Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Информационные сети. Основы безопасности

ОТЧЕТ

к лабораторной работе №2

на тему

**ИДЕНТИФИКАЦИЯ И АУТЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЕЙ. ПРОТОКОЛ KERBEROS.**

Студент В. М. Вергасов

Преподаватель Е. А. Лещенко

Минск 2024

СОДЕРЖАНИЕ

[Введение 3](#_Toc157471308)

[1 Результат выполнения 4](#_Toc157471309)

[Заключение 5](#_Toc157471310)

[Приложение А (обязательное) Листинг кода 6](#_Toc157471311)

[Приложение Б (обязательное) Блок-схема алгоритма 8](#_Toc157471312)

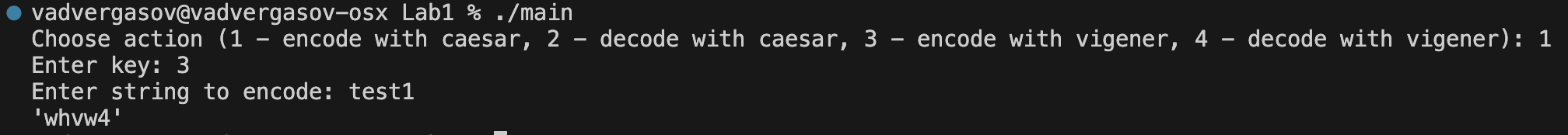
**ВВЕДЕНИЕ**

Лабораторная работа ставит перед собой задачу разработки программных средств для шифрования и дешифрования текстовых файлов с использованием двух классических методов шифрования: шифра Цезаря и шифра Виженера.

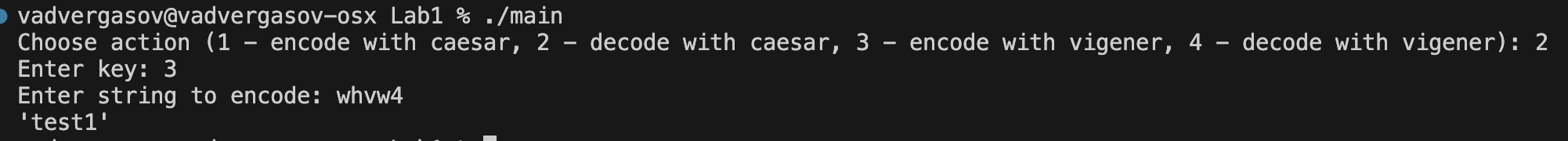
Основная цель работы – изучение принципов работы указанных алгоритмов и их реализация на языке программирования *C++*. В ходе работы будет осуществлен анализ методов шифрования, разработка алгоритмов шифрования и дешифрования, а также создание программного продукта, позволяющего осуществлять шифрование и дешифрование текстовых файлов.

1. **РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ**

В результате работы было создано два приложения для шифрования и дешифрования текстовых файлов: с использованием шифра Цезаря   
(рисунок 1) и шифра Виженера (рисунок 2).



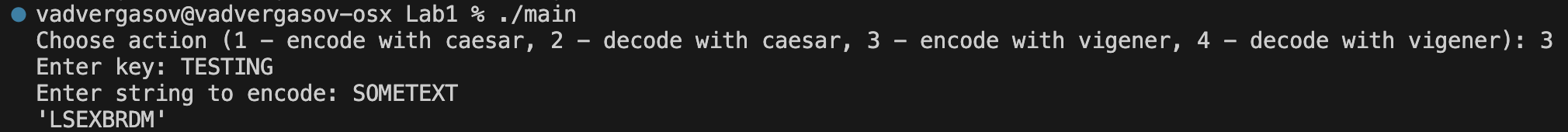
а



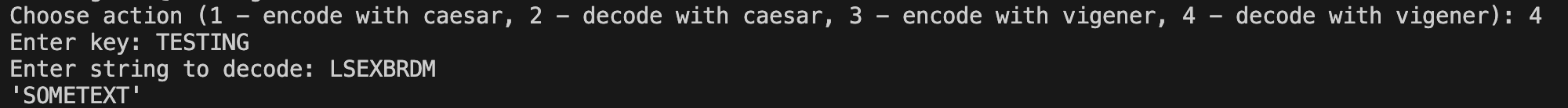
б

а – шифрование; б – дешифрование

Рисунок 1 – Шифр Цезаря



а



б

а – шифрование; б – дешифрование

Рисунок 2 – Шифр Виженера

# ЗАКЛЮЧЕНИЕ

В результате выполнения данной лабораторной работы были разработаны программные средства для шифрования и дешифрования текстовых файлов с применением шифра Цезаря и шифра Виженера на языке программирования *C++*.

