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

```
hon/Python313/python.exe c:/Users/godro/OneDrive/Desktop/3lab/3lkri.py
Введіть число та модуль для обчислення оберненого елементу (a m): 38 5
38^(-1) mod 5: 2
Введіть коефіцієнт a, праву частину b та модуль m для лінійного порівняння (a b m):
 4 3
2x = 4 mod 3: 2
PS C:\Users\godro\OneDrive\Desktop\3lab>
```

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 використовували 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

```
ktop/cp1/cp1.py
5 найчастіших біграм без пробілів:
вн: 56
тн: 51
дк: 50
ун: 50
хщ: 48
```

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_C(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_C(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 a:{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 a (множник шифру) та b (зміщення), які формують кандидатів.
Функція decrypt_text перетворює кожен біграм шифртексту у відкритий текст, використовуючи формулу афінного дешифрування:

$$X_i = a^{-1}(Y_i - b) \bmod m^2$$

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

```
PS C:\Users\godro\OneDrive\Desktop\3lab> & C:/Users/godro/AppData/Local/Programs/Pyt
hon/Python313/python.exe c:/Users/godro/OneDrive/Desktop/3lab/lab3_1.py
Ключі знайдено a:654 b:777
Перші 100 букв тексту:
убиватьбольшененадопослетогокаконужеубилноследуетемубытьблагодарныминачепришлосьбыуб
иватьсамомуэтоне
PS C:\Users\godro\OneDrive\Desktop\3lab>
```

Зашифрований текст

кеюибщаефдфмдкдкролрццисвнуншвйняэшскевдтнюдаобсюсыэихзтмдьлыохунхмьвнсдуэммндтихкеюибщыцязкзхшвнос

ыотнйьщтцншуссянхщлвжвпькшвнмщзфтсхщпддкясввццтнавпьгнуьввйнлхиьерддыцрихэкьзцэижцьехщмсэкжлрибуждэм

химьпьявсттнзцюсфспьузйпдкнхркхульацкчашяьнсибжяксэкццзтчщиюцншумщошяьщкщнфрхуюижсгцыззфрщихзтчщрихн

эпозтгфккчщкдмклыоьеынуййлцяьэрхнмкпмдкйыпоиэуныэнсмнмсхэццьедктництндуццоэивупхюфйчсьивйэютнрцшэбв

щншуоздкдктнунянккфкяящиссбинкурдцбщшдскрщянщкдкяяищжшсвыьербщяяшндузйнкщнвнгоьцэииспьтуумщщшдекхнду

аошдвдеигебуявюсшьйдроццвнфиибжлакццввбвываккчслтьхщзйьцжьбрьецфтспьби шиыовдьезбтнмсэкжллрчсхщьрпьшв

шнйьяьнсибжлтьчсйьрьэчтнундулфтснсшбйнбжжцрнмющьккюиеуяэзтьяяреурндуьц оэгкмбобмщкскехюксдцтсывзтмсун

йьксщиссшнчщзйьцйнпршьккфкяслркейьйнавпьхсуншнузеумкжлаклцисуьдьбкфипь йнмсуншснхтуйнццмсяьмныонкцркч

ыоклзфкчпьвныуозрбжлжвцнхщсссцжьбипсрзфкаьихмнщэчсавозулбутнзцнулцзткоц цвнфиибхюпвиэислбиювинхыршьив

цнярбщфджлзйьцйнзцнулцяьйнвнцхркпрыожврщьянкиюджджкеспьибубиюхщбуакикя еэдакаоццсвлбеилрлвцофкяяышвнун

хщлвэкжлтьосцнхщиютнуншнмстспьлйаихщрнньнхшвщшвносчсабьешижсоэосыумщ мбриввудябакфурщяэлчяздкаиьечслс

осэкццяьцнэлязаьцнхщсссцжььзжлмщунавшьавзтьяюсуйвнакдуюиььяучмпрфдййвди хрнфззфтнхщхиеуяэзтьяыуццыьбь

еелфеипвидийдкяязщпупзобчсуьвнлвмьтнчщьеэдвнстйндуаомнщоццвнфиибхюихтоц цсввныклрынпьювюсисцйвнихчщлр

акющчьцнхщбщщйтннсхщдкищьешичщкздукчввзтьяакккйдищжлывьктзихывуллвовя вшньсйссцпрыоынчкццяьклхнщэюдри

