МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Кафедра кібербезпеки

Дисципліна “Системи та засоби криптоаналізу”

Звіт до лабораторної роботи № 3

на тему “ Дослідження криптоаналітичних атак на афінний шифр. Частина 1**”**

Студент	Борщ Д.О.
Варіант	№ 1
Група	КБ-01
Перевірила	Лаврик Т.В

Суми 2022

ЗВІТ

1. (5 б.) Програмний код, який шифрує введену з клавіатури українською мовою інформацію за допомогою афінної системи підстановок Цезаря. Ключ шифрування (a, b) і зашифрований текст зберегти для звіту.

Текст програми:

```
import sys
import math


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 brute(string):
    output = ""
    for keyA in range(0, len(alphabet)-1):
        if math.gcd(keyA, len(alphabet)) == 1:
            for keyB in range(0, len(alphabet)-1):
                print(
                    f"Result with keys: A = {keyA} | B = {keyB}\n",
                    decrypt(string, keyA, keyB),
                    "\n\n"
                )
```

```
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")
        brute(inputText)

    return 0


if __name__ == "__main__":
    main()
```

Скріншоти з результатами роботи програми (мінімум 3 різні тексти)

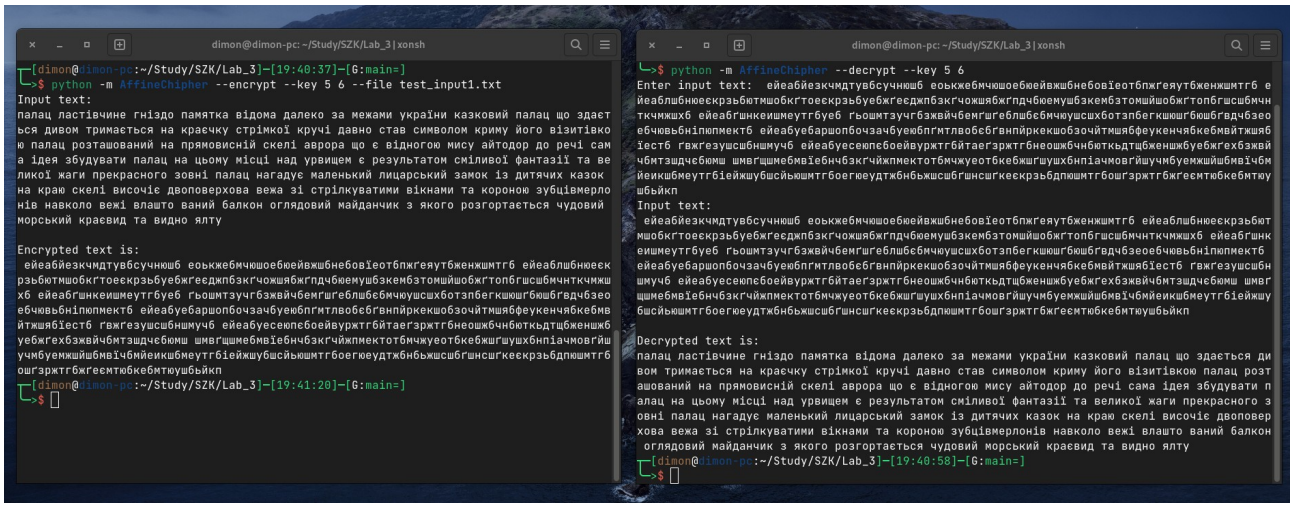

Рис. 1 — Перший приклад роботи програми з ключами 5 та 6.

