КРИПТОГРАФІЯ

КОМП’ЮТЕРНИЙ ПРАКТИКУМ №2

Експериментальна оцінка ентропії на символ джерела відкритого тексту

ФБ-13 Владислав Садохін та Данило Розумовський

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.

Код:

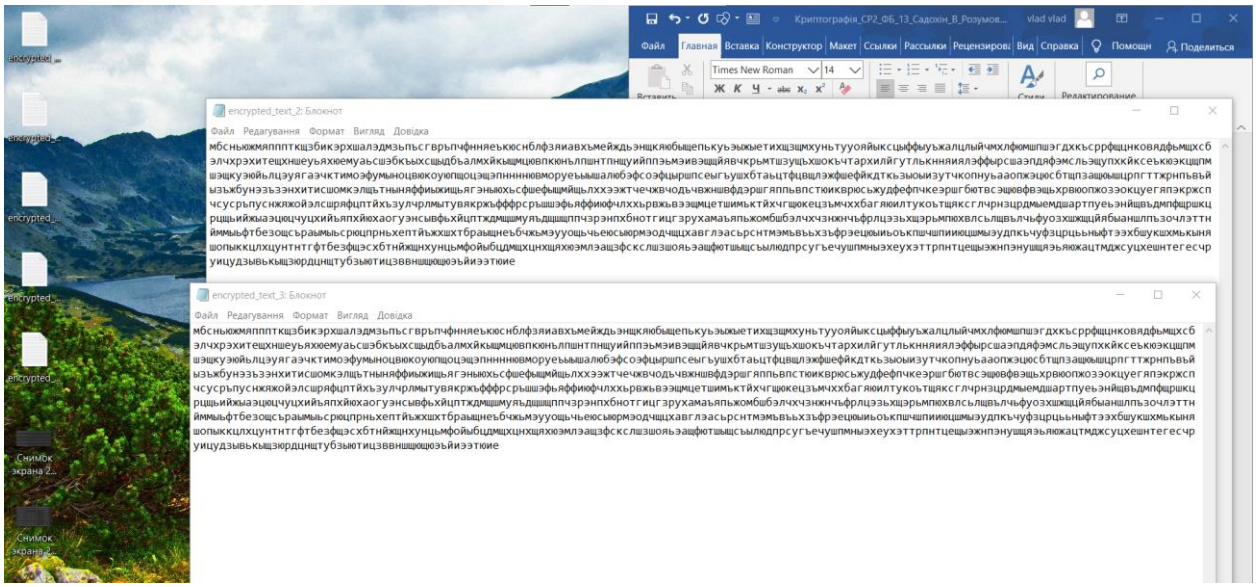
```
import random
import matplotlib.pyplot as plt
from tabulate import tabulate

def vigenere_encrypt(plaintext, key):
    encrypted_text = ""
    key_length = len(key)
    alphabet = 'абвгдежзийклмнопрстуфхцчшщъыьэюя'
    for i in range(len(plaintext)):
        char = plaintext[i]
        if char.isalpha():
            char = char.replace('ё', 'е')
            shift = alphabet.index(key[i % key_length].lower())
            if char.islower():
                encrypted_char = alphabet[(alphabet.index(char) + shift) %
len(alphabet)]
            else:
                encrypted_char = alphabet[(alphabet.index(char.lower()) +
shift) % len(alphabet)].upper()
            encrypted_text += encrypted_char
        else:
            encrypted_text += char
    return encrypted_text

def generate_random_key(length, alphabet):
    key = ''.join(random.choice(alphabet) for _ in range(length))
    return key

def calculate_ioc(text, alphabet):
    total_chars = len(text)
    ioc = 0
    for char in alphabet:
        char_count = text.count(char)
        ioc += (char_count * (char_count - 1)) / (total_chars * (total_chars
```

```
- 1))
    return ioc

def plot_histogram(indices):
    periods = list(indices.keys())
    ic_values = list(indices.values())

    plt.bar(periods, ic_values)
    plt.xlabel('Період')
    plt.ylabel('Індекс відповідності')
    plt.title('Графік індексів відповідності')
    plt.show()

# Завантаження тексту з файлу
with open(r'D:\Криптографія\Lab2\text.txt', 'r', encoding='utf-8') as file:
    plaintext = file.read()

# Російський алфавіт
alphabet = 'абвгдежзийклмнопрстуфхцчшщъыьэюя'

# Довжини ключів
key_lengths = [2, 3, 4, 5, 15]

# Таблиця для зберігання результатів
table = []
indices = {}
# Обчислення індексів відповідності для відкритого тексту
plaintext_ioc = calculate_ioc(plaintext, alphabet)
table.append(['Відкритий текст', plaintext_ioc])

# Зашифрування тексту з кожним ключем та обчислення індексів відповідності

for i, key_length in enumerate(key_lengths):
    key = generate_random_key(key_length, alphabet)
    encrypted_text = vigenere_encrypt(plaintext, key)
    encrypted_ioc = calculate_ioc(encrypted_text, alphabet)
    indices[key_length] = encrypted_ioc
    table.append([f'Зашифрований текст (довжина ключа {key_length})',
encrypted_ioc])
    # Виведення зашифрованого тексту для поточного ключа
    print(f'Зашифрований текст з ключем довжиною
{key_length}:\n{encrypted_text}\n')

# Виведення результатів у вигляді таблиці
print(tabulate(table, headers=['Текст', 'Індекс відповідності'],
tablefmt='fancy_grid'))
# Графік індексів відповідності
plot_histogram(indices)
# Збереження зашифрованого тексту у файл
output_directory = r'D:\Криптографія\Lab2\\'  # Вказати бажаний шлях до папки
for i, key_length in enumerate(key_lengths):
    with open(f'{output_directory}encrypted_text_{key_length}.txt', 'w',
encoding='utf-8') as file:
        file.write(encrypted_text)
```

