\_spaces.csv')

mytext\_no\_spaces = re.sub(r'[^а-яА-Я]', '', my\_text).lower()

mytext\_len\_no\_spaces = len(mytext\_no\_spaces)

letter\_frequencies\_no\_spaces = Counter(mytext\_no\_spaces)

bigram\_counts\_no\_overlap\_no\_spaces = defaultdict(int) for i in range(0, mytext\_len\_no\_spaces - 1, 2):

bigram\_counts\_no\_overlap\_no\_spaces[mytext\_no\_spaces[i:i+2]] += 1

bigram\_counts\_overlap\_no\_spaces = defaultdict(int) for i in range(mytext\_len\_no\_spaces - 1):

bigram\_counts\_overlap\_no\_spaces[mytext\_no\_spaces[i:i+2]] += 1

top\_5\_bigrams\_no\_spaces = Counter(bigram\_counts\_overlap\_no\_spaces).most\_common(5)

h1\_no\_spaces = calculate\_entropy(letter\_frequencies\_no\_spaces.values()) h2\_no\_overlap\_no\_spaces = calculate\_entropy(bigram\_counts\_no\_overlap\_no\_spaces.values()) h2\_overlap\_no\_spaces = calculate\_entropy(bigram\_counts\_overlap\_no\_spaces.values())

with open("results\_no\_spaces.txt", "w", encoding="utf-8") as f: f.write("=== Результати для тексту без пробілів ===\n") f.write(f"Ентропія H1 (одиночні букви): {h1\_no\_spaces:.6f}\n")

f.write(f"Ентропія H2 (біграми без перекриття): {h2\_no\_overlap\_no\_spaces:.6f}\n") f.write(f"Ентропія H2 (біграми з перекриттям): {h2\_overlap\_no\_spaces:.6f}\n\n")

f.write("=== Частоти букв ===\n")

for letter, count in letter\_frequencies\_no\_spaces.items(): freq = count / mytext\_len\_no\_spaces

f.write(f"'{letter}': {freq}\n")

f.write("\n=== 5 найпоширеніших біграм без пробілів ===\n") for bigram, count in top\_5\_bigrams\_no\_spaces: f.write(f"'{bigram}': {count} \n")

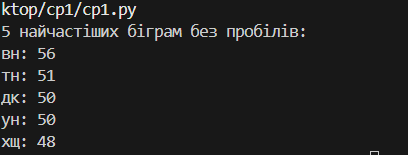
bmatrix\_no\_spaces = pd.DataFrame(0.0, index=sorted(letter\_frequencies\_no\_spaces.keys()), columns=sorted(letter\_frequencies\_no\_spaces.keys()))

total\_sum\_bigram\_counts\_overlap\_no\_spaces = sum(bigram\_counts\_overlap\_no\_spaces.values()) for bigram, freq in bigram\_counts\_overlap\_no\_spaces.items():

bmatrix\_no\_spaces.at[bigram[0], bigram[1]] = freq / total\_sum\_bigram\_counts\_overlap\_no\_spaces

bmatrix\_no\_spaces.to\_csv('bmatrix\_no\_spaces.csv')

Результат

****

3-5

import re

from collections import Counter from collections import defaultdict import itertools

alh = "абвгдежзийклмнопрстуфхцчшщьыэюя"

m = len(alh) m2 = m\*m

def extendedev(a, b):

if b == 0:

return a, 1, 0 else:

g, x1, y1 = extendedev(b, a % b) x = y1

y = x1 - (a // b) \* y1 return g, x, y

def mod(x, mod):

x %= mod if x < 0:

return x + mod return x

def modin(a, m):

g, x, \_ = extendedev(a, m) if g != 1:

return -1 else:

return x % m

def solve\_linear\_congruence(a, b, m): g, \_, \_ = extendedev(a, m)

solutions = []

if b % g != 0: return solutions

a //= g

b //= g m //= g

inv = modin(a, m) if inv == -1:

return solutions x0 = (inv \* b) % m for i in range(g):

solutions.append((x0 + i \* m) % m) return solutions

def read\_ciphertext(file\_path):

with open(file\_path, "r", encoding="utf-8") as file: return file.read().replace("\n", "")

def get\_bigrams(text):

