МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра кібербезпеки

**Дисципліна “Системи та засоби криптоаналізу”**

**Звіт до лабораторної роботи №2**

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

Студент Борщ Дмитро Олександрович

Варіант № 1

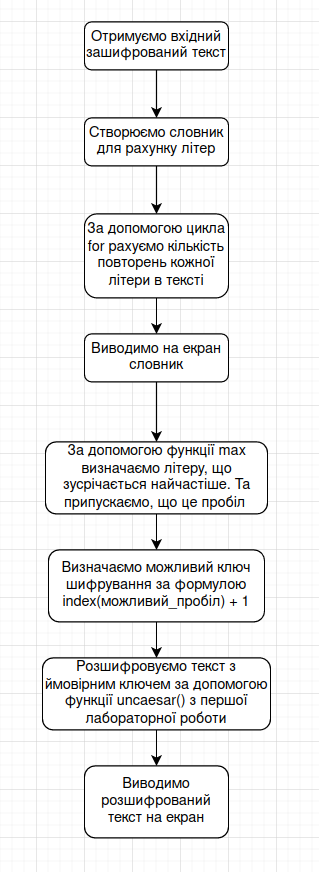
Група КБ-01

Перевірила Лаврик Т.В

**Суми 2022**

**ЗВІТ**

1. (3 б.) Блок-схема алгоритму реалізації методу частотного аналізу:



1. (5 б.) Програмний код методу частотного аналізу.

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 uncaesar(string, key):

output = ""

for i in string:

index = alphabet.index(i)-key

while index < 0: index += len(alphabet)

output += alphabet[index]

return output

def frequencyAnalysis(string):

lettersFreq = {}

for letter in string:

try: lettersFreq[letter] += 1

except: lettersFreq[letter] = 1

print(lettersFreq)

