­­МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Інститут прикладної математики та фундаментальних наук

Кафедра прикладної математики

**ЗВІТ**

про виконання лабораторної роботи № 4  
з дисципліни

***«*АЛГОРИТМИ ЗАХИСТУ ІНФОРМАЦІЇ В ІНФОРМАЦІЙНИХ МЕРЕЖАХ*»***

Виконав:  
студент  
групи ПМ-43  
Павло Гутковський  
Прийняв:   
Ліснічук А. Є.

Львів 2023

­­ **Лабораторна робота №4**

***Тема:***

Шифр модульного гамування: процедури шифрування та дешифрування

***Мета:***

Отримати навики практичної реалізації процедур шифрування та дешифрування для шифру модульного гамування.

***Постановка задачі:***

1. Використовуючи українську абетку запропонувати свій варіант реалізації шифру модульного гамування;

2. Програмно реалізувати процедуру шифрування;

3. запропонувати свій текст про кафедру прикладної математики, довжиною понад 100 літер і записати його у файл (не використовувати пробіли та розділові знаки, а формувати у вигляді ‘КафедраПрикладноїМатематики …’) – це файл відкритого тексту;

4. зашифрувати відкритий текст; як парольну фразу використати слово чи фразу, що означає Ваше хобі (без пробілів, апострофів і т.д.); результат шифрування записати у файл; 5. програмно реалізувати процедуру дешифрування; дешифрувати отриману в попередньому завданні криптограму з використанням тієї самої гами і пересвідчитися, що отримано початковий відкритий текст;

6. підготувати опис/звіт про виконану роботу

***Короткі теоретичні відомості:***

У цьому шифрі використовують гаму, яку будують на основі свого роду парольної фрази. Літерам абетки присвоюють номери від 0 до N −1 , де N – число літер абетки. Шифрування здійснюють використовуючи для кожної літери формулу C M = + ( ) mod Г N , де С – номер літери шифрованого тексту (криптограми), M – номер літери відкритого тексту, Г – номер літери гами, mod N – функція обчислення залишку ділення на N . Якщо відкритий текст є довший від парольної фрази, то при формуванні гами парольну фразу програмно немов би «продовжуємо» необхідне число раз (фактично не записуємо багаторазово парольну фразу, а натомість обчислюємо кожен раз номер відповідної літери гами).

***Програмна реалізація:***

import random  
  
  
def writeToFile(*line*, *fileName*):  
 with *open*(*fileName* + '.txt', "w") as file:  
 file.write(*line*)  
 file.close()  
  
  
def generateGamma(*str*, *size*):  
 if *len*(*str*) > *size*:  
 return *str*[0:*size*]  
 else:  
 return *str* + generateGamma(*str*, *size* - *len*(*str*))  
  
  
def generateReshuffle(*array*):  
 random.shuffle(*array*)  
  
  
def gammaEncrypt(*string\_for\_encrypt*, *gamma*, *alphabet*):  
 return moveTo(*string\_for\_encrypt*, *gamma*, *alphabet*, 1)  
  
  
def gammaDecrypt(*encrypt\_string*, *gamma*, *alphabet*):  
 return moveTo(*encrypt\_string*, *gamma*, *alphabet*, -1)  
  
  
def moveTo(*string*, *gamma*, *alphabet*, *action*):  
 output\_string = ''  
 length\_of\_string = *len*(*string*)  
 gamma = generateGamma(*gamma*, length\_of\_string)  
 for i in *range*(length\_of\_string):  
 summa = (*alphabet*.index(*string*[i].lower()) + *action* \* *alphabet*.index(*gamma*[i].lower())) % *len*(*alphabet*)  
 output\_string += *alphabet*[summa]  
 return output\_string  
  
  
ukrainian\_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"]  
text = "ЛьвівцемістоякерозташованеназаходіУкраїниславитьсясвоєювеликоюісторієючарівною" \  
 "архітектуроювеликимкультурнимспадкомтанеймовірноюгостинністюсвоїхмешканців"  
gamma = "футбол"  
  
generateReshuffle(ukrainian\_alphabet)  
writeToFile(text, 'text')  
encrypt\_string = gammaEncrypt(text, gamma, ukrainian\_alphabet)  
writeToFile(encrypt\_string, 'encrypt')  
decrypt\_string = gammaDecrypt(encrypt\_string, gamma, ukrainian\_alphabet)  
*print*(decrypt\_string)