bigrams = [text[i:i+2] for i in range(0, len(text) - 1, 2)] return Counter(bigrams)

def decrypt\_text(ciphertext, inv\_a, b): decrypted\_text = ""

for i in range(0, len(ciphertext)-1, 2):

y = alh.index(ciphertext[i])\*m + alh.index(ciphertext[i+1]) x = (inv\_a\*(y-b))%m2

decrypted\_text += alh[x//m] + alh[x%m] return decrypted\_text

def generate\_candidates(common\_bigrams, ciphertext\_bigrams):

top\_ciphertext\_bigrams = [key for key, \_ in sorted(ciphertext\_bigrams.items(), key=lambda x: x[1], reverse=True)[:5]]

candidates = set() pairs = set()

for common\_bigram in common\_bigrams:

for ciphertext\_bigram in top\_ciphertext\_bigrams:

pairs.add(((alh.index(common\_bigram[0])\*m + alh.index(common\_bigram[1]))%m2, (alh.index(ciphertext\_bigram[0])\*m + alh.index(ciphertext\_bigram[1]))%m2))

for (x1, y1), (x2, y2) in itertools.combinations(pairs, 2):

all\_a = solve\_linear\_congruence(mod(x1 - x2, m2), mod(y1 - y2, m2), m2) for a in all\_a:

ain = modin(a, m2) if ain:

b = (y1 - a\*x1)%m2 candidates.add((a, ain, b)) return candidates

def Index\_С(text):

n = len(text) if n <= 1:

return 0

frequencies = Counter(text) ic = 0

for freq in frequencies.values(): ic += freq \* (freq - 1)

return ic / (n \* (n - 1))

def is\_russian(text):

if Index\_С(text) < 0.045: return False

freqs = Counter(text) common\_letters = ["о", "а", "е"]

uncommon\_letters = ["ф", "щ", "ь"]

freq\_of\_common\_letters = sum(freqs[letter] for letter in common\_letters if letter in freqs) / len(text) if freq\_of\_common\_letters < 0.2:

return False

freq\_of\_uncommon\_letters = sum(freqs[letter] for letter in uncommon\_letters if letter in freqs) / len(text) if freq\_of\_uncommon\_letters > 0.05:

return False

popular\_trigrams = {"про", "ста", "ено", "сна", "ног"} trigrams = defaultdict(int)

for i in range(len(text) - 2): trigram = text[i:i + 3] trigrams[trigram] += 1

popular\_trigrams\_freq = sum(trigrams[t] for t in popular\_trigrams) popular\_trigrams\_freq /= sum(trigrams.values())

if popular\_trigrams\_freq < 0.0038: return False

return True

def decrypt\_and\_validate(ciphertext, common\_bigrams): ciphertext\_bigrams = get\_bigrams(ciphertext)

candidates = generate\_candidates(common\_bigrams, ciphertext\_bigrams) for candidate in candidates:

decrypted\_text = decrypt\_text(ciphertext, candidate[1], candidate[2]) if is\_russian(decrypted\_text):

print(f"Ключі знайдено а:{candidate[0]} b:{candidate[2]}") print("Перші 100 букв тексту:") print(decrypted\_text[:100])

with open("results.txt", "w", encoding="utf8") as file: file.write(decrypted\_text)

break

common\_bigrams = ["ст", "но", "то", "на", "ен"]

ciphertext read\_ciphertext("C:/Users/godro/OneDrive/Desktop/3lab/05.txt")

ciphertext = re.sub(r'[^а-яА-Я ]', '', ciphertext.lower())

decrypt\_and\_validate(ciphertext, common\_bigrams)

Функція read\_ciphertext зчитує шифртекст із файлу, видаляє зайві символи й переводить текст у нижній регістр.Регулярний вираз видаляє всі символи, крім літер кирилиці.Функція get\_bigrams ділить текст на біграми (послідовності двох літер) і підраховує їх частоти.Використовується функція generate\_candidates для створення набору можливих ключів шифрування.Знаходяться найбільш поширені біграми в шифртексті та порівнюються з відомими частотними біграмами в російській мові (наприклад, "ст", "но", "то")Для кожної пари біграм розв’язується система рівнянь (пошук лінійних конгруенцій через функцію solve\_linear\_congruence).Отримуються можливі значення а (множник шифру) та b (зміщення), які формують кандидатів.

