МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра кібербезпеки

**Дисципліна “Системи та засоби криптоаналізу”**

**Звіт до лабораторної роботи № 4**

на тему “ Дослідження криптоаналітичних атак на афінний шифр. Частина 2**”**

Студент Борщ Д.О.

Варіант № 1

Група КБ-01

Перевірила Лаврик Т.В

**Суми 2022**

**ЗВІТ**

Написати програмний код методу частотного аналізу для:

1. (10 б.) визначення ключа, яким зашифровано текст.

**import** **sys**

**import** **math**

**import** **json**

**import** **random**

alphabet = [

'а', 'б', 'в', 'г', 'ґ', 'д', 'е', 'є', 'ж', 'з', 'и', 'і', 'ї', 'й',

'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч',

'ш', 'щ', 'ь', 'ю', 'я', ' '

]

**def** encrypt(string, keyA, keyB):

output = ""

**for** i **in** string:

index = (keyA\*alphabet.index(i) + keyB) % len(alphabet)

output += alphabet[index]

**return** output

**def** decrypt(string, keyA, keyB):

output = ""

**for** i **in** string:

index = pow(keyA, -1, len(alphabet))\*(alphabet.index(i) - keyB) % len(alphabet)

output += alphabet[index]

**return** output

**def** hack(string):

lettersFreq = {}

**for** letter **in** string:

**try**: lettersFreq[letter] += 1

**except**: lettersFreq[letter] = 1

**print**(f"Number of each letter: {json.dumps(lettersFreq, indent=1, ensure\_ascii=False)}")

*# Finding max values*

max1 = max(lettersFreq, key=lettersFreq.get)

max2 = max1

**while** max2 == max1: max2 = random.choice(list(lettersFreq))

**for** letter **in** lettersFreq:

**if** (

lettersFreq[letter] > lettersFreq[max2]

**and** lettersFreq[letter] < lettersFreq[max1]

): max2 = letter

Yletters = (max1, max2)

**print**(f"Most frequent letters: {Yletters}")

*#creating key-value pairs*

XY1 = (' ', Yletters[0])

XY2 = ('о', Yletters[1])

pair1 = (

(alphabet.index(XY1[0]) - alphabet.index(XY2[0])) % len(alphabet),

(alphabet.index(XY1[1]) - alphabet.index(XY2[1])) % len(alphabet)

)

*# Finding A and B*

keyA = (pow(pair1[0], -1, len(alphabet)) \* pair1[1]) % len(alphabet)

keyB = (alphabet.index(XY1[1]) - keyA \* alphabet.index(XY1[0])) % len(alphabet)

**print**(f"Found key A: {keyA}**\n** Found key B: {keyB}")

**print**(f"Trying to decrypt...")

**print**(f"Decrypted text is:**\n**{decrypt(string, keyA, keyB)}")

**def** main(args=sys.argv):

*# Using file as input source*

**if** "--file" **in** args:

**try**:

fileName = args[args.index("--file")+1]

**except**:

**print**("ERROR, no file specified!")

**return** 0

**try**:

f = open(fileName, "r")

inputText = ''.join(f.read())

**except**:

**print**("ERROR, can`t open file!")

**return** 0

*# Using plain text as input source*

**else**:

inputText = input("Enter input text: ")

*# Taking key from args*

**try**:

keyA = int(args[args.index("--key")+1])

keyB = int(args[args.index("--key")+2])

**if** 0 > keyA > len(alphabet) **or** 0 > keyB > len(alphabet) **or** math.gcd(keyA, len(alphabet)) != 1:

**print**("ERROR, wrong key specified!")

**return** 0

**except**:

**if** "--hack" **not** **in** args:

**print**("ERROR, no key specified!")

**return** 0

**else**: **pass**

**print**(f"Input text:**\n**{inputText}**\n**")

*# Choosing an option*

**if** "--encrypt" **in** args:

**print**(f"Encrypted text is:**\n**{encrypt(inputText, keyA, keyB)}")

**if** "--decrypt" **in** args:

**print**(f"Decrypted text is:**\n**{decrypt(inputText, keyA, keyB)}")

**if** "--hack" **in** args:

**print**("Trying to hack chipher...**\n**")

hack(inputText)