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

key = alphabet.index(possibleSpace)+1

print(

f"Result with key {key}:\n",

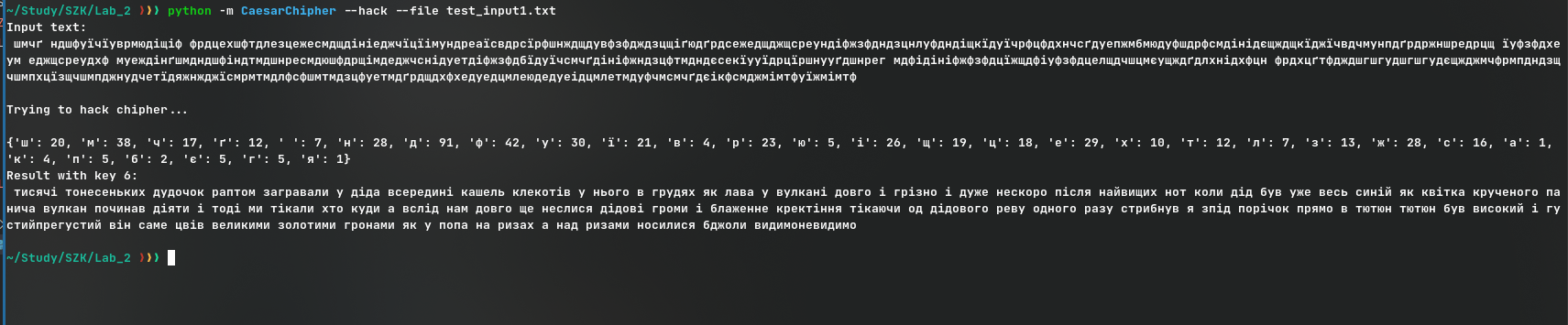
uncaesar(string, key),

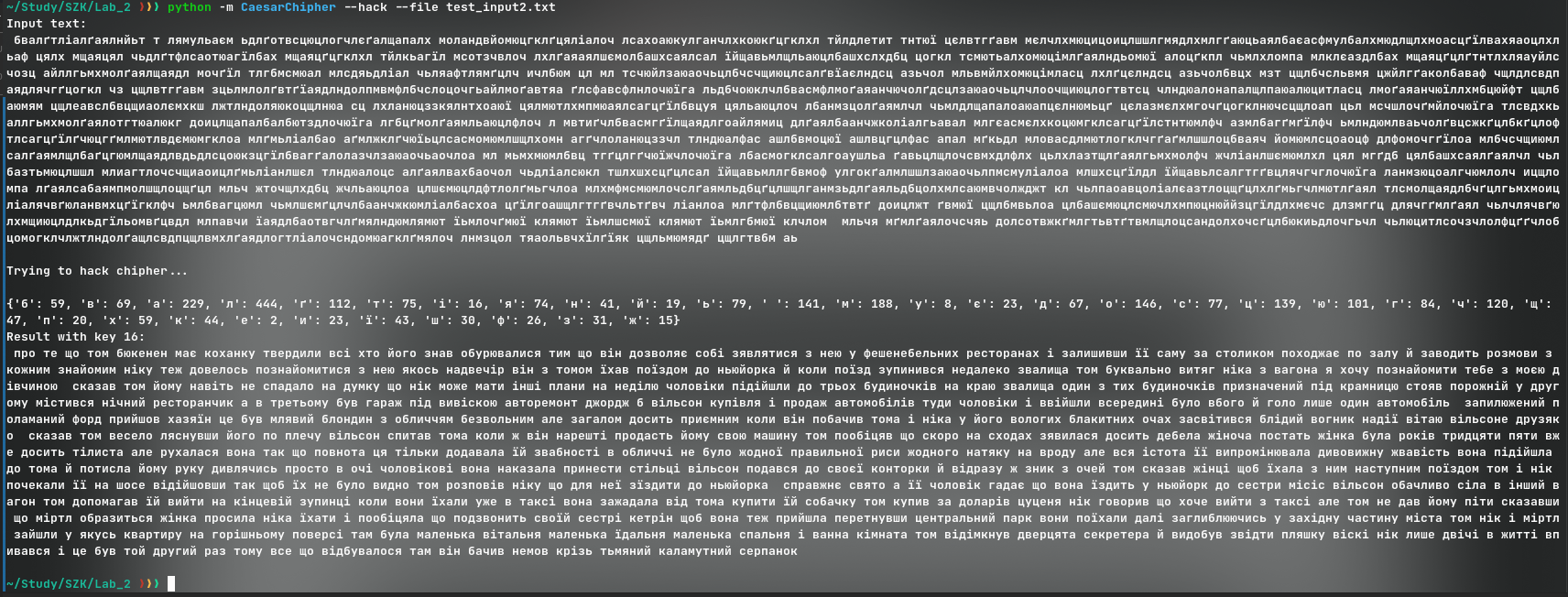
"\n"

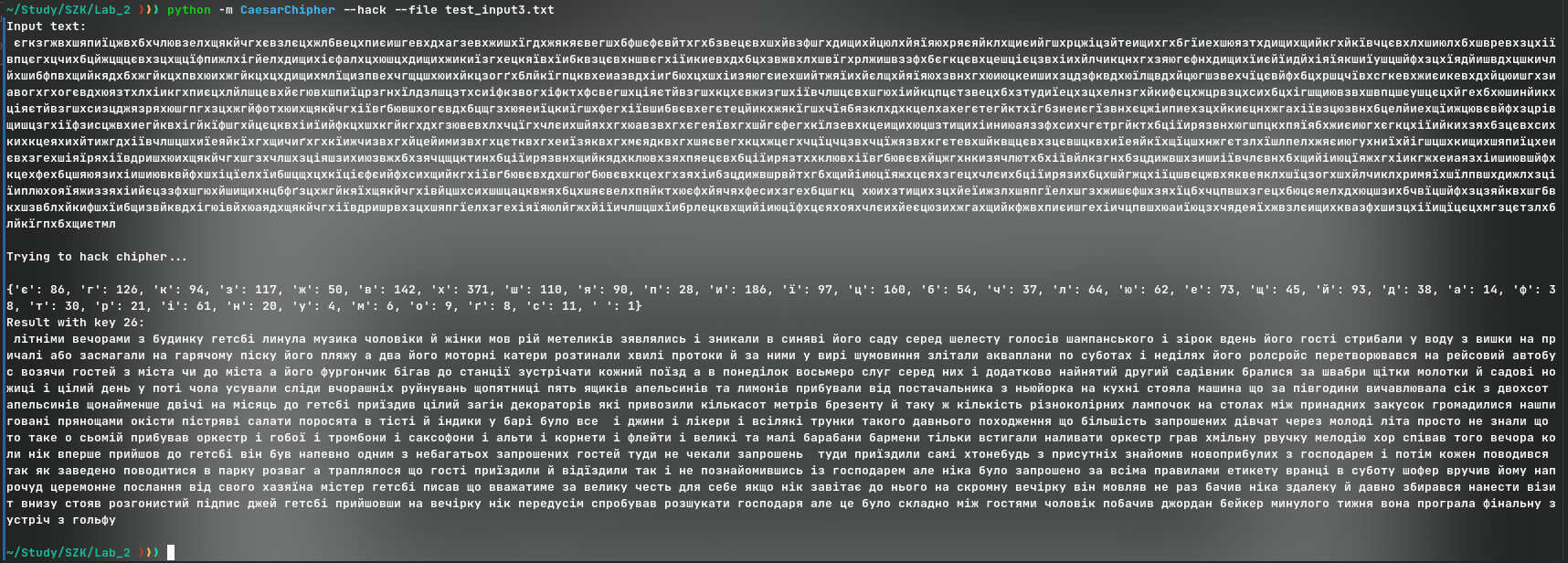
)

Скріншоти з результатами роботи програми (мінімум 3 різні шифротексти)

Для перевірки програми я використовав 3 різні уривки з твору, які попередньо зашифрував з ключами 6, 16 та 26 відповідно.

 Рис. 1 — Перший тест.

 Рис. 2 — Другий тест.

 Рис. 3 — Третій тест.