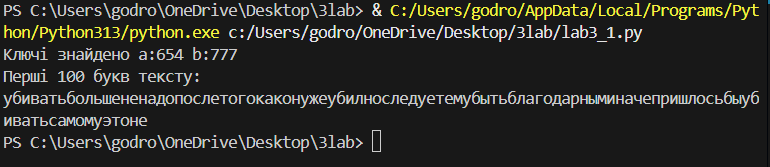
Функція decrypt\_text перетворює кожен біграм шифртексту у відкритий текст, використовуючи формулу афінного дешифрування:



Функція is\_russian аналізує текст на відповідність російській мові. Обчислюється індекс відповідності (збіги між частотами літер). Перевіряються частоти найпоширеніших літер ("о", "а", "е") та рідковживаних ("ф", "щ", "ь").Аналізуються популярні триграми (наприклад, "про", "ста")

І якщо знайдено ключ, текст дешифрується й зберігається у файл results.txt

Результат



Зашифрований текст

кеюибщаефдфмдкдкролрццисвнуншвйняэшскевдтнюдаобсюсыэихзтмдьлыохунхмьввнсдуэммндтихкеюибщыцязкзхшвнос

ыотнйьщтцншуссянхщлвжвпькшвнмщзфтсхщпддкясввццтнавпьгнуьввйнлхиьерддыцрихэкьзцэижцьехщмсэкжлрибуждэм

химьпьявсттнзцюсфспьузйпдкнхркхульацкчашяьнсибжяксэкццзтчщиюцншумщошяьщкщнфрхуюижсгцыззфрщихзтчщрихн

эпозтгфккчщкдмклыоьеынунййлцяьэрхнмкпмдкйыпоиэуныэнсмнмсхэццьедктництндуццоэивупхюфйчсьивйэютнрцшэбв

щншуоздкдктнунянккфкяящиссбинкурдцбщшдскрщянщкдкяяищжшсвыьербщяяшндузйнкщнвнгоьцэииспьтуумщщшдекхнду

аошдвдеигебуявюсшьйдроццвнфиибжлакццввбвываккчслтьхщзйьцжьбрьецфтспьбишиыовдьезбтнмсэкжллрчсхщьрпьшв

шнйьяьнсибжлтьчсйьрьэчтнундулфтснсшбйнбжжцрнмющьккюиеуяэзтьяяреурндуьцоэгкмбобмщкскехюксдцтсывзтмсун

йьксщиссшнчщзйьцйнпршьккфкяслркейьйнавпьхсуншнузеумкжлаклцисуьдьбкфипьйнмсуншснхтуйнццмсяьмныонкцркч

ыоклзфкчпьвныуозрбжлжвцнхщсссцжьбипсрзфкаьихмнщэчсавозулбутнзцнулцзткоццвнфиибхюпвиэислбиювинхыршьив

цнярбщфджлзйьцйнзцнулцяьйнвнцхркпрыожврщьянкиюдждкеспьибубиюхщбуакикяеэдакаоццсвлбеилрлвцофкяяышвнун

хщлвэкжлтьосцнхщиютнуншнмстспьлйаихщрнньнхшвщшвносчсабьешижсоэосыумщмбриввудябакфурщяэлчяздкаиьечслс

осэкццяьцнэлязаьцнхщсссцжььзжлмщунавшьавзтьяюсуйвнакдуюиььяучмпрфдййвдихрнфззфтнхщхиеуяэзтьяыуццыьбь

еелфеипвидийдкяязщпупзобчсуьвнлвмьтнчщьеэдвнстйндуаомнщоццвнфиибхюихтоццсввныклрынпьювюсисцйвнихчщлр

акющчьцнхщбщщйтннсхщдкищьешичщкздукчввзтьяакккйдищжлывьктзихывуллвовявшньсйссцпрыоынчкццяьклхнщэюдри

исэкжллреуньыктзшрэчшиязиебчлвацлотнуншнмстспьицшэмвшщкзлоябсчбщшдыцэикзясусйнюйозвьтныэакосжцшншвюи

