МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

ЛАБОРАТОРНА РОБОТА №3

З дисципліни «Криптографія»

Варіант 1

Виконали:
студенти 3 курсу ФТІ
групи ФБ-93
Абдуллаєва Есміра
Шовак Мирослав

Викладач:
Селюх П. В.

м.Київ – 2021

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

Завдання:

1. Уважно прочитати методичні вказівки до виконання комп'ютерного практикуму.
2. 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.
3. За допомогою програми обчислення частот біграм, яка написана в ході виконання комп'ютерного практикуму No1, знайти 5 найчастіших біграм запропонованого шифртексту (за варіантом).
4. 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).
5. Для кожного кандидата на ключ дешифрувати шифртекст. Якщо шифртекст не є змістовним текстом російською мовою, відкинути цього кандидата.
6. Повторювати дії 3-4 доти, доки дешифрований текст не буде змістовним.

Хід роботи

Першим кроком у нашій лабораторній роботі було те , що ми розбили текст на біграми без перетину. Після цього ми підрахували частоту кожної біграми і виділили 5 найчастіших. Наступним кроком було те , що для розв'язання системи лінійних рівнянь і в подальшому знаходженні ключів, нам потрібно було перебрати комбінації біграм мови і біграм зашифрованого тексту . Ми вирішили автоматизувати цей процес і створили цикл , який буде вирізувати СЛР і створювати масив можливих ключів. Після цього ми використовували цей масив для розшифрування тексту. Змістовність тексту ми визначали за допомогою аналізатора , який перевіряв частоти букв та індекс відповідності. В результаті ми знайшли нашу пару біграм і відповідно наш ключ , але розшифрований текст був дещо з помилками. Так як у методичці було написано , що можливо букву 'ь' змінювали на 'ы' , ми вирішили їх поміняти у нашому масиві літер, після чого результат розшифрування став правильним.

Код програми

```
from itertools import permutations

file1 =
open("/Users/esmira.23/Desktop/KPI/3курс/Крипт
а/1.txt", "r").read()
file2 =
open("/Users/esmira.23/Desktop/KPI/3курс/Крипт
а/2.txt", "w")
file3 =
open("/Users/esmira.23/Desktop/KPI/3курс/Крипт
а/result.txt", "r").read()

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']

# вихідный текст
bigram1 = list(permutations([['с', 'т'], ['н',
'о'], ['т', 'о'], ['н', 'а'], ['е', 'н']], 2))
# вхідний текст
bigram2 = list(permutations([['р', 'н'], ['ы',
'ч'], ['н', 'к'], ['ц', 'з'], ['и', 'а']], 2))


# bigram1 = [['с', 'т'], ['е', 'н']]
# bigram2 = [['р', 'н'], ['н', 'к']]


def euclid_ext(a, n):
    if n == 0:
        return a, 1, 0
    else:
        d, x, y = euclid_ext(n, a % n)
        return d, y, x - y * (a // n)


def reverse(a, n):
    gcd, x, y = euclid_ext(a, n)
    if gcd == 1:
        return (x % n + n) % n
    else:
        return -1


def euclid(a, y, n):
    gcd, y1, x1 = euclid_ext(a, n)
    if gcd == 1:  # знаходимо обернений
        x = reverse(a, n)
        return x
    elif y % gcd != 0:  # немає розв`язкі
        return False
    else:
        euclid(a / gcd, y / gcd, n / gcd)


def max_bigram(text):
    mass = []
    mass1 = []
    line = [text[k:k + 2] for k in range(0,
len(text), 2)]
    new_line = set(line)
    for i in new_line:
        number = line.count(i)
        mass.append([i, number])
    sorted_bigrams = sorted(mass, key=lambda
x: x[1])
    for i in range(5):
        mass1.append(sorted_bigrams[-(i + 1)])
        mass.clear()
    for i in range(len(mass1)):
        mass.append(mass1[i][0])
    print(mass)


def compliance_index(text):
    arr = []
    for i in alphabet:
        letter = text.count(i)
        arr.append(letter * (letter - 1))
    I = sum(arr) / (len(text) * (len(text) -
1))
    return I


def index():
    mass = []
    arr = []
    for i in range(len(bigram1)):
        for j in range(len(bigram2)):
            mass.append(bigram1[i] +
bigram2[j])
    for j in mass:
        X1 = alphabet.index(j[0][0]) * 31 +
alphabet.index(j[0][1])
        Y1 = alphabet.index(j[2][0]) * 31 +
alphabet.index(j[2][1])
        X2 = alphabet.index(j[1][0]) * 31 +
alphabet.index(j[1][1])
        Y2 = alphabet.index(j[3][0]) * 31 +
alphabet.index(j[3][1])
        arr.append([X1, Y1, X2, Y2])
    return arr


def find_key():
    mass = []
    XY = index()
    for i in range(len(XY)):
        X1 = XY[i][0]
        Y1 = XY[i][1]
        X2 = XY[i][2]
        Y2 = XY[i][3]
        a = (euclid(X1 - X2, Y1 - Y2, 31 ** 2)
* (Y1 - Y2)) % (31 ** 2)
        b = (Y1 - a * X1) % (31 ** 2)
        mass.append([a, b])
    return mass


def decrypt(text):
    arr = []
    arr1 = []
    AB = find_key()
    line = [text[k:k + 2] for k in range(0,
len(text), 2)]
    for j in range(len(AB)):
        for i in range(len(line)):
            A = AB[j][0]
            B = AB[j][1]
            Y = alphabet.index(line[i][0]) *
31 + alphabet.index(line[i][1])
            X = (reverse(A, 31 ** 2) * (Y -
B)) % 31 ** 2
            arr.append(X)
        for i in range(len(arr)):
            letter = alphabet[arr[i] // 31] +
alphabet[arr[i] % 31]
            arr1.append(letter)
        answer = ''.join(arr1)
        arr1.clear()
        arr.clear()
        # аналізатор російської мови
        if compliance_index(answer) <=
compliance_index(file3) and (
                answer.count('о') /
len(answer) < 0.11 or answer.count('а') /
len(answer) < 0.06):
            continue
        else:
            print(answer)
            return file2.write(answer)


# main
decrypt(file1)
```

Результат роботи

```
1. [['рн', 62], ['ыч', 41], ['нк', 34], ['цз', 32], ['иа', 30]]
2. ['рн', 'ыч', 'нк', 'цз', 'иа']
3. X1: 545 Y1: 509
   X2: 168 Y2: 413
4. A: 13    B: 151
```

1. Знайшли у тексті 5 найчастіших біграм та вивели їх кількість.
2. Залишили лише самі біграми.
3. Знайдені X та Y, які підійшли нам для знаходження ключів A та B.
4. Ключі A та B, за допомогою яких було розшифровано текст.

Біграми, які допомогли знайти ключі:

```
Bigram plaintext:  ['с', 'т'] ['е', 'н']
Bigram ciphertext:  ['р', 'н'] ['н', 'к']
```