НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

з лабораторної роботи №*1*

з дисципліни «Криптографічні методи захисту інформації»

на тему:

*Розробка криптосистем на основі шифрів Цезаря та Тритемія*

|  |  |
| --- | --- |
| Виконав: | Керівник: |
| студент групи КМ-01 | *ст. викладач Бай Ю. П.* |
| *Дюбакін Р. С.* |  |

Київ — 2022

ЗМІСТ

[ЗАВДАННЯ 2](#_Toc94713306)

[Основні теоретичні відомості з шифру Цезаря 3](#_Toc94713307)

[Завдання з шифру Цезаря 3](#_Toc94713308)

[Основні теоретичні відомості з шифру Тритемія 4](#_Toc94713309)

[Завдання з шифру Тритемія 4](#_Toc94713310)

[Список літератури 6](#_Toc94713311)

[Додаток 1 7](#_Toc94713312)

[Додаток 2 8](#_Toc94713313)

# ЗАВДАННЯ

***Мета роботи:*** розробити криптосистеми на основі шифрів Цезаря та Тритемія.

1. Зашифрувати текст довжиною від 10 до 50 символів, користуючись шифром Цезаря з певним ключем *K*. Результат записати в [Таблицю\_1](https://docs.google.com/spreadsheets/d/1vg-aCHGGX2oZDjuAWWRWecRBWjwAYLmkgonEp1CFBGA/edit?usp=sharing), в СВІЙ РЯДОК, стовпчик AG (*1 бал*).

2. Дано криптотекст *C*, одержаний шифруванням Цезаря з невідомим ключем *K* (див. [Таблицю\_1](https://docs.google.com/spreadsheets/d/1vg-aCHGGX2oZDjuAWWRWecRBWjwAYLmkgonEp1CFBGA/edit?usp=sharing), рядок НАД СВОЇМ, стовпчик AG). Методом повного перебору всіх потенційних рішень (методом «грубої сили») дешифрувати заданий криптотекст. У відповіді навести дешифрований текст *P* та ключ *K*. (*2 бали*).

3. Зашифрувати текст довжиною від 10 до 20 символів, користуючись шифром Тритемія за лінійним законом *y = (x + A\*p + B) mod N*. Записати початковий текст та криптотекст в [Таблицю\_2](https://docs.google.com/spreadsheets/d/19vV6-YDYE9fHXQ3q9iaDqeYQlLHj69xGzTOhQase2zI/edit?usp=sharing), в СВІЙ РЯДОК, стовпчики AG, AH (*1 бал*).

4. Уявіть, що Ви перехопили початковий текст та криптотекст (див. [Таблицю\_2](https://docs.google.com/spreadsheets/d/19vV6-YDYE9fHXQ3q9iaDqeYQlLHj69xGzTOhQase2zI/edit?usp=sharing), рядок НАД СВОЇМ, стовпчики AG, AH). Необхідно визначити ключ: знайти параметри *A, B*, але не методом «грубої сили», а розв’язуючі систему двох рівнянь. Відповідь записати в стовпчик AI (рядок НАД СВОЇМ) (*2 бали*).

5. Зашифрувати текст довжиною від 10 до 50 символів, користуючись шифром Тритемія **за квадратичним законом** **(*A, B, C*****від 1 до 5)**. Записати зашифрований текст в [Таблицю\_2](https://docs.google.com/spreadsheets/d/19vV6-YDYE9fHXQ3q9iaDqeYQlLHj69xGzTOhQase2zI/edit?usp=sharing), в СВІЙ РЯДОК, стовпчик AJ (*1 бал*).

6. Методом "brute force" дешифрувати текст, що знаходиться в стовпчику AJ, в рядку НАД СВОЇМ (відомо, що був використаний шифр Тритемія, за квадратичним законом, *A, B, C* від 1 до 5). Відповідь записати в стовпчик AK, знайдений ключ - в AL (рядок НАД СВОЇМ) (*2 бали*).

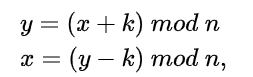
7. Оформити звіт (*1 бал*).

# Основні теоретичні відомості з шифру Цезаря

Шифр Цезаря або шифр зсуву — симетричний моноалфавітний алгоритм шифрування, в якому кожна буква відкритого тексту заміняється на ту, що віддалена від неї в алфавіті на сталу кількість позицій. Римський імператор Юлій Цезар використовував для приватного листування шифр зсуву з ключем 3 — замість літери A підставляв D, замість B — E і так далі.

Як і будь-який моноалфавітний шифр, Цезарів вразливий до частотного криптоаналізу. Навіть легше зламати його простим перебором, бо кількість можливих ключів зовсім невелика — для українського алфавіту 32 варіанти. Метод Цезаря не надає майже ніякого криптографічного захисту, але він ліг в основу дещо складніших алгоритмів, наприклад шифру Віженера. Варіант шифру зсуву ROT13 використовується в англомовному сегменті інтернету для приховування спойлерів, розгадок головоломок тощо.

Принцип дії полягає в тому, щоб циклічно зсунути алфавіт, а ключ — це кількість літер, на які робиться зсув.

Якщо зіставити кожному символу алфавіту його порядковий номер (нумеруючи з 0), то шифрування і дешифрування можна виразити формулами:

де x — порядковий номер символу відкритого тексту, y — порядковий номер символу шифрованого тексту, n — потужність алфавіту, а k — ключ.

# Завдання з шифру Цезаря

1. Зашифруємо текст «Illusion is the first of all pleasures», користуючись шифром Цезаря з ключем *К* = 5

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| 🡫 | 🡫 | 🡫 | … |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e |

Результат шифрування:

Nqqzxnts nx ymj knwxy tk fqq uqjfxzwjx

2. Дано криптотекст, одержаний шифруванням Цезаря:

VMVIP URP ZJ R EVN TYRETV

Одержимо всі варіанти його дешифрування:

|  |  |
| --- | --- |
| K = 0 | vmvip urp zj r evn tyretv |
| K = 1 | wnwjq vsq ak s fwo uzsfuw |
| K =2 | xoxkr wtr bl t gxp vatgvx |
| K = 3 | ypyls xus cm u hyq wbuhwy |
| K = 4 | zqzmt yvt dn v izr xcvixz |
| K = 5 | aranu zwu eo w jas ydwjya |
| K = 6 | bsbov axv fp x kbt zexkzb |
| K = 7 | ctcpw byw gq y lcu afylac |
| K = 8 | dudqx czx hr z mdv bgzmbd |
| K = 9 | every day is a new chance |
| … |  |
| K = 25 | uluho tqo yi q dum sxqdsu |

Аналізуючи одержані результати, робимо висновок щодо початкового тексту та ключа:

*P* = ‘EVERY DAY IS A NEW CHANCE’, *K* = 9

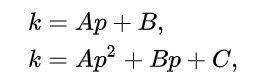
# Основні теоретичні відомості з шифру Тритемія

Шифр Тритеміуса – система шифрування, розроблена Йоганном Тритемієм. Це удосконалений шифр Цезаря, тобто шифр підстановки. За алгоритмом шифрування, кожен символ повідомлення зміщується на символ, що відстає на певний крок. Тут крок усунення робиться змінним, тобто що залежить від будь-яких додаткових чинників. Наприклад, можна задати закон усунення у вигляді лінійної функції (рівняння зашифрування) позиції літери, що шифрується. Сама функція має гарантувати ціле значення. Пряма функція шифрування повинна мати зворотну функцію шифрування, також цілочисленну.

Рівнянням зашифрування називається співвідношення, що описує процес утворення зашифрованих даних із відкритих даних у результаті перетворень, заданих алгоритмом криптографічного перетворення.

Рівняння зашифрування для шифру Тритеміуса має такий вигляд:

де L- номер зашифрованої літери в алфавіті; m — номер позиції літери тексту, що шифрується, в алфавіті; k - крок усунення (функціональна залежність від позиції літери в повідомленні); N – число букв алфавіту.

Деякі варіанти обчислення кроку зміщення k:

де p – позиція літери у повідомленні.