Результати:

```
C:\Users\alexd\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:\Users\alexd\PycharmProjects\pythonProject1\proba.py
Зашифрований текст з ключем довжиною 2:
жцмнтшъктчгшжчоетзжчжнтчпхйдрхйпмгшконкьбцутпипрйыбмбцмшзнгеяюбдтчбчэзткпжънннйпцууатйкннваубругцжсжцутпчжфмурхйтачпорутржкжф
Ключ для цього тексту: бе

Зашифрований текст з ключем довжиною 3:
нъчръябоэъляныщиъонысръюцщфзшьручжасхсхяйэъцъличчрямпйэуьтрлмжвмзъюиыикъсцкерхрцъюцищрошркзъеыцлэнхсщыщцысчфъчщфхиюцтыцъчносч
Ключ для цього тексту: ийм

Зашифрований текст з ключем довжиною 4:
чйфиглвегклучкцагъотчаътаисябискэцаеяатчтйынаычлъойзтйфушаларсйягкйтоъъеащвиюьчсджинъэхиууыьбжлсчдосдечтчзфобиснскчйбжълчэоп
Ключ для цього тексту: тшйа

Зашифрований текст з ключем довжиною 5:
гпэжхсччпцасчрсюпфгцгжгртожснфжизыыглъзыюпдлтбмэжъюетппсдъадьчтэхрюдъжпгаяэжкцмхрмслмгкъяяръбмжпгюгхрларйнйанфжлсртзнаппггчн
Ключ для цього тексту: ююткд

Зашифрований текст з ключем довжиною 15:
мбсныюжмяпппткщзбикэрхшалэдмзьпъсгвръпчфнняеъкюснблфзяиавхъмейждьэнщкяюбыщепькуьэыжыетихщзщмхуньтууояйыксцыффыуъжалцлыйчмхлф
Ключ для цього тексту: зржеллнзоэньншм
```

Як видно код зашифрував текст ,котрий ми обрали з 1 СР з Робінзона Крузо за допомогою випадково згенерованих ключів різних довжин.



Окрім того, можемо побачити,що код створив окремі файли для кожного випадку.