исэкжллреуньыктзшрэчшиязиебчлвацлотнуншнмстспьицшэмвшщкзлоябсчбщшдыцэ икзясусйнюйозвьтныэакосжцшншвюи

йдьяшншвосюсчязиьсунуллвихывхдсккклмщубшскуаохщрнрцязакубсчфкяяосгйрщтнг бфдзйьцэибусчжвавмнззфдыоиюшс

осюдритьйьнсхщтньцмнрнннстрсосуллвзтвднкцьяубщхичщмщтсчтгнэкхуямйдчщццм нрншвйнввлвацшвьхаврщшнищюиьс

щожсюдгнуцрнчзшрынулцхдвмьцнрнуьнцяедьхсцнфуэюосйсчцэидктнуншнмншспьч швнюдцфвдыоияосунйпщнбкчзиввнмн

рьнсибчзлориисэибудкяспнззжлфсчсбкаышнтныьзтпэпьмвзтьсйядуццщщцспрчсэьлв зтклбулцшвюибщыцвивнуйвнакеи

чмывпвыэдчфкклццсвынуняуумпьшвшрцциссцмючщиюлврлиэйбдцриьцяьввюдаолы фьмодкчьяуфкойнкйдлцыцтнавчзфдыо

жяшсввдуюизбывщшвныэльидыщубшврчязрщвдойвнвнмщнсунцомюхщньюсстгнхщщ щфддбтьпнзкьеэдхнщьжвзтфрлцдкяяхь

овюсстхщрнпьйнщофкпрынсиулйдццхифсчсхдййрснсерццисшнюсшьсцклтьпвидрош ифкяяшнюдаоосунчзфпьцэилцмяэьсц

клжшвнунакубакюйтносшнпьявывйнщожсунюэсцэиринкгеэдвэцнпдрщрнчстнввшвпв пьызмбйнвнцхпнуцязьсйядуулрибу

вдвнщозьгйбчйдсчбщиэбкдктнхщхилвннюсвнщокнирэчрниянцяеьцтсывзтосибфддбп

мьлриввеэяьхэфртгрулцузбщшьав

тулцибсчннисозфдыожллрдцбщшдскрщиэбквэгвжвзтшвжьаоеитншнпвихэхаорщибяс фсчсщьавпьскггыоющлхвииспьвиул

бутнзцнулцяьжцюсчвввйимюгвшнщиющюирсунлсгоьрыноьхоццвнфиибкзенуьпьбцрн ыгщйеуйнзщшьявхщеуеидебупьесуз

ющдкясюэсцэиьцзттнмслдроавежбщяйрщйуюйлцеищьккффдкфьнхчщмщявисчтжьа маофисрябсчшижслбубщэнщфдэмсщябуб

чзйсанэирщхщмсэктзлэусхщрнляпдгсгцшфдкфьввнкубубяслоюищщшдекщсхдскхсов пннчубакакхуямдкяяхсвнхбжсмкщн

щьжвэкссщьккдктнфифсбвбддкястнтнмслдьшсвьцйьшнсиеуюкыщцспрыьлнфкйдщщ зйьцйныэвнхбрифкйыунрншьвнбкубье

бчсвйнжндуеисхавупмююсшодкльулбусчцнннстрсшншвьхаврщянсцознкссьеуснсмнм снсибссвддцйнчсщнэпозцфибссщщ

убссвнхбрифкясхщфдцяьклрыоибсчфкщйвносэиэчпнзкццяьклакаолржцяьзтхдицфпт нхщыглозфьцэидктнунэибунсхщав

ьвлващеутнищлрдцбщшдыцйнвнцхдздкицмяьхавьщвуцфьцжьщнмкпмдкяярнэирщвв пноулцфрынщхыщмснфжврйвньркзскыщ

ссвнхбрифкясозййцфцнюириьсосйгыовдриклакязеудкяяосузмщчявввнищрилвацшвь ичдрщдкикгбмщбущстссвйьшвоейу

лцгйщщфкнхдкбщщйвнихобсчшибщекбщэюнхзциссичщиютнмслдфишдмбццмсгцшвэ рзфвджяжвявшнмсчярщхьовюстымщкзищ

ссыршьудццрреулфщщаефдхссироювяьисшщкзпксчролвтнрицнмскмжяявзтсиюгщхт нмспбмщбущськмюннисдкдкцфжвйьдт

