МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра кібербезпеки

**Звіт до лабораторної роботи № 2**

на тему “ Програмна реалізація шифру Віженера ”

Виконав студент(ка) *Борщ Дмитро*

Варіант *№ 1*

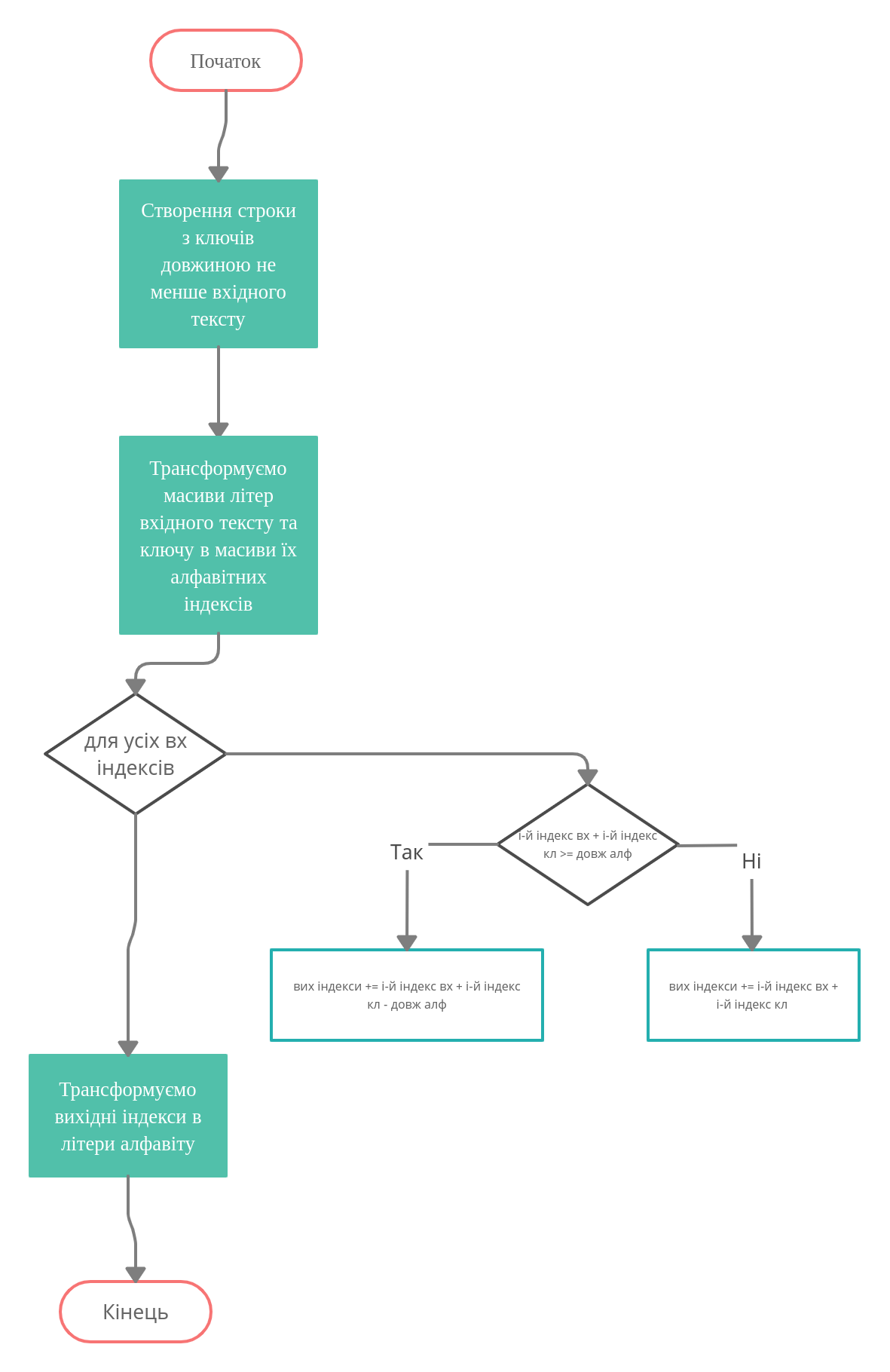
Група *КБ-01*

Перевірила Лаврик Т.В

**Суми 2022**

**ЗВІТ 2**

**Завдання 1 (4 б.).** Побудувати блок-схему алгоритму шифрування Віженера.



Для дешифрування алгоритм буде той же, але замість додавання по модую, будемо віднімати. І коли наш індекс менше нуля, то додаємо до нього розмір алфавіту

**Завдання 2 (10 б.).** Написати програмний код, який зашифровує введену з клавіатури українською мовою інформацію\* за допомогою шифру Віженера. Ключове слово користувач також має вводити самостійно.

Мова програмування обирається самостійно.

\* Враховувати великі та малі літери.