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

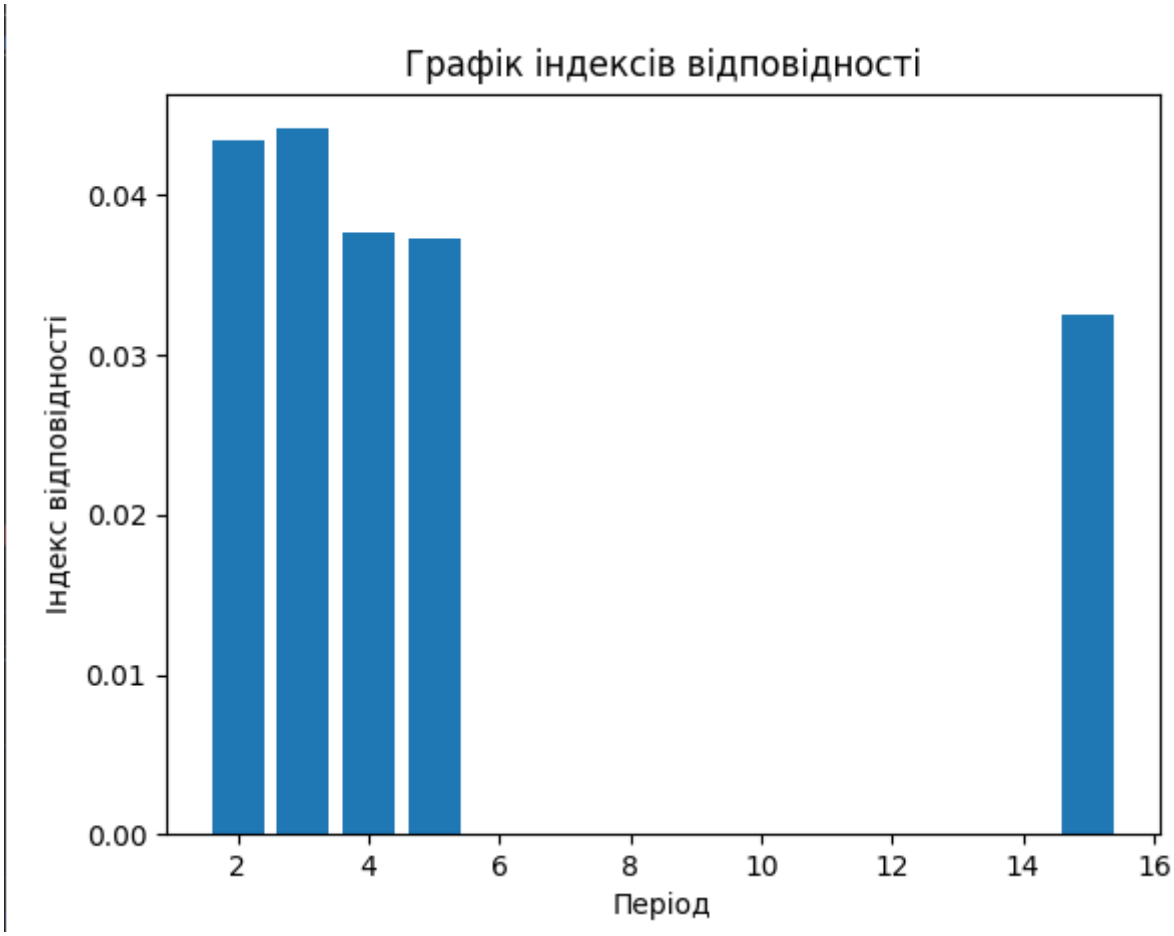
Код той самий що і в першому завданні

Результати:

Текст	Індекс відповідності
Відкритий текст	0.0568629
Зашифрований текст (довжина ключа 2)	0.0449282
Зашифрований текст (довжина ключа 3)	0.0438221
Зашифрований текст (довжина ключа 4)	0.0373629
Зашифрований текст (довжина ключа 5)	0.0414313
Зашифрований текст (довжина ключа 15)	0.033111

Бачимо табличку з порахованими значеннями індексів відповідності для відкритого тексту та кожного випадку окремо.

Графік:



3. Використовуючи наведені теоретичні відомості, розшифрувати наданий шифртекст (згідно свого номеру варіанта №8) .

