НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

КРИПТОГРАФІЯ

**КОМП’ЮТЕРНИЙ ПРАКТИКУМ №2**

Криптоаналіз шифру Віженера

Виконали:

студентки групи ФБ-23

Гуз Вікторія

Шукалович Марія

**Мета роботи:** засвоєння методів частотного криптоаналізу. Здобуття навичок роботи та аналізу потокових шифрів гамування адитивного типу на прикладі шифру Віженера.

**Постановка задачі**

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.

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

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

**Хід роботи**

Для початку ми очистили текст від всіх символи, окрім російського алфавіту (наприклад, цифри, розділові знаки, латиниця тощо).  
def load\_and\_clean\_text(file\_path):

    try:

        with open(file\_path, 'r', encoding='utf-8') as file:

            text = file.read()

        text = text.lower()

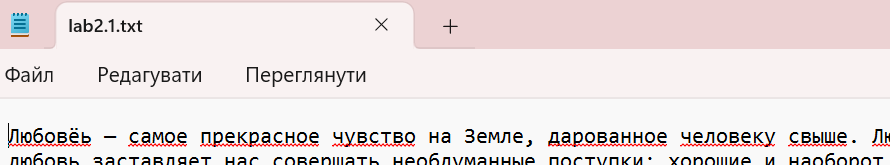
        text = re.sub(r'[^а-яё]', '', text)

        return text

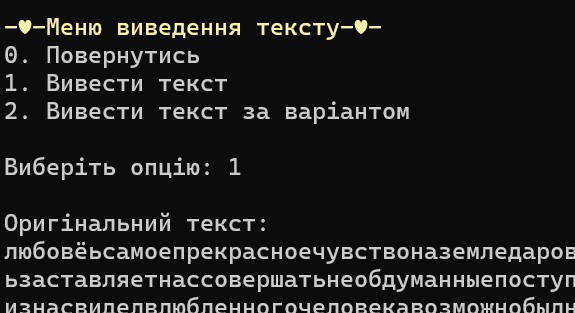
    except FileNotFoundError:

        print(f"Файл '{file\_path}' не знайдено")

        return None

Вхідний текст:  


Очищений текст:



З цим етапом труднощів не виникало.

Після цього ми обрали ключі довжини r = 2, 3, 4, 5, а також довжини 10-20 знаків:

encryption\_keys = {'2': 'хи', '3': 'мяу', '4': 'сова', '5': 'осень',  '10': 'абитуриент', '11': 'бессмертный', '12': 'концентратор','13': 'глазированный', '14': 'мультипликатив', '15': 'ухлёстывавшийся', '16': 'христадельфианин', '17': 'яфетидологический', '18': 'задокументировавши', '19': 'жизнеобеспечивающий', '20': 'евростандартизировав'}

Наступним етапом була реалізація шифрування тексту за допомогою шифру Віженера. Функція vigenere приймає два параметри: plaintext (вхідний текст для шифрування) і key (ключ для шифрування). Вона використовує алфавіт, що містить 33 символи і проходить по кожному символу вхідного тексту, якщо символ є маленькою літерою, vigenere застосовує зсув згідно з відповідною літерою в ключі. Зсув обчислюється як індекс літери ключа в алфавіті. Потім для кожної букви вхідного тексту відбувається зсув у алфавіті на кількість, що відповідає індексу відповідної літери ключа. Зашифрований текст формується шляхом поєднання зашифрованих літер у новий рядок

def vigenere(plaintext, key):

    cipher\_text = []

    key\_length = len(key)

    alphabet = 'абвгдеёжзийклмнопрстуфхцчшщъыьэюя'

    for i, char in enumerate(plaintext):

        if char.islower():

            key\_char = key[i % key\_length].lower()

            shift = alphabet.index(key\_char)  # Отримаємо зсув за ключем

            encrypted\_char = alphabet[(alphabet.index(char) + shift) % 33]

            cipher\_text.append(encrypted\_char)

    return ''.join(cipher\_text)

На цьому етапі не виникло значних труднощів. Ініціалізація змінних, обробка кожного символу, визначення зсуву за допомогою ключа та обчислення зашифрованих символів пройшли без проблем. Алгоритм шифрування Віженера виявився простим для реалізації, і всі операції, такі як перевірка на малі літери та обчислення зсуву в алфавіті, були виконані без помилок.

|  |  |
| --- | --- |
| Довжина ключа | Індекс відповідності |
| 2 | 0.04316476921719178 |
| 3 | 0.037057629511958345 |
| 4 | 0.036958029174073655 |
| 5 | 0.03626713062773428 |
| 10 | 0.03410996381607978 |
| 11 | 0.03334215867972818 |
| 12 | 0.034548709608280694 |
| 13 | 0.03349471109598195 |
| 14 |  |
| 15 |  |
| 16 |  |
| 17 |  |
| 18 |  |
| 19 |  |
| 20 |  |