­­МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Інститут прикладної математики та фундаментальних наук

Кафедра прикладної математики

**ЗВІТ**

про виконання лабораторної роботи № 5  
з дисципліни

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

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

Львів 2023

­­ **Лабораторна робота №5**

***Тема:***

Шифр Плейфера: процедура шифрування

***Мета:***

Отримати навики практичної реалізації процедури шифрування для шифру Плейфера

***Постановка задачі:***

1. Запропонувати свою таблицю-ключ для реалізації шифру Плейфера для текстів українською мовою;

2. Програмно реалізувати процедуру шифрування, використовуючи свою таблицю ключ;

3. Написати свій текст про любимий напиток (довжиною понад 100 літер – відкритий текст; пробіли, розділові знаки та цифри не використовувати) і записати його у файл відкритого тексту; зашифрувати цей текст шифром Плейфера;

4. Вихідний текст (криптограму) записати у файл, який буде використовуватися у лабораторній роботі №6;

5. Підготувати опис/звіт про виконану роботу.

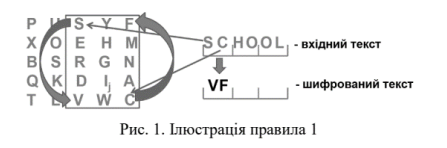
***Короткі теоретичні відомості:***

Якщо шифруємо по одній літері, то частотний аналіз дає можливість порівняно легко зламувати шифр, адже для кожної природної мови відомою є частота вживання кожної з літер. Щоб обійти цю проблему, використовують блокові методи, в яких здійснюється шифрування відразу кількох літер (блоку). Шифр Плейфера – це перший блоковий шифр, запропонований у 1854 році, у якому використовується найменший можливий блок – дві літери (їх називають ‘біграми’). Для англійської мови, наприклад, у цьому випадку пари літер AA, AB, AC, , AX, AY, AZ BA, BB, BC,…, BX, BY, BZ . . . ZA, ZB, ZC,…, ZX, ZY, ZZ можна розглядати як окремі літери значно ‘потужнішого’ алфавіту, який має 26\*26=676 літер із значно меншими частотами зустрічання. Як ключ при шифруванні і дешифруванні використовуємо матрицю, у яку випадковим чином записуємо літери абетки. Якщо в англійській абетці є 26 літер, то зручно використовувати, наприклад, матрицю 5х5. Щоб записати 26 літер англійської абетки у матрицю 5х5, одну з маловживаних літер ‘долучаємо’ до іншої.



При шифруванні використовуємо три правила, які нижче проілюстровано на прикладі шифрування слова ‘SCHOOL’. Пам’ятаємо, що шифруємо біграмами (відразу по 2 літери).

Правило 1: якщо дві літери окреслюють прямокутник, то вибираємо відповідні літери з протилежних кутів цього прямокутника. На рис. 1 проілюстровано, що літери ‘SC’ утворюють синій прямокутник, тому вибираємо відповідні їм літери ‘VF’ з протилежних кутів. Якщо б послідовність літер на вході була іншою – ‘CS’, то їм відповідали б при шифруванні літери ‘FV’.



Правило 2: якщо дві літери знаходяться в одному рядку матриці, то вибираємо відповідні літери правіші від них (у кільці). На рис. 2 проілюстровано, що літери ‘HO’ знаходяться в одному рядку, тому вибираємо правіші від них літери ‘ME’. Якщо б послідовність літер на вході була іншою – ‘OH’, то їм відповідали б при шифруванні літери ‘EM’. ‘У кільці’ означає, що якщо літера є останньою у рядку, то її заміняємо першою літерою рядка (наприклад, ‘RN’ при шифруванні відповідає ‘GB’).



Правило 3: якщо дві літери знаходяться в одному стовпці матриці, то вибираємо відповідні літери, які знаходяться у матриці нижче від них (у кільці). На рис. 3 проілюстровано, що літери ‘OL’ знаходяться в одному стовпці, тому вибираємо літери, які знаходяться нижче від них. Причому, якщо літера L є останньою у стовпці, то як ‘нижчу’ літеру вибираємо першу літеру стовпця – U. Якщо б послідовність літер на вході була іншою – ‘LO’, то їм відповідали б при шифруванні літери ‘US’.



Винятком із цих правил є випадок, коли у блоці для шифрування є дві однакові літери (наприклад, ‘RR’). У такому випадку домовляємося вибирати дві однакові літери, що розміщені у матриці правіше (у кільці). Тобто, ‘RR’ при шифруванні відповідає ‘GG’, натомість ‘AA’ відповідає ‘QQ’.