***Опис реалізації:***

writeToFile() функція запису у файл, generateGamma() генерація гамии потрібної довжини

gammaEncrypt() функція шифрування, gammaDecrypt() функція дешифрування

moveTo() функція в якій реалізований алгоритм модульного гамування

***Результати виконання:***

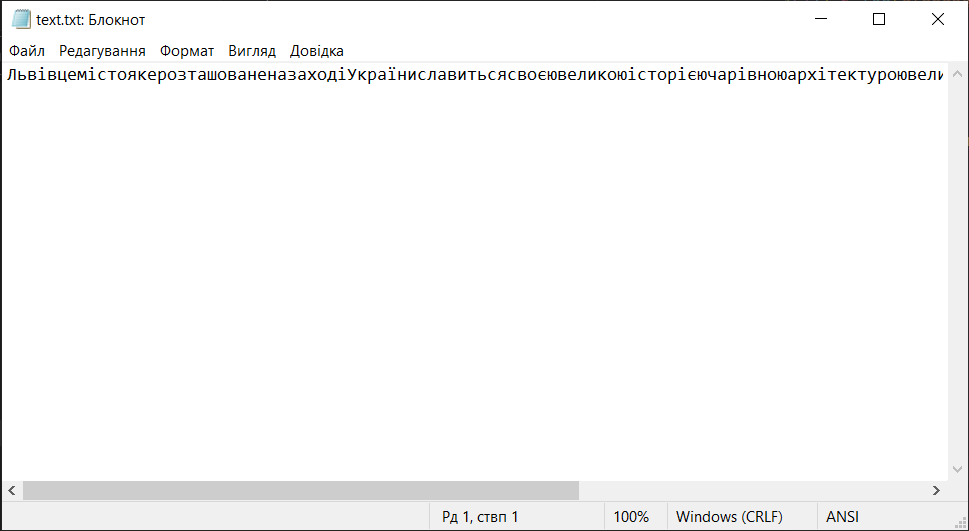
******

Рисунок Вхідна стрічка введена у файл

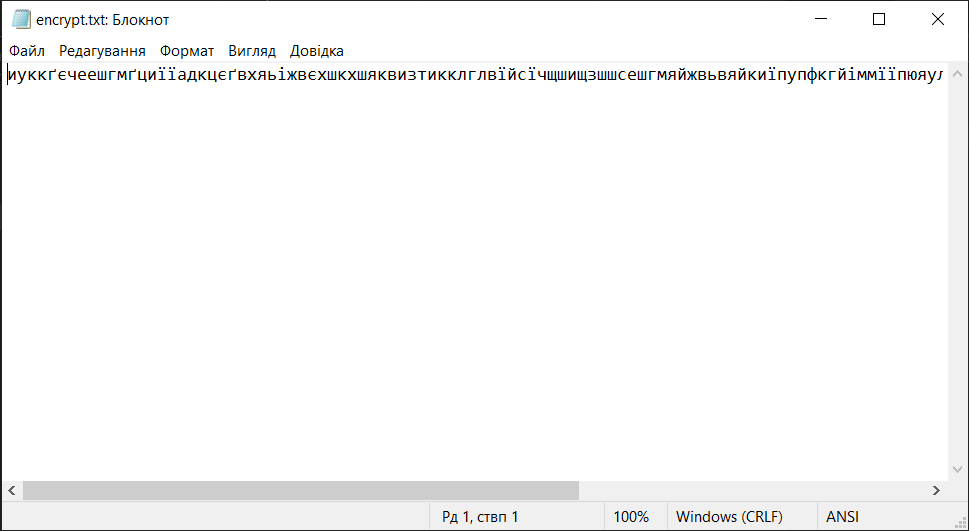
******

Рисунок 2 Шифрована стрічка введена у файл

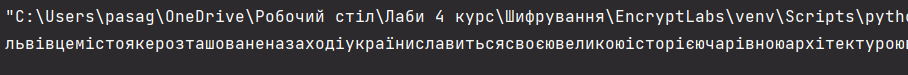


Рисунок 3 Розшифрована стрічка виведена в консоль