# Завдання з шифру Тритемія

3. Зашифруємо текст «ALLER ANFANG IST SCHWER», користуючись шифром Тритемія за лінійним законом з параметрами *A* = 2, *B* = 3

Результат шифрування:

DQSNC PEYVKF LXA DPWNXM

4. Дано початковий текст та криптотекст, одержаний шифруванням Тритемія за лінійним законом з невідомими параметрами *A*, *B* ( *A*, *B* – цілі числа від 1 до 5):

*plaintext* = ' WINX CLUB ', *cryptotext* = ' ZOWJ UGSC '

Визначимо параметри *А, В*:

5. Зашифруємо текст «ENDE GUT, ALLES GUT», користуючись шифром Тритемія за квадратичним законом з параметрами *A* = 1, *B* = 3, *С* = 2 (*A*, *B*, *С* ϵ [1; 5]):

Результат шифрування:

GTPY WYN, CLLGY AYJ

6. Дано криптотекст, одержаний шифруванням Тритемія за квадратичним законом з невідомими параметрами *A*, *B*, *С* ( *A*, *B*, *С* – цілі числа від 1 до 5):

FDXZZRGMAG KZI IVB UDATMMC QL SCZP

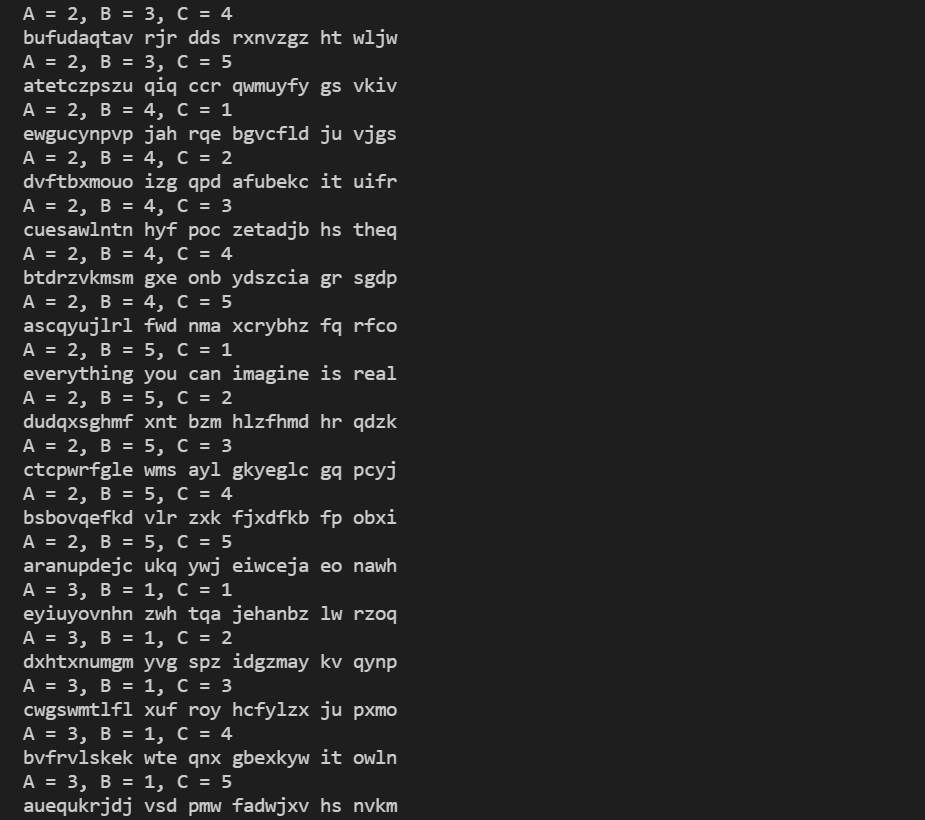
Одержимо всі варіанти його дешифрування для *A*, *B*, *С* ϵ [1; 5]:

Рис. 1.1 – Скріншот частини результату роботи програми дешифрування

Аналізуючи одержані результати, робимо висновок щодо початкового тексту та значень параметрів *A*, *B*, *С*:

*P* = ' EVERYTHING YOU CAN IMAGINE IS REAL ', *A* = 2, *B* = 5, *С* = 1

# Список літератури

1. Тарнавський Ю.А. Технології захисту інформації [Електронний ресурс] / Ю. А. Тарнавський. – Київ: КПІ ім. Ігоря Сікорського, 2018. – 162 с.
2. Шнайер Б. Прикладная криптография: Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – М.: Диалектика, 2003. – 610 с.
3. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии. – М.: Гелиос АРВ, 2001. – 480 с.
4. Столлингс В. Криптография и защита сетей: принципы и практика, 2-е изд.: Пер. с англ. – М.: «Вильямс», 2001. – 672 с.
5. Сингх С. Книга шифров. Тайная история шифров и их расшифровки, Пер. с англ. – М.: «Астрель», 2007. – 447 с.