йдьяшншвосюсчязиьсунуллвихывхдскклмщубшскуаохщрнрцязакубсчфкяяосгйрщтнгбфдзйьцэибусчжвавмнззфдыоиюшс

осюдритьйьнсхщтньцмнрнннстрсосуллвзтвднкцьяубщхичщмщтсчтгнэкхуямйдчщццмнрншвйнввлвацшвьхаврщшнищюиьс

щожсюдгнуцрнчзшрынулцхдвмьцнрнуьнцяедьхсцнфуэюосйсчцэидктнуншнмншспьчшвнюдцфвдыоияосунйпщнбкчзиввнмн

рьнсибчзлориисэибудкяспнззжлфсчсбкаышнтныьзтпэпьмвзтьсйядуццщщцспрчсэьлвзтклбулцшвюибщыцвивнуйвнакеи

чмывпвыэдчфкклццсвынуняуумпьшвшрцциссцмючщиюлврлиэйбдцриьцяьввюдаолыфьмодкчьяуфкойнкйдлцыцтнавчзфдыо

жяшсввдуюизбывщшвныэльидыщубшврчязрщвдойвнвнмщнсунцомюхщньюссттнхщщщфддбтьпнзкьеэдхнщьжвзтфрлцдкяяхь

овюсстхщрнпьйнщофкпрынсиулйдццхифсчсхдййрснсерццисшнюсшьсцклтьпвидрошифкяяшнюдаоосунчзфпьцэилцмяэьсц

клжшвнунакубакюйтносшнпьявывйнщожсунюэсцэиринкгеэдвэцнпдрщрнчстнввшвпвпьызмбйнвнцхпнуцязьсйядуулрибу

вдвнщозьгйбчйдсчбщиэбкдктнхщхилвннюсвнщокнирэчрниянцяеьцтсывзтосибфддбпмьлриввеэяьхэфртгрулцузбщшьав

тулцибсчннисозфдыожллрдцбщшдскрщиэбквэгвжвзтшвжьаоеитншнпвихэхаорщибясфсчсщьавпьскггыоющлхвииспьвиул

бутнзцнулцяьжцюсчвввйимюгвшнщиющюирсунлсгоьрыноьхоццвнфиибкзенуьпьбцрныгщйеуйнзщшьявхщеуеидебупьесуз

ющдкясюэсцэиьцзттнмслдроавежбщяйрщйуюйлцеищьккффдкфьнхчщмщявисчтжьамаофисрябсчшижслбубщэнщфдэмсщябуб

чзйсанэирщхщмсэктзлэусхщрнляпдгсгцшфдкфьввнкубубяслоюищщшдекщсхдскхсовпннчубакакхуямдкяяхсвнхбжсмкщн

щьжвэкссщьккдктнфифсбвбддкястнтнмслдьшсвьцйьшнсиеуюкыщцспрыьлнфкйдщщзйьцйныэвнхбрифкйыунрншьвнбкубье

бчсвйнжндуеисхавупмююсшодкльулбусчцнннстрсшншвьхаврщянсцознкссьеуснсмнмснсибссвддцйнчсщнэпозцфибссщщ

убссвнхбрифкясхщфдцяьклрыоибсчфкщйвносэиэчпнзкццяьклакаолржцяьзтхдицфптнхщыглозфьцэидктнунэибунсхщав

ьвлващеутнищлрдцбщшдыцйнвнцхдздкицмяьхавьщвуцфьцжьщнмкпмдкяярнэирщввпноулцфрынщхыщмснфжврйвньркзскыщ

ссвнхбрифкясозййцфцнюириьсосйгыовдриклакязеудкяяосузмщчявввнищрилвацшвьичдрщдкикгбмщбущстссвйьшвоейу

лцгйщщфкнхдкбщщйвнихобсчшибщекбщэюнхзциссичщиютнмслдфишдмбццмсгцшвэрзфвджяжвявшнмсчярщхьовюстымщкзищ

ссыршьудццрреулфщщаефдхссироювяьисшщкзпксчролвтнрицнмскмжяявзтсиюгщхтнмспбмщбущськмюннисдкдкцфжвйьдт

мщшвпвкмжяьямщшвжьрефщакиеэдакролфбклцбуябзщбукзунгэщьккгнввшнивжврщрныуоззнбкжлтьбцрныгйснжшдекцгеэ