**return** 0

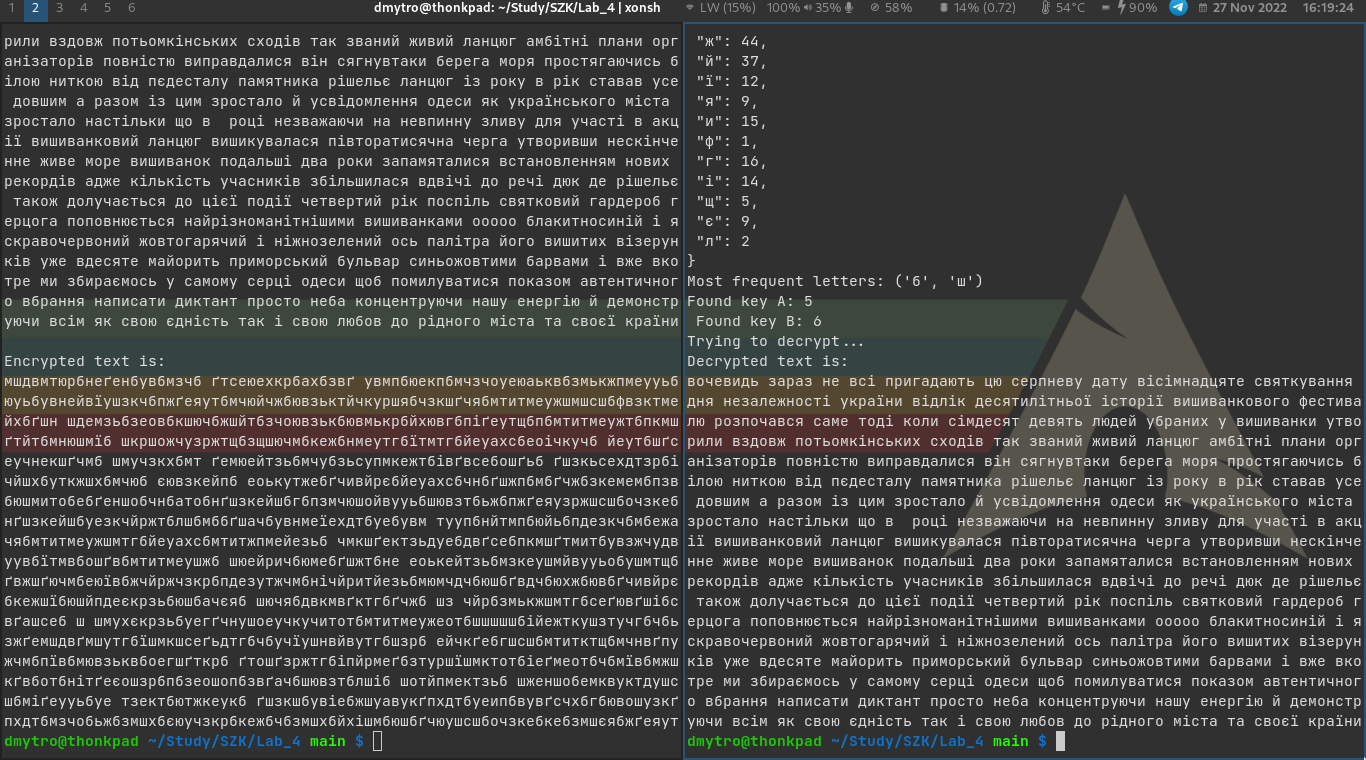
**if** \_\_name\_\_ == "\_\_main\_\_":

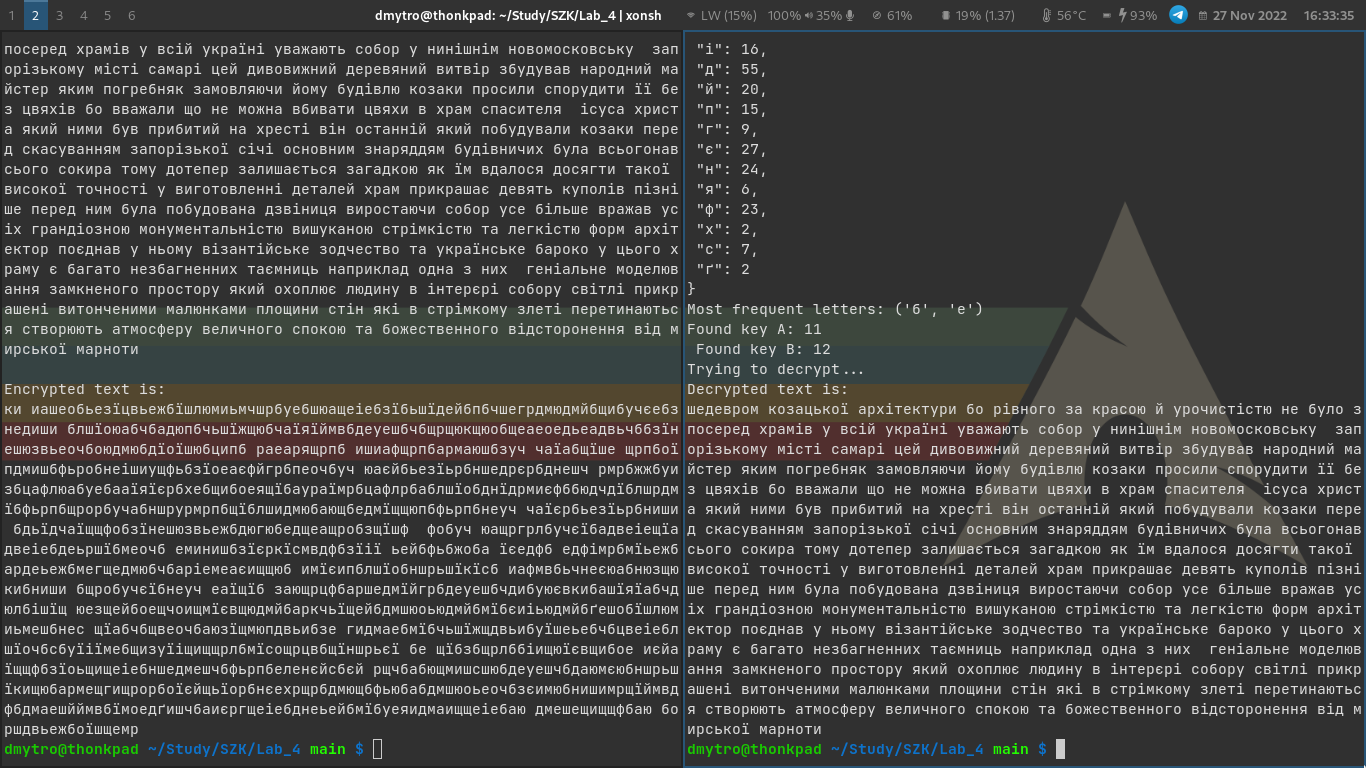
main()

