**ЖИТОМИРСЬКИЙ АГРОТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ**

**ЦИКЛОВА КОМІСІЯ СПЕЦІАЛЬНОСТІ**

**«КОМП’ЮТЕРНІ НАУКИ»**

**КУРСОВА РОБОТА**

з дисципліни «Алгоритмізація та програмування» на тему:

«Розробка програми для обліку викладачів відділення "Електрифікація та інформаційні системи".»

Студента \_2\_ курсу \_\_П-21\_\_\_\_ групи

Галузь знань 12 « Інформаційні технології»

спеціальності 122 «Комп’ютерні науки та інформаційні технології»

Рудюк Мар`ян Миколайович

(прізвище та ініціали)

Керівник Устименко Л.М.

Національна шкала

Кількість балів: \_\_\_\_\_\_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Устименко Л.М.

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

м. Житомир – 2023 рік

**ЖИТОМИРСЬКИЙ АГРОТЕХНІЧНИЙ КОЛЕДЖ**

**Відділення** Електрифікації та інформаційні системи

**Циклова комісія** **спеціальності** «Комп’ютерні науки»

**Освітньо-кваліфікаційний рівень** Молодший спеціаліст

**Галузь знань** 12 « Інформаційні технології»

**Спеціальність** 122 «Комп’ютерні науки та інформаційні технології»

**ЗАТВЕРДЖУЮ**

**Голова циклової комісії**

**\_\_\_\_\_\_\_\_\_\_\_\_** Можаровський С.В..

“ ” січень 2023 року

**З А В Д А Н Н Я**

**НА КУРСОВУ РОБОТУ СТУДЕНТУ**

Рудюк Мар`ян Миколайович

(прізвище, ім’я, по батькові)

1. Тема роботи: **«Розробка програми для обліку викладачів відділення "Електрифікація та інформаційні системи".»**

керівник роботи **Устименко Людмила Миколаївна**

( прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “\_\_\_”\_\_\_\_\_\_\_\_\_\_20\_\_ року №\_\_

2. Строк подання студентом проекту (роботи) **22 травня 2023 року**

3. Вихідні дані до проекту (роботи)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Дата видачі завдання  **20 січня 2023 р.**

**КАЛЕНДАРНИЙ ПЛАН**

| №  з/п | Назва етапів курсового  проекту (роботи) | Строк виконання етапів проекту  ( роботи ) | Примітка |
| --- | --- | --- | --- |
|  | Вступ. Аналіз предметної області. | січень |  |
|  | Пошук потрібної інформації. | лютий |  |
|  | Оцінювання знайденої інформації. | березень |  |
|  | Розробка плану програми. | квітень |  |
|  | Написання основного коду програми. | квітень |  |
|  | Організація функцій. | травень |  |
|  | Організація динамічного виділення пам’яті та запису даних у файл. | червень |  |
|  | Завершення розробки програми. Висновки. | червень |  |

**Студент**  Рудюк М.М.

( підпис ) (прізвище та ініціали)

**Керівник проекту (роботи) \_\_\_\_\_\_\_\_\_\_\_**  Устименко Л.М.

( підпис ) (прізвище та ініціали



**Зміст**

Вступ………………………………………………………………….…….7

1. Постановка задачі ……………………………………………………….8
2. Довідка для програміста…………………………..………………………
   1. Вибір методу розв’язання………..………………….…..……13
   2. Структура програми…………………………………….….….15
   3. Алгоритм…………………………………………………….....16
3. Довідка для користувача………………………………………….……17
   1. Мінімальні вимоги до системи……………………………….17
   2. Тестування програми …………………………………………18

Висновки……………………………………………………………….…21

Використана література……………………………………………….…22

Додаток А. Код програми…………………………………………………24

## Вступ:

Розвиток інформаційних технологій відкриває нові можливості для автоматизації різноманітних процесів в усіх сферах діяльності людини. Однією з таких сфер є освіта. Для ефективної роботи навчального закладу потрібно мати якісний інструментарій для обліку викладачів. Тому тема розробки програми для обліку викладачів відділення є актуальною та потребує розробки.