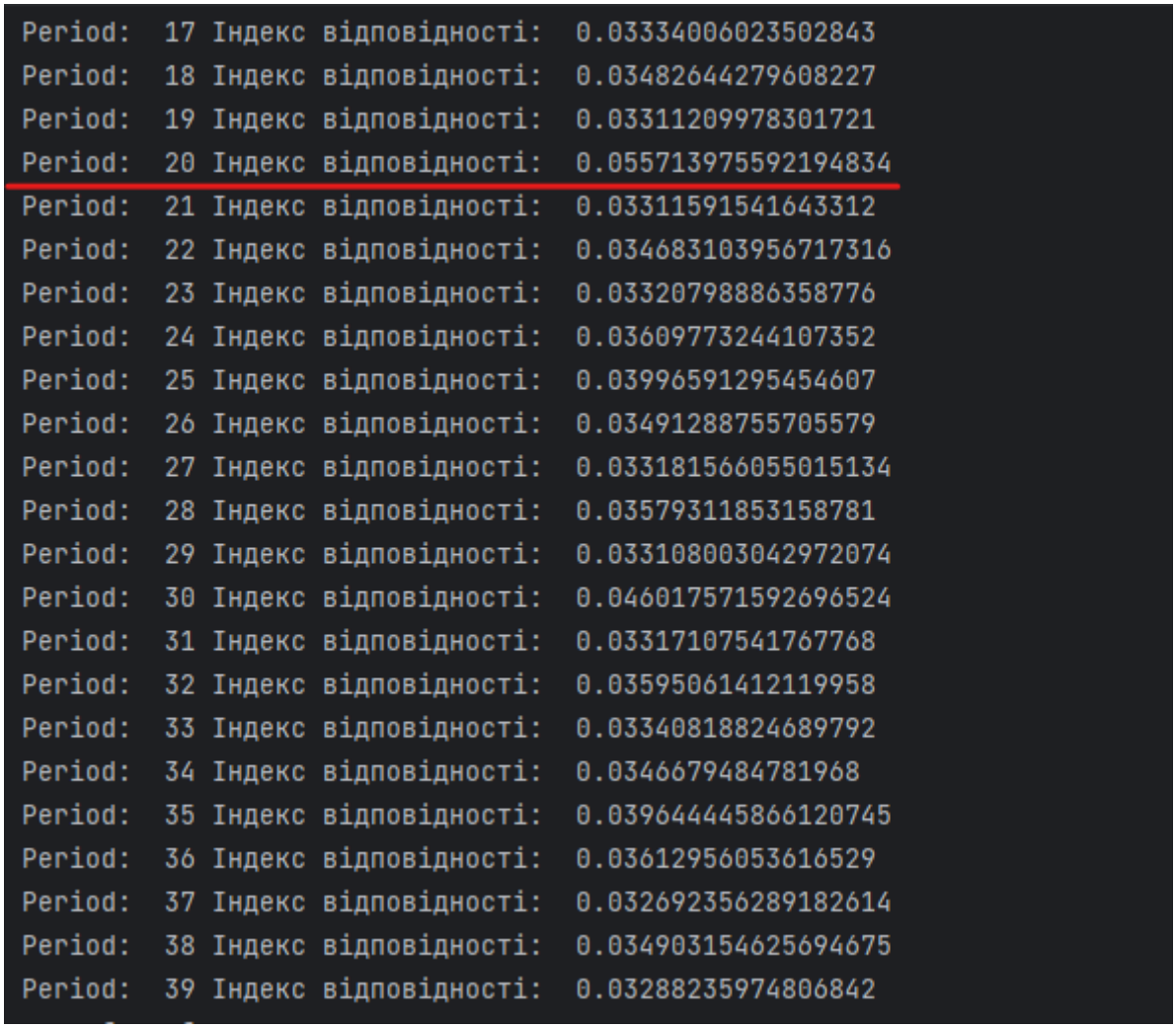
Код:

```
import matplotlib.pyplot as plt
from collections import Counter
alphabet = "абвгдежзийклмнопрстуфхцчшщъыьэюя"


def calculate_ic(text):
    # Підраховуємо кількість кожної літери у тексті
    letter_count = Counter(text)
    ic = 0.0
    total_letter_count = sum(letter_count.values())
    for count in letter_count.values():
        ic += (count * (count - 1)) / (total_letter_count *
(total_letter_count - 1))
    return ic


# Функція для розбиття тексту на блоки та обчислення індексу відповідності
def find_and_print_ic(filename):
    with open(filename, 'r', encoding='utf-8') as file:
        ciphertext = file.read()  # Зчитуємо шифртекст
    indices = {}
    for r in range(2, 40):
        # Розбиваємо шифртекст на блоки довжиною r
        blocks = []
        for i in range(r):
            block = ciphertext[i::r]
            blocks.append(block)

        block_ic_values = [calculate_ic(block) for block in blocks]
        average_ic = sum(block_ic_values) / r
        indices[r] = average_ic
        print("Period: ", r, "Індекс відповідності: ", average_ic)

    return indices


def plot_histogram(indices):
    periods = list(indices.keys())
    ic_values = list(indices.values())

    plt.bar(periods, ic_values)
    plt.xlabel('Період')
    plt.ylabel('Індекс відповідності')
    plt.title('Графік індексів відповідності')
    plt.show()


def find_key(filename, key_length):
    with open(filename, 'r', encoding='utf-8') as file:
        ciphertext = file.read().replace(" ", "").lower()  # Зчитуємо
шифртекст, прибираємо пропуски та переводимо в нижній регістр
    alphabet = "абвгдежзийклмнопрстуфхцчшщъыьэюя"
    key = ""
    for i in range(key_length):
        block = ciphertext[i::key_length]
        most_common_letter = max(alphabet, key=lambda letter:
```