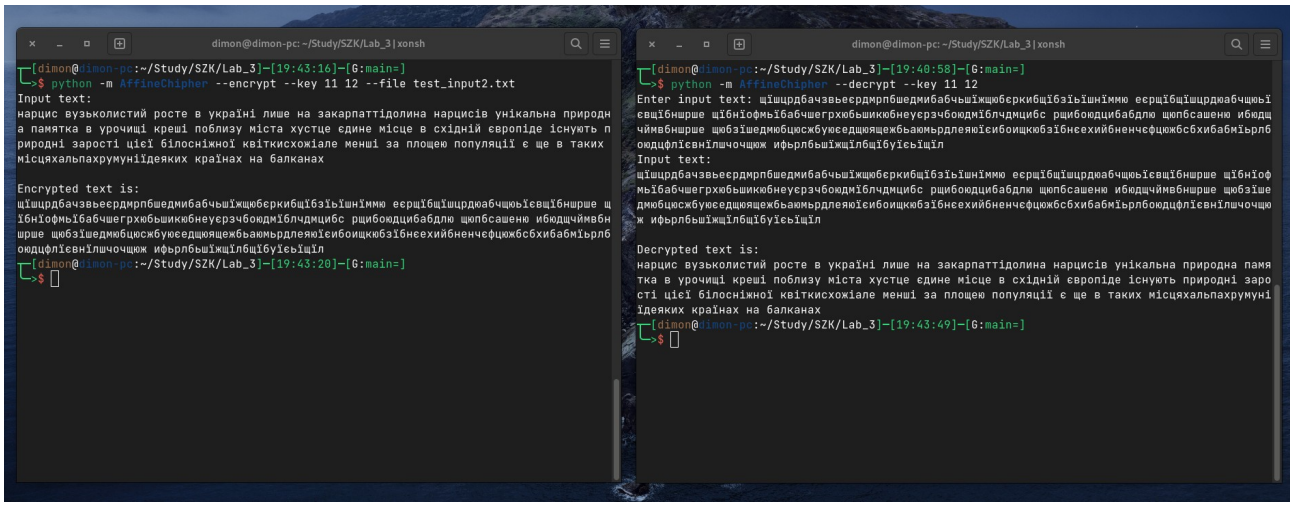

Рис. 2 — Другий приклад роботи програми з ключами 11 та 12.

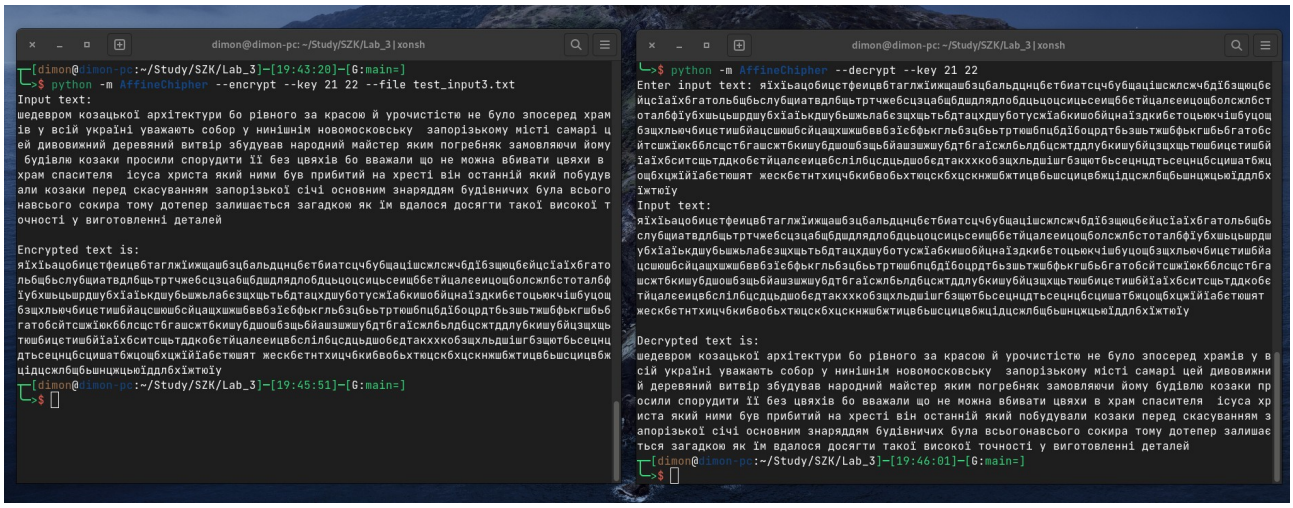

Рис. 3 — Третій приклад роботи з ключами 21 та 22.

2. (3 б.) Блок-схема алгоритму реалізації методу повного перебору.