**Мета та завдання курсової роботи:** Метою курсової роботи є розробка програми для асинхронного шифрування даних.

***Об’єкт*** дослідження:Шифрування даних асинхронним алгоритмом .

***Предмет*** дослідження:Побудова асиметричного алгоритму шифрування даних.

***Метод*** дослідження: аналітичний.

**1. Постановка задачі:**

1. програмний продукт може бути у вигляді консольної програми

2. програма має здійснювати збереження тексту в файл, завантаження

тексту з файлу

3. програма повинна містити меню користувача

4. код програми має містити функції користувача

5. програма має здійснювати шифрування та дешифрування тексту

**Таблиця 1.1. Основні функції**

| **Назва функції** | **Опис** | **Призначення (результат роботи)** |
| --- | --- | --- |
| bool isPrime(int num) | Функція для генерації випадкового простого числа | Функція яка допомагає знайти нам випадкові числа “p” і “q” які в подальшому допомагають виконувати саме шифрування. |
| int gcd(int a, int b) | Функція для знаходження НСД (найбільшого спільного дільника) | Функція знаходження найбільшого спільного дільника. |
| int findCoPrime(int eulerPhi) | Функція для знаходження взаємно простого числа | Функція знаходження взаємного простого числа яке ми в подалі будемо використовувати в формулі Ейлера. |
| int encrypt(int message, int e, int n) | Функція шифрування повідомлення | Практично основа всієї програми в якій ми використовуємо формулу Ейлера для самого асинхронного шифрування даних |
| intdecrypt(int encryptedMessage, int d, int n) | Функція для дешифрування шифрованого повідомлення | Частина програми в якій виконується дешифрування даних з повним урахуванням перед цим шифрованих даних |
| void saveToFile(const std::string& filename, const std::string& text) | Функція дозволяюча нам робити збереження шифрованих і дешифрованих даних у файл | Завдяки частині цієї програми ми можемо виконувати збереження результатів програми в файлах формату на вибір |
| std::string loadFromFile(const std::string& filename) | Функція для завантаження тексту з файлу | А завдяки цій частині ми можемо відкрити файл з потрібними нам даними для шифрування |
| int main() | Функція в якій виконується вся “магія” а саме буквально весь процес шифрування і взаємодії з меню | Сама функція містить:генерацію простих чисел,обчислення функції Ейлера,знаходження простого числа e,знаходження числа d і саму взаємодію в вигляді меню користувача |

Основні дані описані у табл. 1.2.

**Таблиця 1.2. Основні дані**

| **Назва змінної** | **Тип** | **Призначення** |
| --- | --- | --- |
| sqrt Num | int | Використовується для зберігання цілих чисел, таких як прості числа, ключі, розміри тощо. |
| isPrime | bool | Використовується для повернення логічного значення, наприклад, у функції isPrime, яка перевіряє, чи є число простим. |
| filename,text | string | Функції які відіграють роль в збереженні тексту у файл |
| loadFromFile | string | завантажує текст з файлу |
| teacher.посада | string | Зберігає посаду викладача. |
| teachers | vector | зберігає список викладачів. |
| file | ifstream | використовується для зчитування даних з файлу "teachers.txt". |
| file | ofstream | використовується для запису даних у файл "teachers.txt". |
| file(filename) | std::ofstream і std::ifstream: | Використовуються для створення об'єктів файлів для запису та читання даних з файлів |
| std::cout і std::cin: | std::cout і std::cin: | Використовуються для виводу повідомлень на екран і зчитування введених користувачем значень з клавіатури |
| std::cerr: | std::cerr: | Використовується для виведення помилок |
| ss(message) | std::stringstream | Використовується для зчитування та обробки рядків, які містять зашифровані символи. |
| static\_cast<unsigned> | srand і rand: | Використовуються для генерації випадкових чисел для вибору простих чисел p і q. |
| time | time | Використовується для ініціалізації генератора випадкових чисел srand зі значенням часу. |
| Масиви символів | int | Використовуються для зберігання зашифрованих символів та розшифрованих символів. |
| Математичні дії | sqrt і pow | Використовуються для обчислення квадратного кореня та піднесення до степеня в функціях перевірки простоти чисел. |

**2.1. Вибір методу розв’язування**

Метод RSA (Rivest-Shamir-Adleman) є асиметричним алгоритмом шифрування, який використовує два ключі: публічний ключ для шифрування і приватний ключ для дешифрування. Основна ідея полягає в тому, що шифрування за допомогою публічного ключа є легким обчислювальним завданням, тоді як дешифрування зашифрованого повідомлення можливе тільки за допомогою приватного ключа, який залишається виключно у власника.

Переваги методу RSA:

- Безпека: RSA забезпечує високий рівень безпеки, оснований на складності факторизації великих простих чисел.