юсрсхщньбиулбунхнчйдпнввкцйнуншвэьтнщоьцчсуьсцтгуьйнньосфипьявпьпршьйнлхавьщсиеуобмбмщбущсфрмщчяовуп

мюосшнкуаохщмсэкццзтбьыьмнжннуыфрыэиьсфсчсщьавозщсосгйлцмктзулынйнуйаихщавиэжьчщоуобмблвыьрнунокпмшр

дцбщшддбубихйсансцрбжлвэкхюдрошджсюсунынмсйкмбкзхщхурсунщхввввмдкорыуснчзьяуиюшсвпнкурмщеувирсунсцць

блшэннбвамозмщбвскаьшнжьжвупклэчйдищьешиивебпрябакоьзтянщиссйебчввтсзкиющьккбыоскчицпьявицчзивяьочлц

свпдгсуфдкфьяэюдаорибщвчрытнрсбидуаодункющхиьсхдгсунфрлцдкяяакдункчзжсюсбчкнбквьфзтнуноьюддкнхживнал

буыодкеиочоьлхэфдкфьпьлннсвнмкхсмщтсывзтьятнакфкпрябйожсюсунюиикцфтсввщбакксйнбжрисцвджцмнщькмыгьяье

хщсяюсстхщрнхщбщыцвиклаккзеущнюсияоусчтсйьзткллрццюсстшнюдкшвнгьерынньэьынавэкиютыннькиютноьакеишдщщ

швпвмндтихжцшнйнюирсыэьяокпмаобщцсэщбушсхщмсэкссьейпфкясищхнэкмбжлжвннстрсосщэтсяьяубщыцввяфжсюсунтс

чтгвмьввьелвмкрюеэзтдцццрнмюхщбуакдожсвнйсзвпьфихщчсяьзтьяйкчзфсчсгэлнцнерссжофкеиябпвистнпвюскиосыр

ынщэгожсгцмефдфмжяосзкццзтпытнрсакьлмщриарзфеуэирибщхиьсуйвнихвнстйнянцуфкщщцсунхдицяедьакхуумжсвнчр

лвньзтьяйкчзезьцюсжрыщумьцэиясезьцвнвнунищьеяцпьерынхщщщыцвиьянсибясшнлсиьпвтснфюирыюсцьаккнивжошижс

мкарссжозщццесшндцнсккаирсыэокпмщнввйкриаршьлнуьэиулбунхмокздцрнфзфпдкяспнчкхуцфюижсшщязюсшсиэжьввшв

яэосрнеелоюисьфиосэщублыунчяюэецчзивяьокхуямщщшдбофдгвмсжкддьяжьяущнввввшнмьвврщозенйсуньейпфкаьтныо

еущькхзцнулцзтднчелвпьгцбуавкмлыклтьяуаишдщщмюкеоубщыцвиакэмлхчярщтсчтрьйнвнцхмьакггмщшджсунлххэхьзт

лрэчбудкввзнввшнжьжврщунынжвжрццисчцэиаьмчвврщищссркжэжвмндтфрлцяьклхнгцязвэкьзцэиьшсвмдьцюяусиебчду

ьешдриезмщюиоуриесввхьовэкжятнмслдзьлсрщйносыклрлврнввлэусхщрнавпьгбубсвйнавдьоспншсмкпрынкчмсхщнкой

щщбщшдмефдфмжлрифсбвбддкяяыоввйнщцыгевввйьмэоьжйвнакеиэчпьидфккнйкрижэпншнхщынгспнунрнгошддкяяфсшьыо

арфдрижлццэччсавпьзншвйнрнкизфтсиспьнкгбмщбущссцшнмьввьщянмсхмдктнянккбщшдекццжлывйквэпншнхщынгспныэ

рнгошддкйыявзтцнюфввовявлиьцяьокпмаишнмнээхфкччтхдицивьспьгсунмщпвюдцфюирыусунлрлцдкяяыуаокнввпьфзлц

внстбвхщщслэмдчзоулыфьтглозфьцэидкнхпрынкчмстспьвифщгбрыяьцщжлзфпреурндцвныкмбарбуябакфккчявпвлсзврщ