# Додаток 1

Текст програми, що реалізує криптосистему Цезаря

alphabet\_eng = 'abcdefghijklmnopqrstuvwxyz'

alphabet\_rus = 'абвгдеёжзийклмнопрстуфхцчшщъыьэюя'

def zesar(alphabet, plainText, step):

    cryptoText = ''

    for letter in plainText.lower():

        if letter in alphabet:

            cryptoText += alphabet[((alphabet.find(letter)) + step) % len(alphabet)]

        else:

            cryptoText += letter

    print(cryptoText)

    return cryptoText

def deZesar(alphabet, cryptoText):

    step = len(alphabet)

    while step != 0:

        plainText = ''

        for letter in cryptoText.lower():

                if letter in alphabet:

                    plainText += alphabet[((alphabet.find(letter)) - step) % len(alphabet)]

                else:

                    plainText += letter

        print(len(alphabet) - step)

        print(plainText)

        step -= 1

deZesar(alphabet\_eng, 'VMVIP URP ZJ R EVN TYRETV')

# Додаток 2

Текст програми, що реалізує криптосистему Тритемія

ALPHABET\_ENG = 'abcdefghijklmnopqrstuvwxyz'

ALPHABET\_RUS = 'абвгдеёжзийклмнопрстуфхцчшщъыьэюя'

def tritemiusLinear(alphabet, plainText, valueA, valueB):

    cryptoText = ''

    letter\_position = 0

    for letter in plainText.lower():

        if letter in alphabet:

            valueK = (valueA \* letter\_position +valueB) % len(alphabet)

            valueY = (alphabet.find(letter) + valueK) % len(alphabet)

            cryptoText += alphabet[valueY]

            # letter\_position += 1

        else:

            cryptoText += letter

        letter\_position += 1

    print(cryptoText)

    return cryptoText

def tritemiusSquare(alphabet, plainText, valueA, valueB, valueC):

    cryptoText = ''

    letter\_position = 0

    for letter in plainText.lower():

        if letter in alphabet:

            valueK = (valueA \* letter\_position \* letter\_position + valueB \* letter\_position + valueC) % len(alphabet)

            valueY = (alphabet.find(letter) + valueK) % len(alphabet)

            cryptoText += alphabet[valueY]

            # letter\_position += 1

        else:

            cryptoText += letter

        letter\_position += 1

    print(cryptoText)

    return cryptoText

def deTritemiusLinear(alphabet, cryptoText):

    plainText = ''

    letter\_position = 0

    for valueA in range(1, 6):

        for valueB in range(1, 6):

                for letter in cryptoText.lower():

                    if letter in alphabet:

                        valueK = (valueA \* letter\_position + valueB) % len(alphabet)

                        valueX = (alphabet.find(letter) -

                                  valueK) % len(alphabet)

                        plainText += alphabet[valueX]

                    else:

                        plainText += letter

                    letter\_position += 1

                print(f'A = {valueA}, B = {valueB}')

                print(plainText)

                letter\_position = 0

                plainText = ''

def deTritemiusSquare(alphabet, cryptoText):

    plainText = ''

    letter\_position = 0

    for valueA in range(1,6):

        for valueB in range(1,6):

            for valueC in range(1,6):

                for letter in cryptoText.lower():

                    if letter in alphabet:

                        valueK = (valueA \* letter\_position \* letter\_position + valueB \* letter\_position + valueC) % len(alphabet)

                        valueX = (alphabet.find(letter) - valueK) % len(alphabet)

                        plainText += alphabet[valueX]

                    else:

                        plainText += letter

                    letter\_position += 1

                print(f'A = {valueA}, B = {valueB}, C = {valueC}')

                print(plainText)

                letter\_position = 0

                plainText = ''

# tritemiusSquare(ALPHABET\_ENG, "Ende gut, alles gut", 1, 3, 2)

# tritemiusLinear(ALPHABET\_ENG, "Aller Anfang ist schwer", 2, 3)

# tritemiusLinear(ALPHABET\_ENG, "AND IT IS LIFE", 1, 2)

deTritemiusSquare(ALPHABET\_ENG, "FDXZZRGMAG KZI IVB UDATMMC QL SCZP")

# deTritemiusSquare(ALPHABET\_ENG, "gtpy wyn, cllgy ayj")

# deTritemiusLinear(ALPHABET\_ENG, "ZOWJ UGSC")