- Асиметричність: Використання двох ключів - публічного і приватного - забезпечує безпеку обміну повідомленнями без необхідності обміну секретним ключем.

- Широке застосування: RSA широко використовується для шифрування повідомлень, підпису даних, аутентифікації і забезпечення безпеки в мережах.

- Ефективність: RSA ефективно застосовується для шифрування невеликих об'ємів даних, таких як ключі шифрування і цифрові підписи.

Однак, метод RSA також має обмеження, пов'язані з обчислювальною складністю шифрування і дешифрування великих об'ємів даних, що призводить до використання симетричних алгоритмів для шифрування фактичних повідомлень. Також, RSA вразливий до атак з використанням падаючих атак та атак з використанням публічного ключа. Тому, для забезпечення повної безпеки, використовуються комбінації RSA з іншими алгоритмами шифрування і підпису даних.

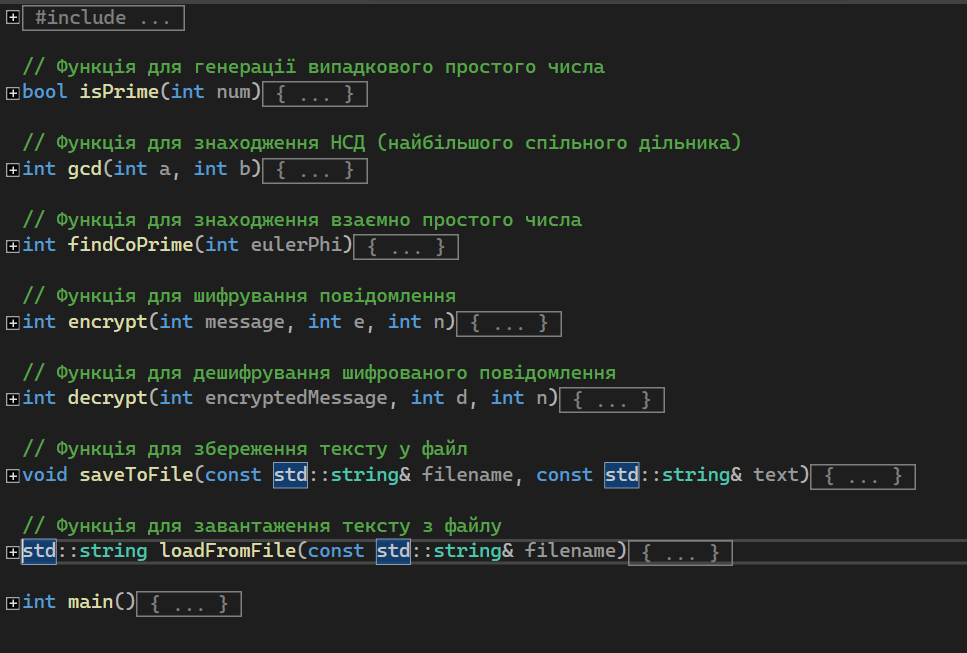
Ця програма реалізує алгоритм RSA для шифрування та дешифрування повідомлень. Вона має меню, яке дозволяє користувачеві вибрати операцію: шифрування повідомлення, дешифрування зашифрованого повідомлення, збереження тексту у файл та завантаження тексту з файлу.

Будь ласка, зверніть увагу, що цей код лише демонстраційний і не має високої стійкості до криптоаналізу. Для реальних застосувань рекомендується використовувати бібліотеки криптографічних функцій, які мають вбудовану підтримку RSA.

## 2.2. Структура програми

Структура проекту наведена на рисунках:

Рис. 2.2.1. Структура програми



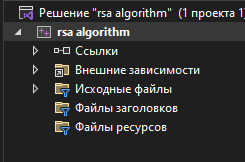


Рис 2.2.2. Структура програми

**2.3. Алгоритм**

.1.Генерація ключів: Генерується два великі прості числа, `p` і `q`. Потім обчислюються модуль `n = p \* q` і функція Ейлера від `n`, яка використовується для знаходження взаємно простого числа `e`. На основі `e` і функції Ейлера обчислюється число `d`, яке є мультиплікативно оберненим до `e` по модулю функції Ейлера. Отримані значення `e` і `n` становлять публічний ключ, а значення `d` і `n` - приватний ключ.

2. Шифрування: Повідомлення розбивається на блоки (наприклад, символи або байти), і кожен блок шифрується окремо. Кожен блок перетворюється у числове значення, і застосовується операція шифрування: `encryptedChar = (message^e) % n`, де `message` - числове значення блоку, `e` - публічний ключ і `n` - модуль.