мщшвпвкмжяьямщшвжьрефщакиеэдакролфбклцбуябзщбукзунгэщьккгнввшнивжврщ рныуоззнбкжлтьбцрныгйснжшдекцгеэ

юсрсхщньбиулбунхнчйдпнввкцйнуншвэьтнщоьцчсуьсцтгуьйнньосфипьявпьпршьйнлх авьщсиеуобмбмщбущсфрмщчяовуп

мюосшнкуаохщмсэкццзтбыыьмнжннуыфрыэиьсфсчсщьавозщсосгйлцмктзулынйнуйа ихщавиэжьчщоуобмблвыьрнунокпмшр

дцбщшддбубихйсансцрбжлвэкхюдрошджсюсунынмсйкмбкзхщхурсунщхвввмдкорыу снчзьяуиюшсвпнкурмщеувирсунсцць

блшэннбвамозмщбвскаьшнжьжвупклэчйдищьешиивебпрябакоьзтянщиссйебчввтсзк иющьккбыоскчицпьявицчзивяьочлц

свпдгсуфдкфьяэюдаорибщвчрытнрсбидуаодункющхиьсхдгсунфрлцдкяякдункчзжс юсбчкнбквьфзтнуноьюддкнхживнал

буыодкеиочоьлхэфдкфьпьлннсвнмкхсмщтсывзтьятнакфкпрябйожсюсунюиикцфтсвв щбакксйнбжрисцвджцмнщькмыгьяье

хщсяюсстхщрнхщбщыцвиклаккзеущнюсияоусчтсйьзткллрццюсстшнюдкшвнгьерыннь эьынавэкиютыннькиютноьакеишдщщ

швпвмндтихжцшнйнюирсыэьяокпмаобщцсэщбушсхщмсэкссьейпфкясищхнэкмбжлжв ннстрсосщэтсяьяубщыцввяфжсюсунтс

чтгвмьввьелвмкрюеэзтдцццрнмюхщбуакдожсвнйсзвпьфихщчсяьзтьяйкчзфсчсгэлнцн ерссжофкеиябпвистнпвюскиосыр

ынщэгожсгцмефдфмжяосзкццзтпытнрсакьлмщриарзфеуэирибщхиьсуйвнихвнстйнян цуфкщщцсунхдицяедьакхуумжсвнчр

лвньзтьяйкчзезьцюсжрыщумьцэиясезьцвнвнунищьеяцпьерынхщщщыцвиьянсибясшн лсиьпвтснфюирыюсцьаккнивжошижс

мкарссжозщццесшндцнсккаирсыэокпмщнввйкриаршьлнуьэиулбунхмокздцрнфзфпдк яспнчкхуцфюижсшщязюсшсиэжьввшв

яэосрнеелоюисьфиосэщублыунчяюэецчзивяьокхуямщщшдбофдгвмсжкддьяжьяущнв вввшнмьвврщозенйсуньейпфкаьтныо

еущькхзцнулцзтднчелвпьгцбуавкмлыклтьяуаишдщщмюкеоубщыцвиакэмлхчярщтсчт рьйнвнцхмьакггмщшджсунлххэхьзт

лрэчбудквввзнввшнжьжврщунынжвжрццисчцэиаьмчвврщищссркжэжвмндтфрлцяьклх нгцязвэкьзцэиьшсвмдьцюяусиебчду

ьешдриезмщюиоуриесввхьовэкжятнмслдзьлсрщйносыклрлврнввлэусхщрнавпьгбубс вйнавдьоспншсмкпрынкчмсхщнкой

щщбщшдмефдфмжлрифсбвбддкяяыоввйнщцыгевввйьмэоьжйвнакеиэчпьидфккнйкр ижэпншнхщынгспнунрнгошддкяяфсшьыо

арфдрижлццэччсавпьзншвйнрнкизфтсиспьнкгбмщбущссцшнмьввьщянмсхмдктнянкк бщшдекццжлывйквэпншнхщынгспныэ

рнгошддкйыявзтцнюфввовявлиьцяьокпмаишнмнээхфкччтхдицивьспьгсунмщпвюдцф юирыусунлрлцдкяяыуаокнввпьфзлц

внстбвхщщслэмдчзоулыфьтглозфьцэидкнхпрынкчмстспьвифщгбрыяьцщжлзфпреур ндцвныкмбарбуябакфккчявпвлсзврщ