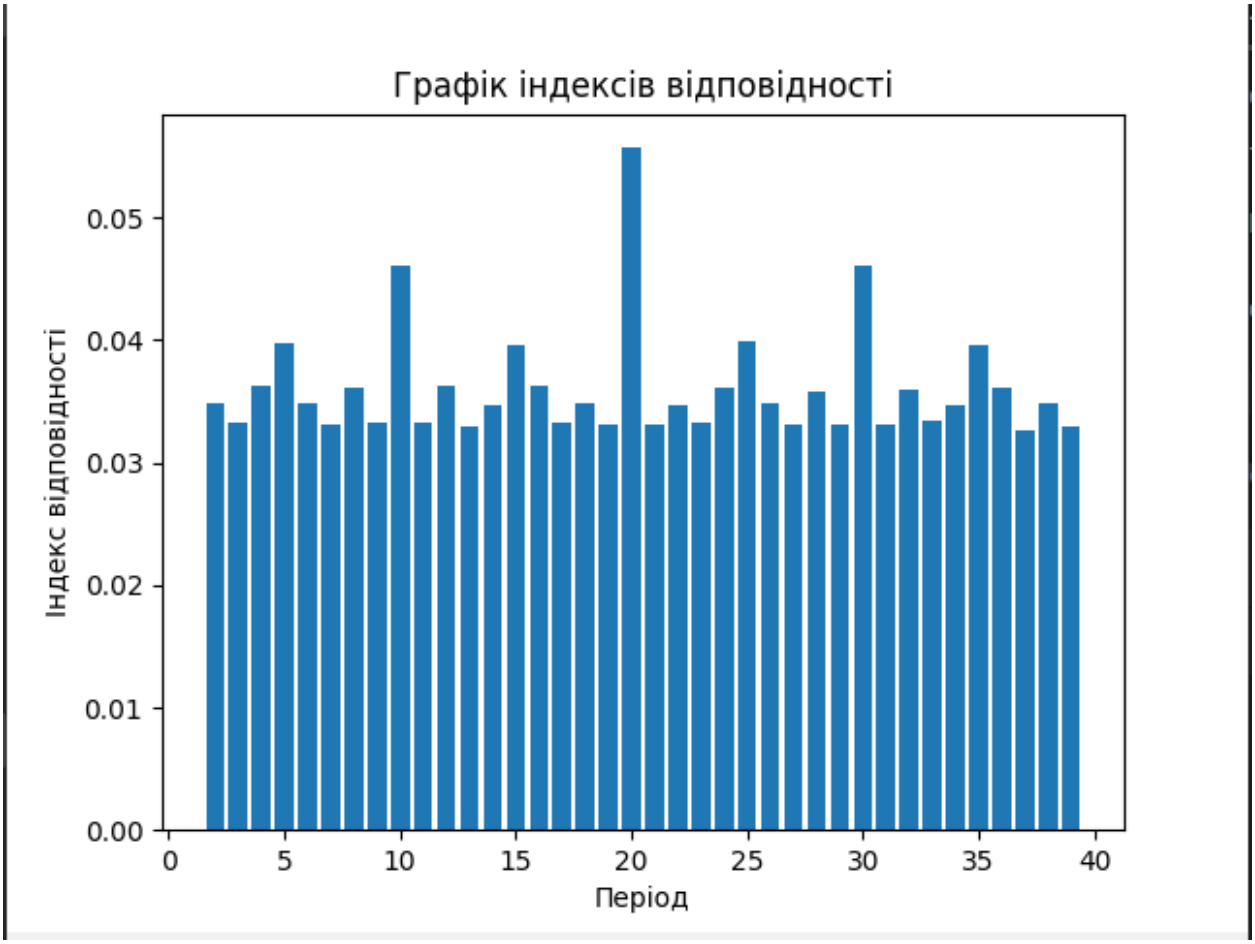
```
block.count(letter))
        shift = (alphabet.index(most_common_letter) - alphabet.index('o')) %
len(alphabet)
        key += alphabet[shift]
    print(key)
    return key


def decrypt_vigenere(filename, key):
    with open(filename, 'r', encoding='utf-8') as file:
        ciphertext = file.read().replace(" ", "").lower()

    decrypted_text = ""
    for i, letter in enumerate(ciphertext):
        shift = (alphabet.index(key[i % 20]))

        decrypted_index = (alphabet.index(letter) - shift) % len(alphabet)
        decrypted_letter = alphabet[decrypted_index]
        decrypted_text += decrypted_letter

    return decrypted_text


# Основний код
cipher_file = "D:\\Криптографія\\Lab2\\textvar8.txt"

indices = find_and_print_ic(cipher_file)
plot_histogram(indices)

find_key(cipher_file, 20)
key = "улановсеребряныепули"
decrypted_text = decrypt_vigenere(cipher_file, key)
with open("decrypted_text.txt", "w", encoding='utf-8') as output_file:
    output_file.write(decrypted_text)
print("Розшифрований текст збережено у файлі decrypted_text.txt")
```