3. Дешифрування: Зашифровані блоки дешифруються за допомогою приватного ключа: `decryptedChar = (encryptedChar^d) % n`, де `encryptedChar` - зашифроване числове значення блоку, `d` - приватний ключ і `n` - модуль. Результат дешифрування є числовим значенням оригінального блоку.

4. Збереження і завантаження повідомлень: Програма надає можливість зберегти шифровані або дешифровані повідомлення у файл і завантажити їх з файлу для подальшого використання.

**3.1 Мінімальні вимоги до системи**

Задля того, щоб запустити програму, Вам потрібно мати комп’ютер зі встановленою Visual Studio (версія не важлива) з попередньо встановленим пакетом (рисунок).   
 Рис 3.1.1 Характеристики мого компютера

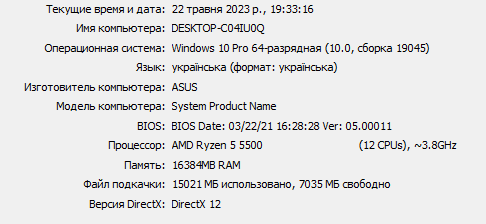
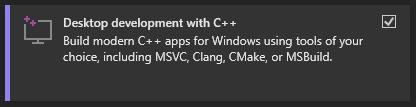


Рис 3.1.2. Потрібний пакет



**3.2. Тестування програми**

Тестування програми показано на рисунках:

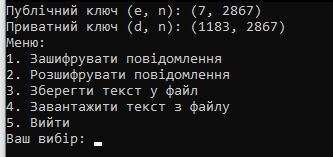


Рис 3.2.1. Тестування програми

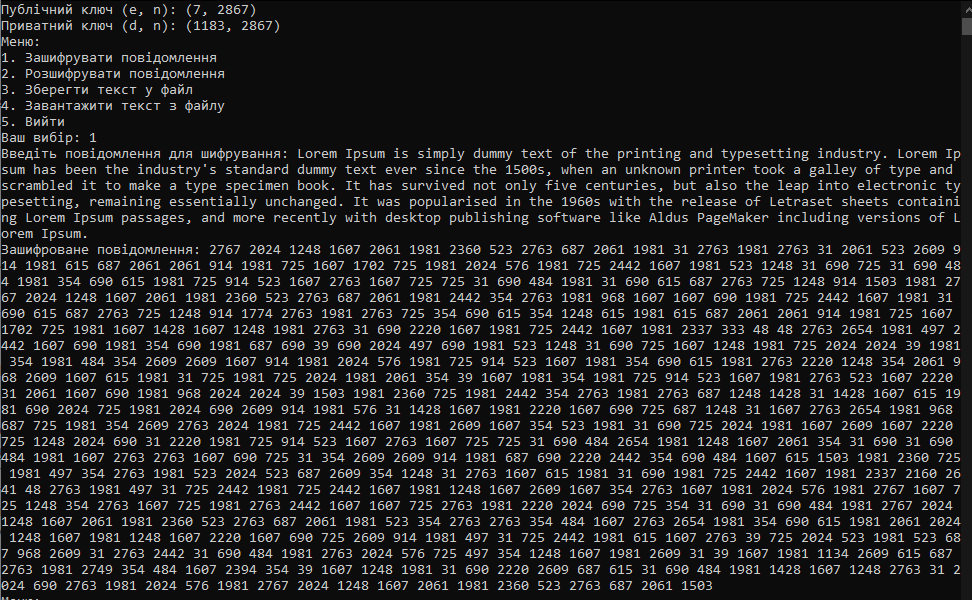


Рис 3.2.2. Тестування програми

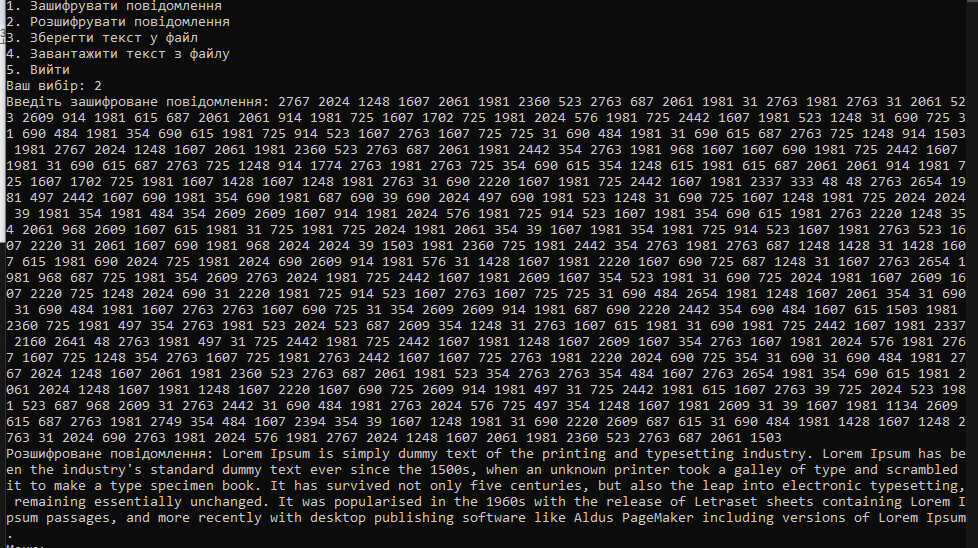


Рис 3.2.3. Тестування програми

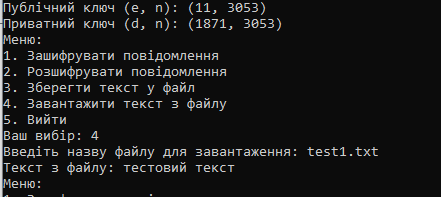


Рис 3.2.4. Тестування програми

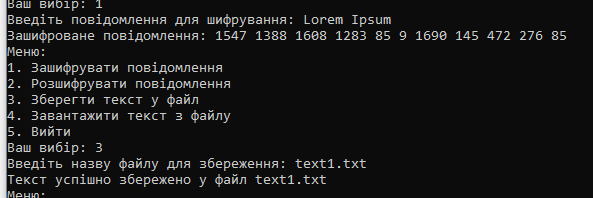
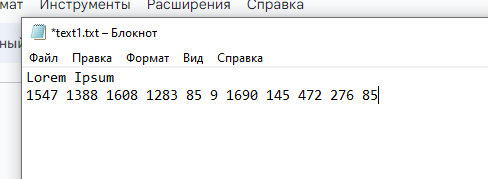


Рис 3.2.5. Тестування програми



**Висновки:** У ході даної курсової роботи було реалізовано програму для асинхронного шифрування і розшифрування даних"Електрифікація та інформаційні системи". Також було вдосконалено знання мови C++, особливо знання функцій, їх реалізація та інше.Вивчено особливості створення звіту з курсових робіт.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1.C++ Reference - https://en.cppreference.com/ - Довідник C++.

2.C++ Standard - https://isocpp.org/ - Стандарт C++.

3.cppreference.com - https://en.cppreference.com/ - Довідник C++.

4.Stack Overflow - https://stackoverflow.com/questions/tagged/c%2B%2B - Вопроси та відповіді C++.

5.LearnCpp.com - https://www.learncpp.com/ - Навчання C++.

6.CPlusPlus.com - http://www.cplusplus.com/ - Джерело інформації C++.

7.C++ Primer - https://www.amazon.com/Primer-5th-Stanley-B-Lippman/dp/0321714113 - Книга "C++ Primer".

8.ISO C++ Foundation - https://isocpp.org/ - Фонд C++.

9.GitHub - https://github.com/ - Сховище проектів.

10.Codecademy - https://www.codecademy.com/learn/learn-cpp - Онлайн-курс C++.

ДОДАТКИ

**Додаток А. Код програми**

#include <iostream>

#include <fstream>

#include <string>

#include <cstdlib>

#include <ctime>

#include <cmath>

#include <sstream>

#include<Windows.h>

#include<locale>

// Функція для генерації випадкового простого числа

bool isPrime(int num) {

setlocale(LC\_CTYPE, "ukr");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

if (num <= 1)

return false;

if (num == 2)

return true;

if (num % 2 == 0)

return false;

int sqrtNum = sqrt(num);

for (int i = 3; i <= sqrtNum; i += 2) {

if (num % i == 0)

return false;

}

return true;

}

// Функція для знаходження НСД (найбільшого спільного дільника)

int gcd(int a, int b) {

while (b != 0) {

int temp = b;

b = a % b;

a = temp;

}

return a;

}

// Функція для знаходження взаємно простого числа

int findCoPrime(int eulerPhi) {

int e = 2;

while (e < eulerPhi) {

if (gcd(eulerPhi, e) == 1)

return e;

e++;