Код програми на мові Python:

import os

import time as t

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',

'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 clear\_screen():

if(os.name == 'posix'):

os.system('clear')

else:

os.system('cls')

def wait\_plz(time\_to\_sleep):

t.sleep(time\_to\_sleep)

def is\_in\_vector(arr, key):

for i in range(len(arr)-1):

if arr[i] == key: return True

return False

def convert\_to\_indeces(str):

arr = []

for i in range(len(str)):

for j in range(len(alphabet)):

if str[i] == alphabet[j]: arr.append(j)

return arr

def convert\_to\_string(arr):

str = ""

for i in range(len(arr)):

for j in range(len(alphabet)):

if arr[i] == j: str += alphabet[j]

return str

def encrypt(user\_input, key\_word):

key = encrypted = ""

user\_indices = key\_indices = encrypted\_indices = []

while len(key) <= len(user\_input): key += key\_word

user\_indices = convert\_to\_indeces(user\_input)

key\_indices = convert\_to\_indeces(key)

for i in range(len(user\_indices)):

if user\_indices[i] + key\_indices[i] >= len(alphabet):

encrypted\_indices.append(user\_indices[i] + key\_indices[i] - len(alphabet))

else: encrypted\_indices.append(user\_indices[i] + key\_indices[i])

encrypted = convert\_to\_string(encrypted\_indices)

return encrypted

def decrypt(user\_input, key\_word):

key = decrypted = ""

user\_indices = key\_indices = decrypted\_indices = []

while len(key) <= len(user\_input): key += key\_word

user\_indices = convert\_to\_indeces(user\_input)

key\_indices = convert\_to\_indeces(key)

for i in range(len(user\_indices)):

if user\_indices[i] - key\_indices[i] < 0:

decrypted\_indices.append(user\_indices[i] - key\_indices[i] + len(alphabet))

else: decrypted\_indices.append(user\_indices[i] - key\_indices[i])

decrypted = convert\_to\_string(decrypted\_indices)

return decrypted

def main():

user\_input = int()

output = ""

while True:

clear\_screen()

print("Hi, enter text: ")

user\_input = str(input())

print("Enter keyword: ")

key\_word = str(input())

print(

"1. Encrypt\n"+

"2. Decrypt\n"+

"Choose your option: "

)

user\_choise = int(input())

match user\_choise:

case 1:

output = encrypt(user\_input, key\_word)

print("\nEncrypted text is \n\n" + output + "\n\n")

break

case 2:

output = decrypt(user\_input, key\_word)

print("\nDecrypted text is \n\n" + output + "\n\n")

break

case \_:

print("Wrong input!!! Try again...")

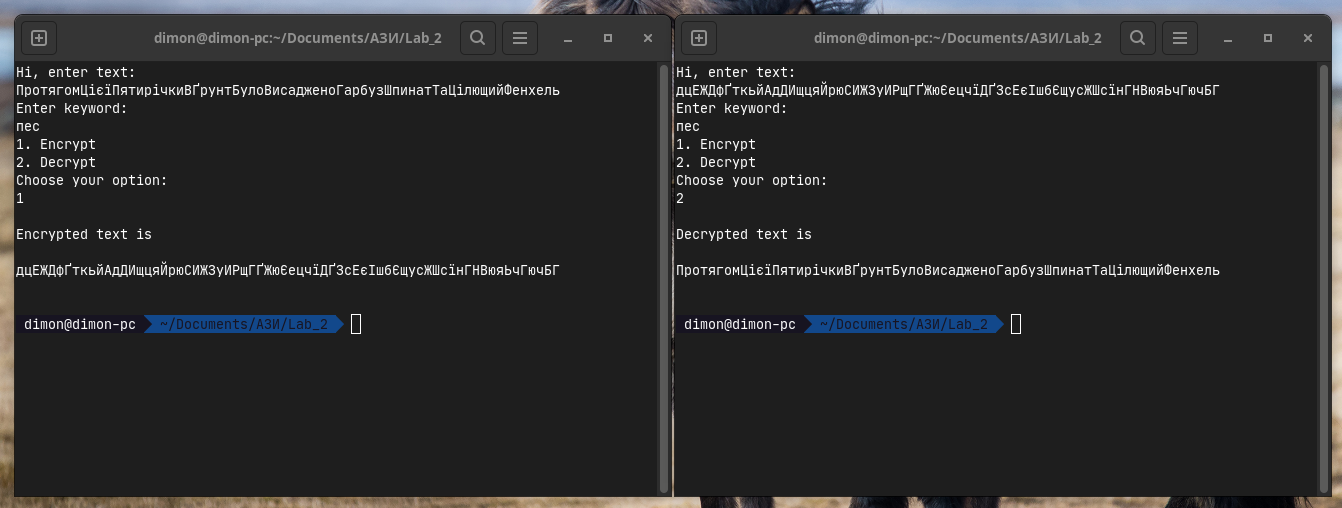
wait\_plz(1)