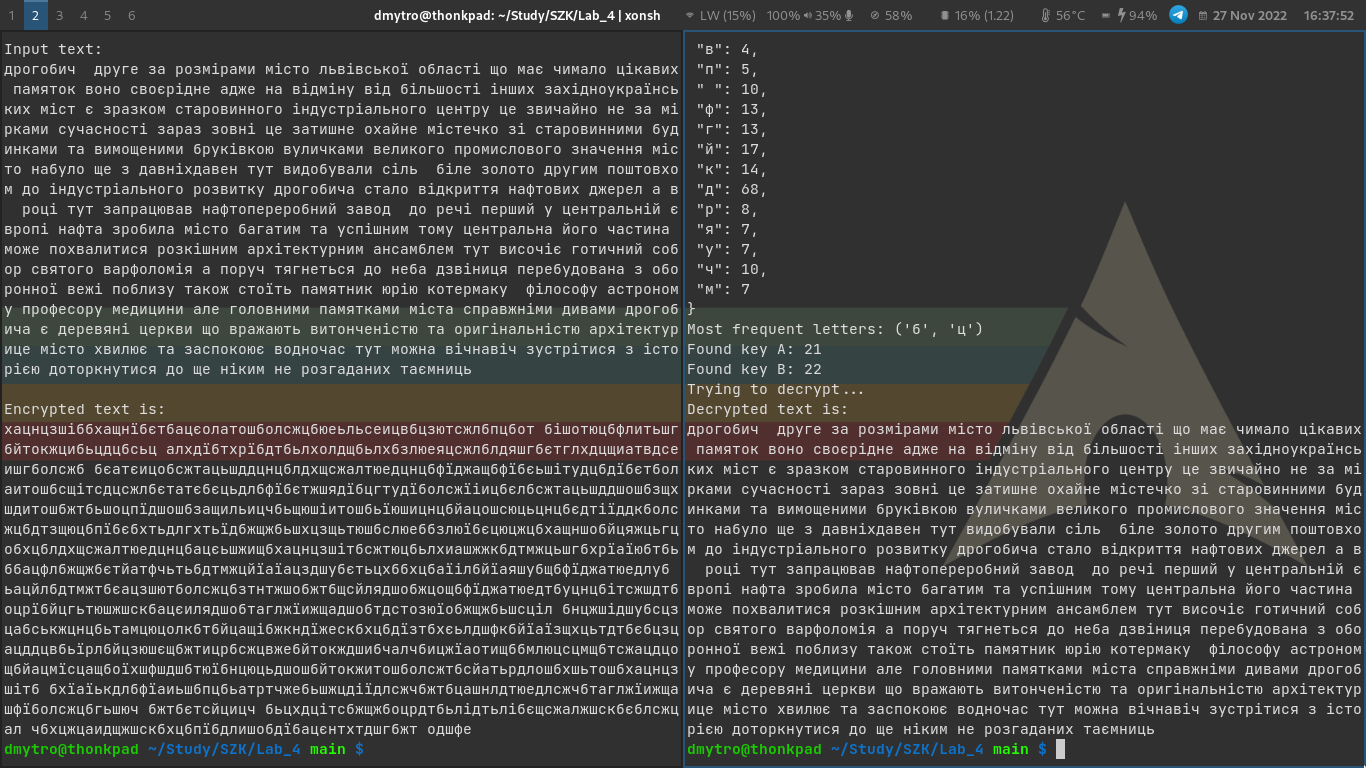
2. (5 б.) розшифрування шифротексту за допомогою визначеного ключа.

Перевірити отримані ключі на правильність.

Скріншоти з результатами роботи програми (мінімум 3 різні шифротексти)

Рис. 1 — Перший приклад з ключами 5 та 6.

Рис. 2 — Другий приклад з ключами 11 та 12

Рис.3 — Третій приклад з ключами 21 та 22.