}

return -1; // В разі невдачі повертаємо -1

}

// Функція для шифрування повідомлення

int encrypt(int message, int e, int n) {

int encryptedMessage = 1;

for (int i = 0; i < e; i++) {

encryptedMessage = (encryptedMessage \* message) % n;

}

return encryptedMessage;

}

// Функція для дешифрування шифрованого повідомлення

int decrypt(int encryptedMessage, int d, int n) {

int decryptedMessage = 1;

for (int i = 0; i < d; i++) {

decryptedMessage = (decryptedMessage \* encryptedMessage) % n;

}

return decryptedMessage;

}

// Функція для збереження тексту у файл

void saveToFile(const std::string& filename, const std::string& text) {

setlocale(LC\_CTYPE, "ukr");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

std::ofstream file(filename);

if (file.is\_open()) {

file << text;

file.close();

std::cout << "Текст успішно збережено у файл " << filename << std::endl;

}

else {

std::cerr << "Помилка при відкритті файлу " << filename << std::endl;

}

}

// Функція для завантаження тексту з файлу

std::string loadFromFile(const std::string& filename) {

std::ifstream file(filename);

std::string text;

if (file.is\_open()) {

getline(file, text);

file.close();

return text;

}

else {

std::cerr << "Помилка при відкритті файлу " << filename << std::endl;

return "";

}

}

int main() {

// Генеруємо два великі прості числа p і q

int p, q;

srand(static\_cast<unsigned>(time(0)));

do {

p = rand() % 100 + 1;

} while (!isPrime(p));

do {

q = rand() % 100 + 1;

} while (!isPrime(q));

// Обчислюємо n і функцію Ейлера від n

int n = p \* q;

int eulerPhi = (p - 1) \* (q - 1);

// Знаходимо взаємно просте число e

int e = findCoPrime(eulerPhi);

if (e == -1) {

std::cerr << "Помилка: не вдалося знайти взаємно просте число" << std::endl;

return 1;

}

// Знаходимо число d, яке є мультиплікативно оберненим до e по модулю eulerPhi

int d = 0;

while ((d \* e) % eulerPhi != 1) {

d++;

}

std::cout << "Публічний ключ (e, n): (" << e << ", " << n << ")" << std::endl;

std::cout << "Приватний ключ (d, n): (" << d << ", " << n << ")" << std::endl;

// Меню користувача

int choice;

std::string message;

do {

std::cout << "Меню:" << std::endl;

std::cout << "1. Зашифрувати повідомлення" << std::endl;

std::cout << "2. Розшифрувати повідомлення" << std::endl;

std::cout << "3. Зберегти текст у файл" << std::endl;

std::cout << "4. Завантажити текст з файлу" << std::endl;

std::cout << "5. Вийти" << std::endl;

std::cout << "Ваш вибір: ";

std::cin >> choice;

switch (choice) {

case 1:

std::cout << "Введіть повідомлення для шифрування: ";

std::cin.ignore();

std::getline(std::cin, message);

{

std::string encryptedMessage;

for (char c : message) {

int encryptedChar = encrypt(c, e, n);

encryptedMessage += std::to\_string(encryptedChar) + " ";

}

std::cout << "Зашифроване повідомлення: " << encryptedMessage << std::endl;

}

break;

case 2:

std::cout << "Введіть зашифроване повідомлення: ";

std::cin.ignore();

std::getline(std::cin, message);

{

std::string decryptedMessage;

std::stringstream ss(message);

std::string encryptedCharStr;

while (getline(ss, encryptedCharStr, ' ')) {

int encryptedChar = std::stoi(encryptedCharStr);

int decryptedChar = decrypt(encryptedChar, d, n);

decryptedMessage += static\_cast<char>(decryptedChar);

}

std::cout << "Розшифроване повідомлення: " << decryptedMessage << std::endl;

}

break;

case 3:

std::cout << "Введіть назву файлу для збереження: ";

std::cin.ignore();

std::getline(std::cin, message);

saveToFile(message, message);

break;

case 4:

std::cout << "Введіть назву файлу для завантаження: ";

std::cin.ignore();

std::getline(std::cin, message);

{

std::string loadedText = loadFromFile(message);

std::cout << "Текст з файлу: " << loadedText << std::endl;

}

break;

case 5:

std::cout << "До побачення!" << std::endl;

break;

default:

std::cout << "Невірний вибір. Спробуйте ще раз." << std::endl;

break;

}

} while (choice != 5);

return 0;

}