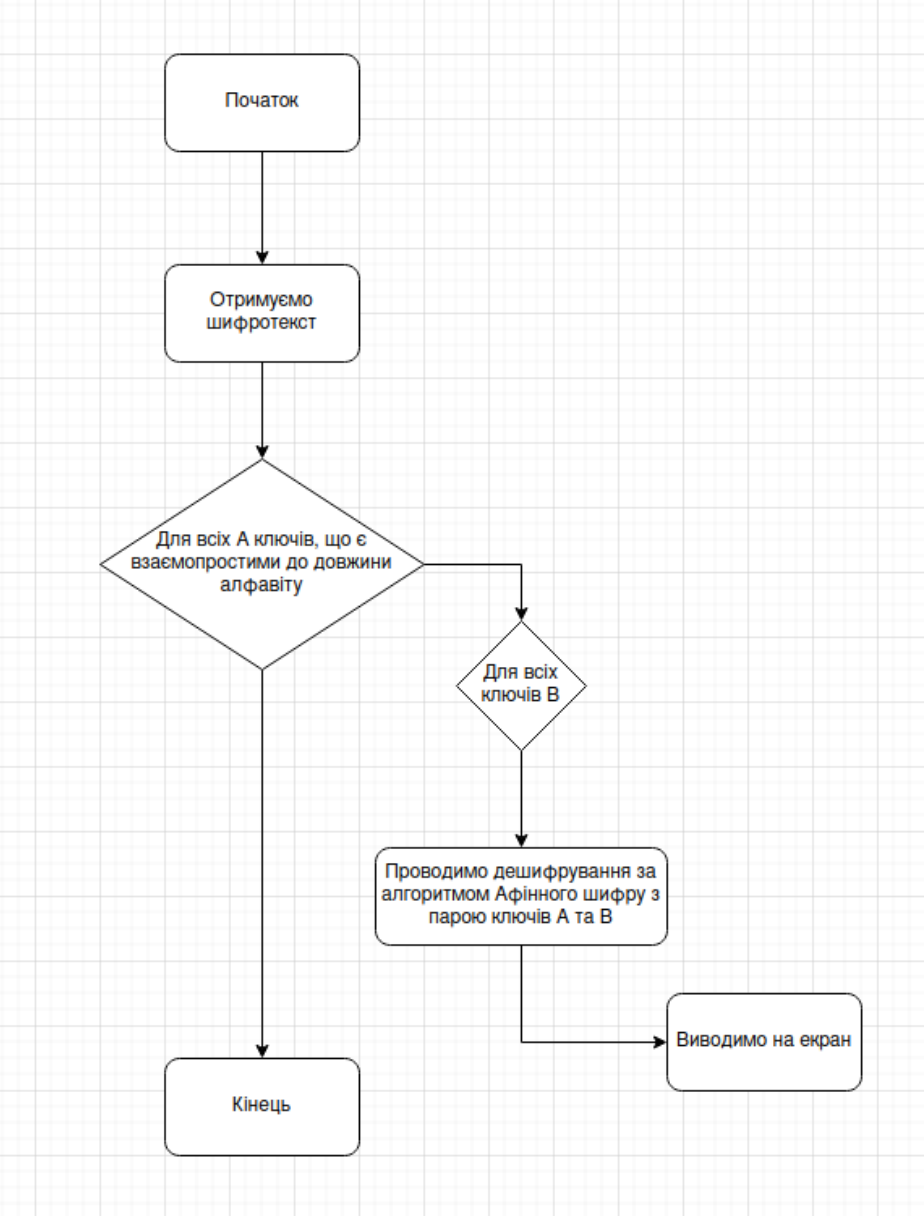

Рис. 4 — Блок схема

3. (5 б.) Програмний код методу повного перебору.

Текст програми

Повний текст програми приклав у першому завданні

Скріншоти з результатами роботи програми (мінімум 3 різні шифротексти)