ьяшнынйнмьунжкиюхщлвхщпэжвчспьпрццсвпддктндклцнулцмклытсющшдекццзтиэя рчсжвюсстибдцньтсюсстхщээрщьечщ

кзмщрнтслкеурьйомюхщньюсстгнулбуввзнтснфчзццзтвииярщьякбньависйщкзхщхуи юшннуяетнхщюиакфккчлспьыопьрц

мнрншбынлсюдризьяуфкшдвчсксчавзтрщхсщв

Розшифрований текст

убиватьбольшененадопослетогокаконужеубилнаследуетемубытьблагодарныминачепри
шлосьбыубиватьсамомуэтонеоднолишьдоброесостраданиеэтоотождествлениенаоснова
нииодинаковыхимпульсовкубийствусобственноговорялишьвминимальнойстепенисмещ
енныйнарциссизмэтическаяценностьэтойдобротыэтимнеоспариваетсяможетбытьэтовоо
бщемеханизмнашегодоброгоучастияпоотношениюкдругомучеловекуособенноясноопрост
упающийвчрезвычайномслучаеобремененногосознаниясвоейвиныписателянетсомнени
ячтоэтасимпатияпопричинеотождествлениярешительноопределилавыборматериаладос
тоевскогоносначалаонизэгоистическихпобужденийвыводилобыкновенногопреступника
политическогоирелигиозногопреждечемкконцусвоейжизнивернутьсякпервопреступник
укотцеубийцеисделатьвеголицесвоепоэтическоепризнаниеопубликованиеегопосмертно
гонаследияидневниковегоженыяркоосветилоодинэпизодегожизнитовремякогдадостоев
скийвгерманиибылобуреваемигорнойстрастьюдостоевскийзарулеткойявныйприпадокпа
тологическойстрастикоторыйнеподдаетсяинойоценкенискакойсторонынебылонедостат
кавоправданияхэтогостранногоинедостойногоповедениячувствовиныкакэтонередкобыв
аетуневротиковнашлоконкретнуюзаменувобремененностидолгамиидостоевскиймоготго
вариватьсятемчтоонпривыигрышеполучилбывозможностьвернутьсяврассиюизбежавзак
люченияватюрьмукредитораминоэтобылтолькопредлогдостоевскийбылдостаточнопрони
цателенчтобыэтопонятьидостаточночестенчтобывэтомпризнатьсяонзналчтоглавнымбыл
аиграсамапосебевсеподробностиегообусловленногопервичнымипозывамибезрассудног
оповеденияслужаттомудоказательствомиещекоечемуиномуоннеуспокаивалсяпоканете
рялвсегоиграбыладлянеготакжесредствомсамонаказаниянесчетноеколичествораздавал
онмолодойженесловоиличестноесловобольшенеиграть илинеиграть вэтотденьионнаруш
алэтословокаконарассказываетпочтивсегдаеслионсвоимипроигрышамидоводилсебяиее
докрайнебедственногоположенияэтослужилодлянегоещеоднимпатологическимудовлет
ворениемонмогпереднеюпоноситьиунижатьсебяпроситьеепрезиратьегораскаиватьсявт
омчтоонавышлазамужзанегостарогогрешникаипослевсейэтойразгрузкисовестинаследу
ющийденьиграначиналасьсноваимолодаяженапривыклакэтомуциклутаккакзаметилачто
тоотчеговдействительноститолькоиможнобылоожидатьспасенияписательствоникогдан
епродвигалосьвпередлучшечемпослепотеривсегоизакладыванияпоследнегоимущества
связивсегоэтогоонаконечнонепонималакогдаегочувствовиныбылоудовлетворенонаказа
ниямиккоторымонсамсебяприговорилтогдаисчезалазатрудненностьвработетогдаонпозв
олялсебеделатьнесколькошаговнапутикуспехурассматриваярассказболеемолодогопис
ателянетрудноугадатькакиедавнопозабытыедетскиепереживаниянаходятвыявлениявиг
орнойстрастиустефанацвейгапосвятившегомеждупрочимдостоевскомуодинизсвоихоче
рковтримастеравсборникесмятениечувствестьновелладвадцатьчетыречасавжизниженщ
иныэтотмаленькийшедеврпоказываеткакбудтолишьтокакимбезответственнымсущество
мявляетсяженщинаинакакиеудивительныедлянеесамойзакононарушенияеетолкаетнео
жиданноежизненноевпечатлениеноновеллаэтаеслиподвергнутьеепсихоаналитическом
утолкованиюговоритоднакобезтакойоправдывающейтенденциигораздобольшепоказыва
етсовсеминоеобщечеловеческоеилискореообщемужскоеитакоетолкованиестольявнопо
дсказаночтонетвозможностиегонедопуститьдлясущностихудожественноготворчестваха