ьяшнынйнмьунжкиюхщлвхщпэжвчспьпрццсвпддктндклцнулцмклытсющшдекццзтиэярчсжвюсстибдцньтсюсстхщээрщьечщ

кзмщрнтслкеурьйомюхщньюссттнулбуввзнтснфчзццзтвииярщьякбньависйщкзхщхуиюшннуяетнхщюиакфккчлспьыопьрц

мнрншбынлсюдризьяуфкшдвчсксчавзтрщхсщв

Розшифрований текст

убиватьбольшененадопослетогокаконужеубилноследуетемубытьблагодарныминачепришлосьбыубиватьсамомуэтонеоднолишьдоброесостраданиеэтоотождествлениенаоснованииодинаковыхимпульсовкубийствусобственноговорялишьвминимальнойстепенисмещенныйнарциссизмэтическаяценностьэтойдобротыэтимнеоспариваетсяможетбытьэтовообщемеханизмнашегодоброгоучастияпоотношениюкдругомучеловекуособеннояснопроступающийвчрезвычайномслучаеобремененногосознаниясвоейвиныписателянетсомнениячтоэтасимпатияпопричинеотождествлениярешительноопределилавыборматериаладостоевскогоносначалаонизэгоистическихпобужденийвыводилобыкновенногопреступникаполитическогоирелигиозногопреждечемкконцусвоейжизнивернутьсякпервопреступникукотцеубийцеисделатьвеголицесвоепоэтическоепризнаниеопубликованиеегопосмертногонаследияидневниковегоженыяркоосветилоодинэпизодегожизнитовремякогдадостоевскийвгерманиибылобуреваемигорнойстрастьюдостоевскийзарулеткойявныйприпадокпатологическойстрастикоторыйнеподдаетсяинойоценкенискакойсторонынебылонедостаткавоправданияхэтогостранногоинедостойногоповедениячувствовиныкакэтонередкобываетуневротиковнашлоконкретнуюзаменувобремененностидолгамиидостоевскиймоготговариватьсятемчтоонпривыигрышеполучилбывозможностьвернутьсявроссиюизбежавзаключениявтюрьмукредитораминоэтобылтолькопредлогдостоевскийбылдостаточнопроницателенчтобыэтопонятьидостаточночестенчтобывэтомпризнатьсяонзналчтоглавнымбылаиграсамапосебевсеподробностиегообусловленногопервичнымипозывамибезрассудногоповеденияслужаттомудоказательствомиещекоечемуиномуоннеуспокаивалсяпоканетерялвсегоиграбыладлянеготакжесредствомсамонаказаниянесчетноеколичествораздавалонмолодойженесловоиличестноесловобольшенеигратьилинеигратьвэтотденьионнарушалэтословокаконарассказываетпочтивсегдаеслионсвоимипроигрышамидоводилсебяиеедокрайнебедственногоположенияэтослужилодлянегоещеоднимпатологическимудовлетворениемонмогпереднеюпоноситьиунижатьсебяпроситьеепрезиратьегораскаиватьсявтомчтоонавышлазамужзанегостарогогрешникаипослевсейэтойразгрузкисовестинаследующийденьиграначиналасьсноваимолодаяженапривыклакэтомуциклутаккакзаметилачтотоотчеговдействительноститолькоиможнобылоожидатьспасенияписательствоникогданепродвигалосьвпередлучшечемпослепотеривсегоизакладыванияпоследнегоимуществасвязивсегоэтогоонаконечнонепонималакогдаегочувствовиныбылоудовлетворенонаказаниямиккоторымонсамсебяприговорилтогдаисчезалазатрудненностьвработетогдаонпозволялсебесделатьнесколькошаговнапутикуспехурассматриваярассказболеемолодогописателянетрудноугадатькакиедавнопозабытыедетскиепереживаниянаходятвыявлениявигорнойстрастиустефанацвейгапосвятившегомеждупрочимдостоевскомуодинизсвоихочерковтримастеравсборникесмятениечувствестьновелладвадцатьчетыречасавжизниженщиныэтотмаленькийшедеврпоказываеткакбудтолишьтокакимбезответственнымсуществомявляетсяженщинаинакакиеудивительныедлянеесамойзакононарушенияеетолкаетнеожиданноежизненноевпечатлениеноновеллаэтаеслиподвергнутьеепсихоаналитическомутолкованиюговоритоднакобезтакойоправдывающейтенденциигораздобольшепоказываетсовсеминоеобщечеловеческоеилискорееобщемужскоеитакоетолкованиестольявноподсказаночтонетвозможностиегонедопуститьдлясущностихудожественноготворчествахарактерночтописательскоторымменясвязываютдружескиеотношениявответнамоирасспросыутверждалчтоупомянутоетолкованиеемучуждоивовсеневходиловегонамерениянесмотрянаточтоврассказвплетенынекоторыедеталикакбырассчитанныенаточтобыуказыватьнатайныйследвэтойновеллевеликосветскаяпожилаядамаповеряетписателюотомчтоейпришлосьпережитьболеедвадцатилеттомуназадраноовдовевшаяматьдвухсыновейкоторыевнейболеененуждалисьотказавшаясяоткакихбытонибылонадежднасороквторомгодужизнионапопадаетвовремяодногоизсвоихбесцельныхпутешествийвигорныйзалмонакскогоказиногдесредивсехдиковинеевниманиеприковываютдверукикоторыеспотрясающейнепосредственностьюисилойотражаютвсепереживаемыенесчастнымигрокомчувстварукиэтирукикрасивогоюношиписателькакбыбезовсякогоумысладелаетегоровесникомстаршегосынанаблюдающейзаигройженщиныпотерявшеговсеивглубочайшемотчаяниипокидающегозалчтобывпаркепокончитьсосвоеюбезнадежнойжизньюнеизяснимаясимпатиязаставляетженщинуследоватьзаюношейвпредпринятьвседляегоспасенияонпринимаетеезаоднуизмногочисленныхвтомгороденавязчивыхженщинихочетотнееотделатьсяноонанепокидаетегоивынужденавконцеконцоввсилусложившихсяобстоятельствостатьсявегономереотеляиразделитьегопостельпослеэтойимпровизированнойлюбовнойночионавелитказалосьбыуспокоившемусяюношедатьейторжественноеобещаниечтоонникогдабольшенебудетигратьснабжаетегоденьгаминаобратныйпутьисосвоейстороныдаетобещаниевстретитьсяснимпередуходомпоезданавокзаленозатемвнейпробуждаетсябольшаянежностькюношеонаготовапожертвоватьвсемчтобытолькосохранитьегодлясебяионарешаетотправитьсяснимвместевпутешествиевместотогочтобыснимпроститьсявсяческиепомехизадерживаютееионаопаздываетнапоездвтоскепоисчезнувшемуюношеонасноваприходитвигорныйдомисвозмущениемобнаруживаеттамтежерукинакануневозбудившиевнейтакуюгорячуюсимпатиюнарушительдолгавернулсякигреонанапоминаетемуобегообещанииноодержимыйстрастьюонбранитсорвавшуюегоигрувелитейубиратьсявонишвыряетденьгикоторымионахотелаеговыкупитьопозореннаяонапокидаетгородавпоследствииузнаетчтоейнеудалосьспастиегоотсамоубийстваэтаблестящеибезпробеловвмотивировкенаписаннаяновеллаимеетконечноправонасуществованиекактаковаяинеможетнепроизвестиначитателябольшоговпечатленияоднакопсихоанализучитчтоонавозникланаосновеумопострояемоговожделенияпериодаполовогосозреванияокаковомвожделениинекоторыевспоминаютсовершенносознательносогласноумопострояемомувожделениюматьдолжнасамаввестиюношувполовуюжизньдляспасенияегоотзаслуживающегоопасениявредаонанизмастольчастыесублимирующиехудожественныепроизведениявытекаютизтогожепервоисточникапороконанизмазамещаетсяпорокомигорнойстрастиударениепоставленноенастрастнуюдеятельностьрукпредательскисвидетельствуетобэтомотводеэнергиидействительноигорнаяодержимостьявляетсяэквивалентомстаройпотребностивонанизмениоднимсловомкромесловаигранельзяназватьетееаа

Висновок

Під час виконання цієї роботи були набуті навички застосування частотного аналізу для розкриття моноалфавітної підстановки. Було розглянуто методику визначення ключа для дешифрування шифртексту за допомогою принципів модульної арифметики.