Результати:

Як бачимо до теоретичного значення для російської мови I=0,0553 найбільш наближений період 20 з індексом відповідності 0,0557, отже можна припустити, що довжина ключа буде 20



Графік:



Далі вказавши у функції find_key довжину ключа 20, функція вивела нам ключ ‘уланобсеребзяныепуля’.Такий ключ не дуже схожий на щось змістовне та логічне,як було сказано в теоретичних відомостях => його треба трохи виправити і тоді вийде ключ – ‘улановсеребряныепули’ ,що являється книгою ‘**Серебряные пули** с урановым сердечником’ - Андрея **Уланова.**

```
уланобсеребзяныепуля
Розшифрований текст збережено у файлі decrypted_text.txt
```

Далі у функцію decrypt_vigenere вказуємо як аргумент цей ключ і функція розшифровує текст та зберігає у файлі decrypted_text.txt

```python
# Основний код
cipher_file = r'C:\Users\alexd\Desktop\textvar8.txt'
find_vigenere_key_length(cipher_file)

find_key(cipher_file, 20)
key = "улановсеребряныепули"
decrypted_text = decrypt_vigenere(cipher_file, key)
with open("decrypted_text.txt", "w", encoding='utf-8') as output_file:
    output_file.write(decrypted_text)
print("Розшифрований текст збережено у файлі decrypted_text.txt")
```

Частина розшифрованого тексту:

этасистемакрасногокарликаникогданеимеланазваниятолькозубодробительнодлинныйномервкаталогеисследовавшийеекиберзондотметилналичиетрехгазовыхгигантовдвухастероидныхполейкометногооблакаизанесвсеэтиданныевсекторвторойочередипомнениюинкакиберзондасистеманепредставляланикакойценностидляпослявшихеголюдейнаверноебудьунегозадействованыконтурывторогоуровнясамостоятельностииазартаонбыпоспорилсамссобойчтовближайшуютысячулетлюдиздесьнепоявятсяипроспорилбылюдипоявилисьвэтойсистеменечерезтысячулетавсеголишьчерезсемьэтобылинетелюдичтопосылализондформальноонивообщенедолжныбылизнатьосуществованииэтойсистемыноутехктоихпосылал

Висновки:

У ході виконання виконання лабораторної роботи ми дізналися що таке шифр Віженера, розібралися як він працює, як зашифровувати та розшифровувати текст за допомогою цього шифру, що таке та навіщо потрібен індекс відповідності, як його знаходити та як його можна використати щоб знайти можливий період ключа шифру Віженера