Был проведен анализ методов шифрования и алгоритмов работы с текстовыми файлами, разработана блок-схема алгоритма, написаны и отлажены программы для шифрования и дешифрования файлов.

При выполнении работы были использованы теоретические сведения о принципах работы шифра Цезаря и шифра Виженера и получены практические навыки работы с данными алгоритмами.

# ПРИЛОЖЕНИЕ А (обязательное) Листинг кода

Листинг 1 – Файл *main.cpp*

#include <string>

#include <iostream>

#include <limits>

#include <cctype>

using namespace std;

void encode\_caesar();

void decode\_caesar();

void encode\_vigener();

void decode\_vigener();

void menu() {

cout << "Choose action (1 - encode with caesar, 2 - decode with caesar, 3 - encode with vigener, 4 - decode with vigener): ";

size\_t number;

cin >> number;

switch (number) {

case 1:

encode\_caesar();

break;

case 2:

decode\_caesar();

break;

case 3:

encode\_vigener();

break;

case 4:

decode\_vigener();

break;

case 5:

break;

default:

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

menu();

break;

}

}

void encode\_caesar() {

string input;

size\_t key;

cout << "Enter key: ";

cin >> key;

while (std::cin.fail()) {

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

cout << cin.fail() << endl;

cout << "Incorrect key. Enter key again: ";

cin >> key;

cout << cin.fail() << endl;

}

cout << "Enter string to encode: ";

cin >> input;

for (auto& symbol : input) {

if (!isalnum(symbol)) {

cout << "Invalid string!";

menu();

}

symbol += key;

if (isalpha(symbol - key)) {

symbol = ((symbol % ((islower(symbol - key) ? 'z' : 'Z') + 1)) + (islower(symbol - key) ? 'a' : 'A')) % (islower(symbol - key) ? 'a' : 'A') + (islower(symbol - key) ? 'a' : 'A');

} else {

symbol = ((symbol % ('9' + 1)) + '0') % '0' + '0';

}

}

cout << '\'' << input << '\'' << endl;

}

void decode\_caesar() {

string input;

size\_t key;

cout << "Enter key: ";

cin >> key;

while (std::cin.fail()) {

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

cout << cin.fail() << endl;

cout << "Incorrect key. Enter key again: ";

cin >> key;

cout << cin.fail() << endl;

}

cout << "Enter string to encode: ";

cin >> input;

for (auto& symbol : input) {

if (!isalnum(symbol)) {

cout << "Invalid string!";

menu();

}

symbol -= key;

if (isalpha(symbol + key)) {

if (symbol < (islower(symbol + key) ? 'a' : 'A')) {

symbol += 26;

}

} else {

if (symbol < '0') {

symbol += 10;

}

}

}

cout << '\'' << input << '\'' << endl;

}

void encode\_vigener() {

string key, input;

cout << "Enter key: ";

cin >> key;

cout << "Enter string to encode: ";

cin >> input;

while (key.size() < input.size()) {

key += key;

}

string result = "";

for (int i = 0; i < input.size(); ++i) {

char plainChar = input[i];

char keyChar = key[i % key.size()];

if (isalpha(plainChar)) {

char base = isupper(plainChar) ? 'A' : 'a';

char encryptedChar = ((plainChar - base) + (keyChar - base)) % 26 + base;

result += encryptedChar;

} else {

result += plainChar;

}

}

cout << '\'' << result << '\'' << endl;

}

void decode\_vigener() {

string key, input;

cout << "Enter key: ";

cin >> key;

cout << "Enter string to decode: ";

cin >> input;

string plaintext;

int keyLength = key.length();

for (int i = 0; i < input.size(); ++i) {

char cipherChar = input[i];

char keyChar = key[i % keyLength];

if (isalpha(cipherChar)) {

char base = isupper(cipherChar) ? 'A' : 'a';

char decryptedChar = ((cipherChar - base) - (keyChar - base) + 26) % 26 + base;

plaintext += decryptedChar;

} else {

plaintext += cipherChar;

}

}

cout << '\'' << plaintext << '\'' << endl;

}

int main() {

menu();

return 0;

}

# ПРИЛОЖЕНИЕ Б (обязательное) Блок-схема алгоритма