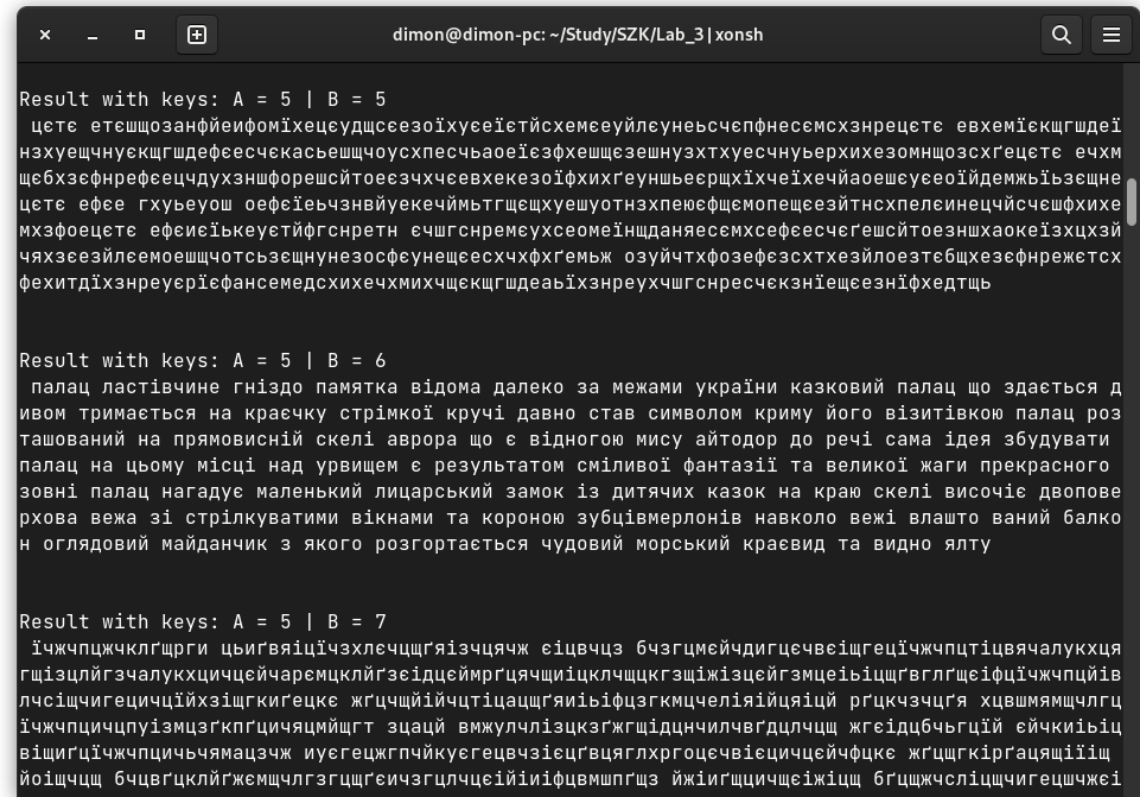

Рис. 5 — Перший приклад повного перебору.

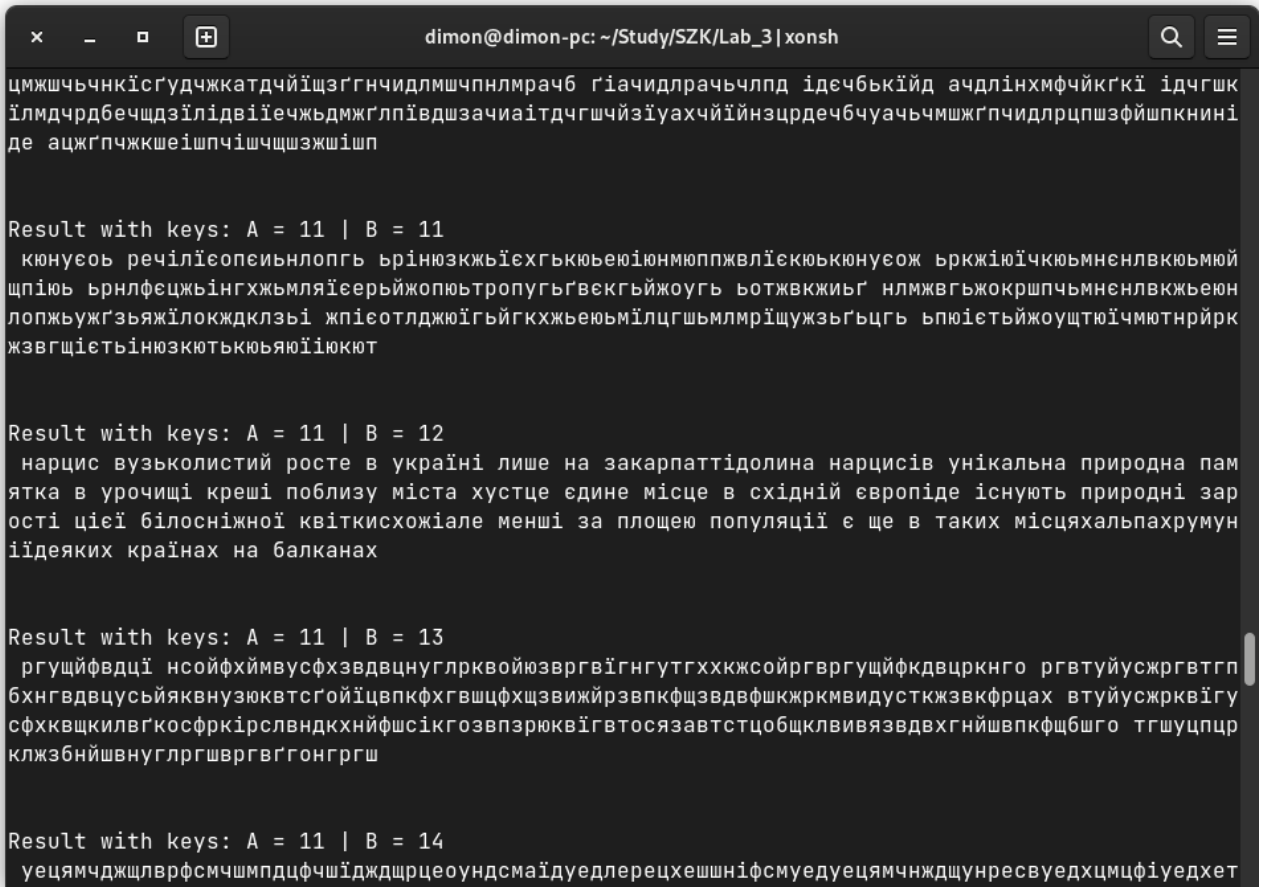

Рис. 6 — Другий приклад повного перебору.

Рис. 7 — Третій приклад повного перебору.