|  |
| --- |
| [Название организации] |
| [Заголовок документа] |
| [Подзаголовок документа] |

|  |
| --- |
|  |

1 ЗАГАЛЬНА ЧАСТИНА 2

1.1 Постановка задачі 2

1.2 Дослідження і аналіз об'єкту програмування С++ 3

1.2.1 Шифр «Магічний квадрат» 3

1.2.2 Квадрат Полібія 3

1.2.3 Порівняльний аналіз 4

1.3 Використані програмні засоби 4

1.3.1 Середовище розробки 4

1.3.2 Мова програмування 5

1.3.3 Додаткові бібліотеки 5

1.4 Вимоги до апаратного та програмного забезпечення 5

1.4.1 Апаратні вимоги 5

# ЗАГАЛЬНА ЧАСТИНА

## Постановка задачі

Метою даної курсової роботи є дослідження та практична реалізація класичних криптографічних алгоритмів перестановки, а саме шифру «Магічний квадрат» та квадрату Полібія. Ці алгоритми представляють собою фундаментальні методи криптографії, які демонструють принципи перестановки символів та заміщення букв відповідно.

Основними завданнями роботи є:

Теоретичне дослідження принципів роботи обраних криптографічних алгоритмів, включаючи вивчення їх історичного розвитку та математичних основ. Важливо розуміти, що шифр «Магічний квадрат» базується на принципі перестановки символів повідомлення згідно з певною матрицею, де кожен елемент має унікальне числове значення. Квадрат Полібія, у свою чергу, є методом заміщення, який кодує кожну букву алфавіту парою координат у квадратній таблиці.

Аналіз переваг та недоліків кожного з алгоритмів з точки зору криптографічної стійкості є наступним важливим завданням. Необхідно оцінити рівень захисту, який забезпечують ці методи, та визначити їх уразливості до різних типів криптоаналітичних атак.

Практична реалізація обох алгоритмів у програмному середовищі дозволить продемонструвати їх функціонування та провести порівняльний аналіз ефективності. Це включає розробку програмного коду для шифрування та дешифрування повідомлень.

Проведення тестування розроблених програм на різних типах вхідних даних забезпечить перевірку коректності реалізації та виявлення можливих помилок у логіці алгоритмів.

Створення зручного користувацького інтерфейсу для взаємодії з реалізованими алгоритмами підвищить практичну цінність розробленого програмного продукту.

## Дослідження і аналіз об'єкту програмування С++

Об'єктом дослідження є два фундаментальні криптографічні алгоритми, кожен з яких представляє різний підхід до забезпечення конфіденційності інформації.

### Шифр «Магічний квадрат»

Шифр «Магічний квадрат» належить до класу перестановочних шифрів та базується на використанні магічного квадрата як ключа для перестановки символів повідомлення. Магічний квадрат являє собою квадратну матрицю розміром n×n, заповнену числами від 1 до n², де сума чисел у кожному рядку, стовпці та діагоналі є однаковою.

Принцип роботи алгоритму полягає у наступному. Спочатку повідомлення записується в таблицю згідно з порядком чисел у магічному квадраті. Потім символи зчитуються в іншому порядку, що й створює зашифроване повідомлення. Ключем шифрування служить сам магічний квадрат та спосіб зчитування символів.

Історично магічні квадрати використовувались у різних культурах для містичних та математичних цілей. У криптографії вони знайшли застосування завдяки своїй здатності створювати складні перестановки, які важко відновити без знання ключа.

Криптографічна стійкість даного методу залежить від розміру магічного квадрата та способу його заповнення. Більші квадрати забезпечують більшу кількість можливих перестановок, що ускладнює криптоаналіз. Однак цей шифр є вразливим до частотного аналізу, оскільки він не змінює частоту появи символів у тексті.

### Квадрат Полібія

Квадрат Полібія є одним з найдавніших відомих методів шифрування та належить до класу моноалфавітних шифрів заміщення. Цей метод був винайдений давньогрецьким істориком Полібієм у II столітті до нашої ери.

Алгоритм використовує квадратну таблицю розміром 5×5 або 6×6, у якій розміщені букви алфавіту. Кожна буква кодується парою чисел, що відповідають номеру рядка та стовпця, в яких вона розташована. Наприклад, у стандартному латинському варіанті буква A може бути закодована як (1,1), якщо вона знаходиться в першому рядку та першому стовпці.