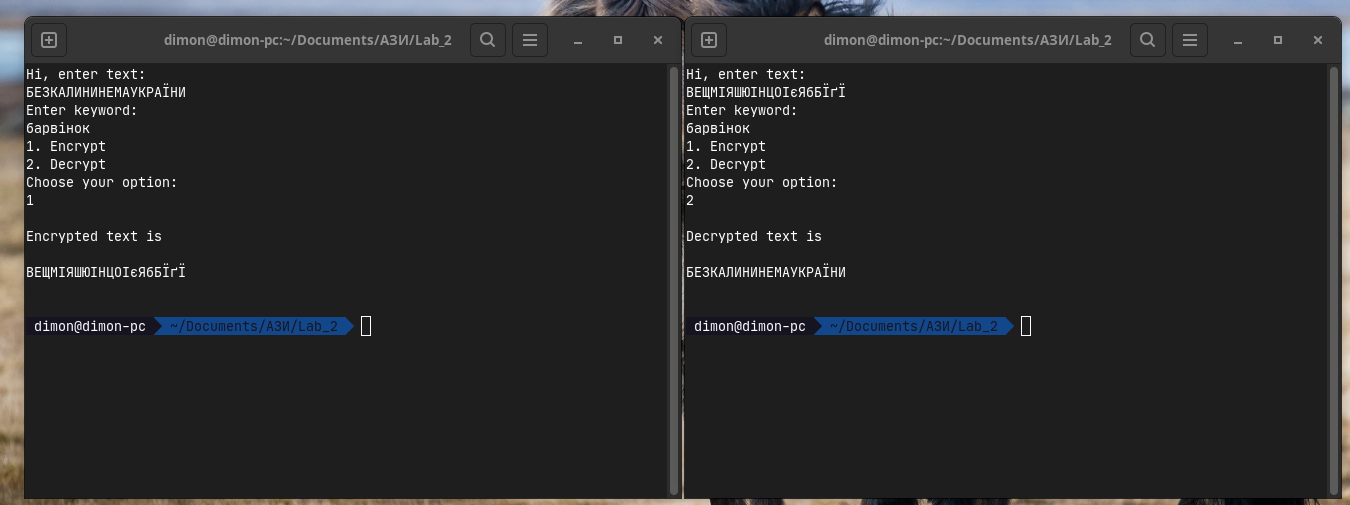
exit(0)

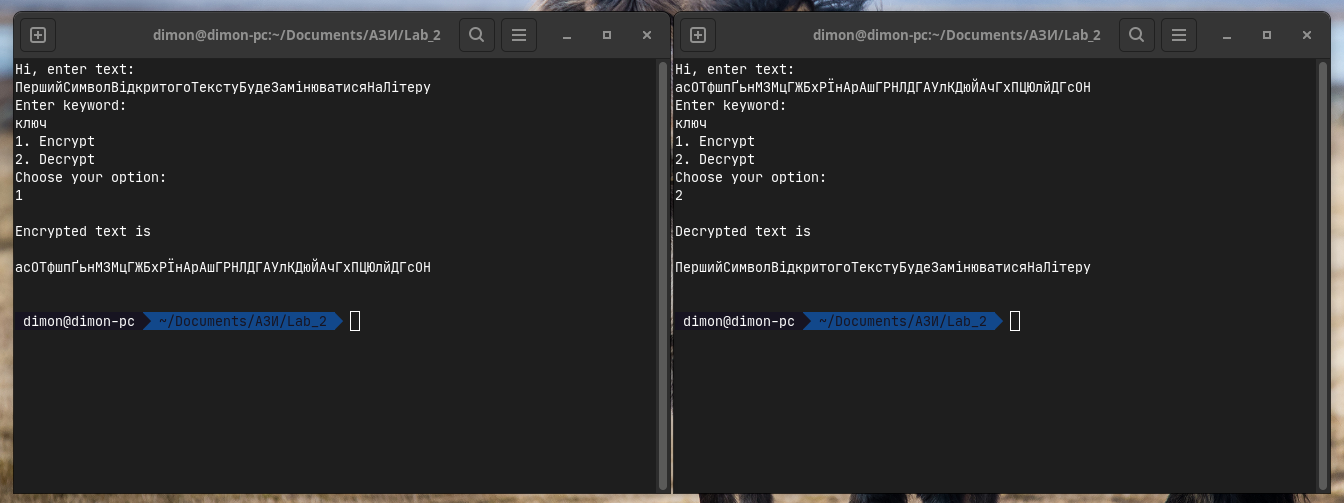
if \_\_name\_\_ == "\_\_main\_\_":

main()

Скріншоти з результатами роботи програми (мінімум 3 різні тексти)

 Рис. 1 - Перший тест.

Рис. 2 - Другий тест.

Рис. 3 - Третій тест.