***Програмна реалізація:***

import random  
*# "ґ"*def cipherTable(*alphabet*):  
 random.shuffle(*alphabet*)  
 return [*alphabet*[i:i+8] for i in *range*(0, *len*(ukrainian\_alphabet), 8)]  
  
  
def encrypt(*text*, *table*):  
 out = ''  
 for i in *range*(0, *len*(*text*), 2):  
 two\_letters = *text*[i:i + 2]  
 i1, j1 = findInMatrix(two\_letters[0], *table*)  
 i2, j2 = findInMatrix(two\_letters[1], *table*)  
 *# По прямокутнику* if i1 != i2 and j1 != j2:  
 out += *table*[i2][j1] + *table*[i1][j2]  
 *# По рядку* elif i1 == i2:  
 out += *table*[i1][0] if j1 >= *len*(*table*[0]) - 1 else *table*[i1][j1 + 1]  
 out += *table*[i1][0] if j2 >= *len*(*table*[0]) - 1 else *table*[i1][j2 + 1]  
 *# По стовпцях* elif j1 == j2:  
 out += *table*[0][j1] if i1 >= *len*(*table*) - 1 else *table*[i1 + 1][j1]  
 out += *table*[0][j1] if i2 >= *len*(*table*) - 1 else *table*[i2 + 1][j2]  
 return out  
  
  
def findInMatrix(*el*, *table*):  
 for i in *range*(*len*(*table*)):  
 for j in *range*(*len*(*table*[0])):  
 if *el* == *table*[i][j]: return i, j  
  
  
def writeToFileMatrix(*table*, *fileName*):  
 with *open*(*fileName*, "w", *encoding*="utf-8") as file:  
 for row in *table*:  
 file.write(" ".join(*map*(str, row)) + "\n")  
  
  
def readFromFileMatrix(*fileName*):  
 matrix = []  
 with *open*(*fileName*, "r", *encoding*="utf-8") as file:  
 lines = file.readlines()  
 for line in lines:  
 row = *list*(line.strip())  
 row = [char for char in row if char != ' ']  
 matrix.append(row)  
 return matrix  
  
  
def writeToFile(*line*, *fileName*):  
 with *open*(*fileName* + '.txt', "w", *encoding*='utf-8') as file:  
 file.write(*line*)  
 file.close()  
  
  
ukrainian\_alphabet = ["а", "б", "в", "г", "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"]  
table = readFromFileMatrix('matrix.txt')  
writeToFileMatrix(table, 'matrix.txt')  
for i in table:  
 *print*(i)  
str = "ЛьвівЦеМістоЯкеРозташованеНаЗаходіУкраїниСлавитьсяСвоєюВеликоюІсторієюЧарівноюАрхітектуроюВеликимКультурнимСпадкомТаНеймовірноюГостинністюСвоїхМешканців"  
writeToFile(encrypt('гутковськийпавловодафутболгутк', table), 'text')  
*print*(encrypt('гутковськийпавловодафутболгутк', table))

***Опис реалізації:***

writeToFile() функція запису у файл, cipherTable() генерація матриці Плейфера

encrypt() функція шифрування, findInMatrix() функція пошуку елемента в матриці

writeToFileMatrix() функція запису матриці у файл, readToFileMatrix() функція зчитування

матриці з файлу

***Результати виконання:***

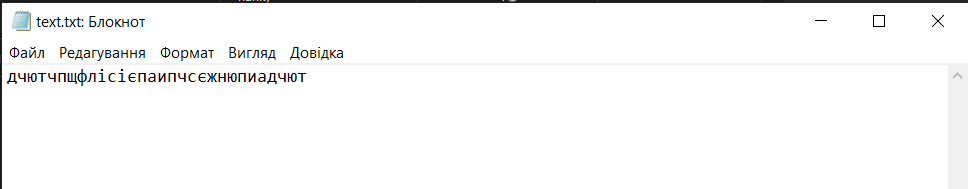
******

Рисунок 2 Шифрована стрічка введена у файл

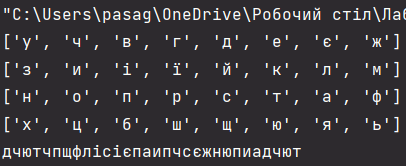


Рисунок 3 Зашифрована стрічка виведена в консоль з матрицею