Основною перевагою квадрата Полібія є його простота та можливість передачі закодованих повідомлень різними способами, включаючи візуальні сигнали та звукові коди. Це робило його особливо корисним для військових цілей у давнину.

Криптографічна стійкість квадрата Полібія є відносно низькою з сучасної точки зору, оскільки він залишається моноалфавітним шифром. Це означає, що кожна буква завжди кодується однаково, що робить його вразливим до частотного аналізу. Проте його можна посилити шляхом використання ключових слів для заповнення квадрата або комбінування з іншими криптографічними методами.

### Порівняльний аналіз

Обидва алгоритми представляють класичні підходи до криптографії, але мають принципово різні механізми роботи. Шифр «Магічний квадрат» змінює порядок символів, зберігаючи їх оригінальний вигляд, тоді як квадрат Полібія замінює кожен символ числовою парою.

З точки зору складності реалізації обидва алгоритми є відносно простими, що робить їх відмінними об'єктами для навчальних цілей. Однак їх криптографічна стійкість не відповідає сучасним стандартам безпеки, що обмежує їх практичне застосування в реальних системах захисту інформації.

## **Використані програмні засоби**

Для реалізації та дослідження обраних криптографічних алгоритмів було обрано комплекс сучасних програмних засобів, які забезпечують ефективну розробку та тестування програмного коду.

### Середовище розробки

Основним середовищем розробки обрано Visual Studio Code, який є сучасним, безкоштовним та потужним редактором коду з підтримкою широкого спектру мов програмування. Цей вибір обумовлений наступними факторами: наявність інтелектуального автодоповнення коду, що прискорює процес розробки та зменшує кількість помилок; вбудовані засоби для відлагодження програм, які дозволяють ефективно виявляти та виправляти помилки в коді; підтримка системи контролю версій Git, що забезпечує можливість відстеження змін у коді та співпраці з іншими розробниками; велика кількість розширень для різних мов програмування та фреймворків.

### Мова програмування

Для реалізації алгоритмів обрано мову програмування C++. Цей вибір обґрунтований наступними перевагами C++ у контексті криптографічних застосувань: висока швидкість виконання програм завдяки компілюванню в машинний код, що особливо важливо для криптографічних операцій; ефективне управління пам'яттю, що дозволяє оптимізувати використання ресурсів при роботі з великими обсягами даних; багата стандартна бібліотека STL (Standard Template Library), яка надає контейнери vector, array, map для ефективної роботи з матрицями та таблицями; строга типізація, що зменшує ймовірність помилок під час розробки криптографічних алгоритмів; можливість низькорівневого програмування для оптимізації критичних ділянок коду.

### Додаткові бібліотеки

Для розширення функціональності програми використовуються наступні C++ бібліотеки та компоненти. Стандартна бібліотека STL забезпечує ефективну роботу з контейнерами vector для створення двовимірних масивів та матриць, що особливо важливо для реалізації шифру «Магічний квадрат». Заголовок <iostream> використовується для введення-виведення даних, <string>.

## Вимоги до апаратного та програмного забезпечення

Для успішної розробки, тестування та експлуатації програмного продукту, що реалізує обрані криптографічні алгоритми, необхідно забезпечити відповідність певним мінімальним вимогам до апаратного та програмного забезпечення.

### Апаратні вимоги

Мінімальні вимоги до апаратного забезпечення є досить помірними, оскільки реалізовані алгоритми не потребують значних обчислювальних ресурсів. Процесор повинен мати тактову частоту не менше 1 ГГц, при цьому підтримуються як Intel, так і AMD архітектури. Рекомендується використання двоядерного процесора для більш комфортної роботи із середовищем розробки.

Оперативна пам'ять повинна становити не менше 4 ГБ для комфортної роботи операційної системи, компілятора C++ та середовища розробки.

Вільне місце на жорсткому диску має становити не менше 2 ГБ для встановлення компілятора C++, середовища розробки, необхідних бібліотек та зберігання файлів проекту. Рекомендується мати додатково 3-5 ГБ вільного місця для тимчасових файлів компіляції та комфортної роботи операційної системи.

Монітор з роздільною здатністю не менше 1024×768 пікселів забезпечить адекватне відображення графічного інтерфейсу програми. Рекомендується роздільна здатність 1920×1080 або вища для більш комфортної роботи з кодом.

Мінімальні вимоги до програмного забезпечення:

Операційна система – Microsoft Windows 7.