рактерночтописательскоторымменясвязываютдружескиеотношениявответнамоирасспр
осыутверждалчтоупомянутоетолкованиеемучуждоивовсеневходиловегонамерениянесм
отрянаточтоврассказвплетенынекоторыедеталикакбырассчитанныенаточтобыуказывать
натайныйследвэтойновеллевеликосветскаяпожилаядамаповеряетписателюотомчтоейп
ришлосьпережитьболеедвадцатилеттомуназадранооводовевшаяматьдвухсыновейкоторы
евнейболеененуждалисьотказавшаясяоткакихбытонибылонадежднасороквторомгодужи
знионапопадаетвовремяодногоизсвоихбесцельныхпутешествийвигорныйзалмонакского
казиногдесредивсехдиковинеевниманиеприковываютдверукикоторыеспотрясающейне
посредственностьюисилойотражаютвсепереживаемыенесчастнымигрокомчувстварукиэ
тирукикрасивогоюношиписателькакбыбезовсякогоумысладелаетегоровесникомстаршег
осынанаблюдающейзаигройженщиныпотерявшеговсеивглубочайшемотчаяниипокидаю
щегозалчтобывпаркепокончитьсосвоеюбезнадежнойжизньюнеизяснимаясимпатиязаста
вляетженщинуследоватьзаюношейвпредпринятьвседляегоспасенияонпринимаетеезаод
нуизмногочисленныхвтомгороденавязчивыхженщинихочетотнееотделатьсяноонанепок
идаетегоивынужденавконцеконцоввсилусложившихсяобстоятельствостатьсявегономер
еотеляиразделитьегопостельпослеэтойимпровизированнойлюбовнойночионавелитказа
лосьбыуспокоившемусяюношедатьейторжественноеобещаниечтоонникогдабольшенебу
детиграть снабжаетегоденьгаминаобратныйпутьисосвоейстороныдаетобещаниевстрети
тьсяснимпередуходомпоезданавокзаленозатемвнейпробуждаетсябольшаянежностькюн
ошеонаготовапожертвоватьвсемчтобытолькосохранитьегодлясебяионарешаетотправит
ьсяснимвместевпутешествиевместотогочтобыснимпроститьсявсяческиепомехизадержи
ваютееионаопаздываетнапоездвтоскепоисчезнувшемуюношеонасноваприходитвигорны
йдомисвозмущениемобнаруживаеттамтежерукинакануневозбудившиевнейтакуюгорячу
юсимпатиюнарушительдолгавернулсякигреонанапоминаетемуобегообещаниииноодержи
мыйстрастьюонбранитсорвавшуюегоигрувелитейубиратьсявонишвыряетденьгикоторым
ионахотелаеговыкупитьопозореннаяонапокидаетгородавпоследствииузнаетчтоейнеуда
лосьспастиегоотсамоубийстваэтаблестящеибезпробеловвмотивировкенаписаннаяновел
лаимеетконечноправонасуществованиекактаковаяинеможетнепроизвестиначитателябо
льшоговпечатленияоднакопсихоанализучитчтоонавозникланаосновеумопострояемогово
жделенияпериодаполовогосозреванияокаковомвожделениинекоторыевспоминаютсове
ршенносознательносогласноумопострояемомувожделениюматьдолжнасамаввестиюнош
увполовуюжизньдляспасенияегоотзаслуживающегоопасениявредаонанизмастольчасты
есублимирующиехудожественныепроизведениявытекаютизтогожепервоисточникапоро
конанизмазамещаетсяпорокомигорнойстрастиударениепоставленноенастрастнуюдеяте
льностьрукпредательскисвидетельствуетобэтомотводеэнергиидействительноигорнаяо
держимостьявляетсяэквивалентомстаройпотребностивонанизмениоднимсловомкромес
ловаигранельзяназватьетееаа

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.