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 №1

на тему

**ШИФР ЦЕЗАРЯ. ШИФР ВИЖЕНЕРА**

Студент Т. П. Власенко

Преподаватель Е. А. Лещенко

Минск 2024

**СОДЕРЖАНИЕ**

[1 Постановка задачи 3](#_6127s8b6t8wy)

[2 Блок-схема](#_r4y5108cx8yu) алгоритма [4](#_r4y5108cx8yu)

[3 Результат выполнения лабораторной работы](#_jnfz1qbvv5j4) 7

[Выводы](#_ldsbh3e5c0xh) 8

[Приложение А (обязательное) Листинг кода](#_54xlhz7mfhe) 9

# 1 ПОСТАНОВКА ЗАДАЧИ

Целью выполнения данной лабораторной работы является разработка программных средств для шифрования и дешифрования текстовых файлов, используя метод Шифра Цезаря (также известного как шифр сдвига или код Цезаря) и метод шифрования Виженера.

# 2 БЛОК-СХЕМА АЛГОРИТМА

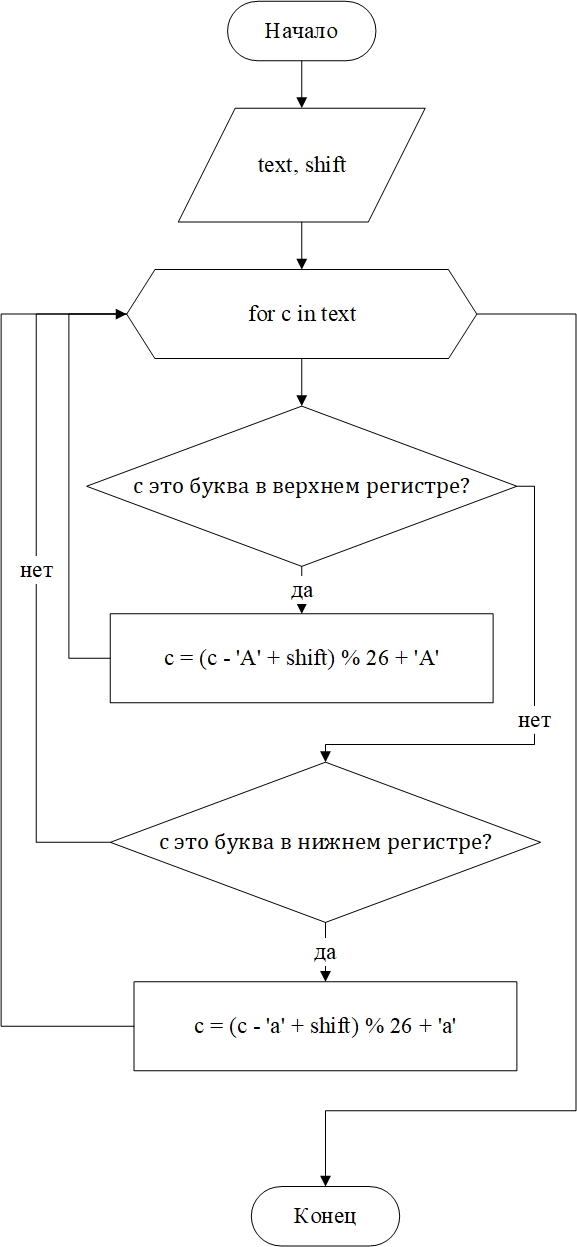


Рисунок 1 – Блок-схема функции *caesarEncrypt*

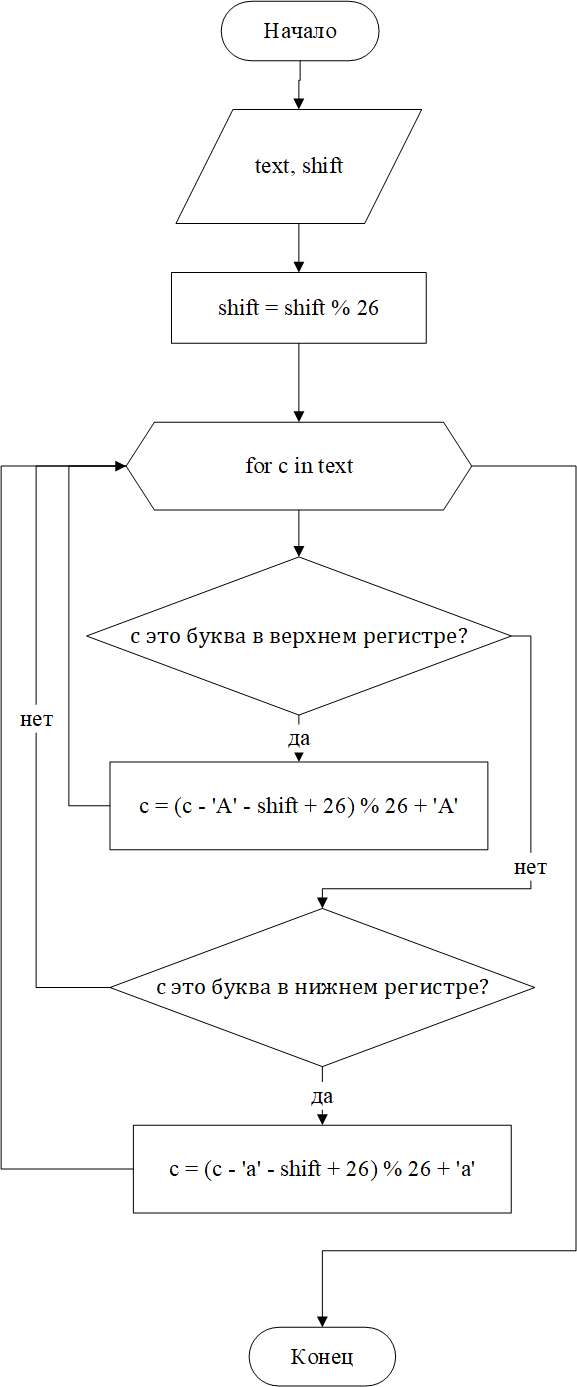


Рисунок 2 – Блок-схема функции *caesarDecrypt*

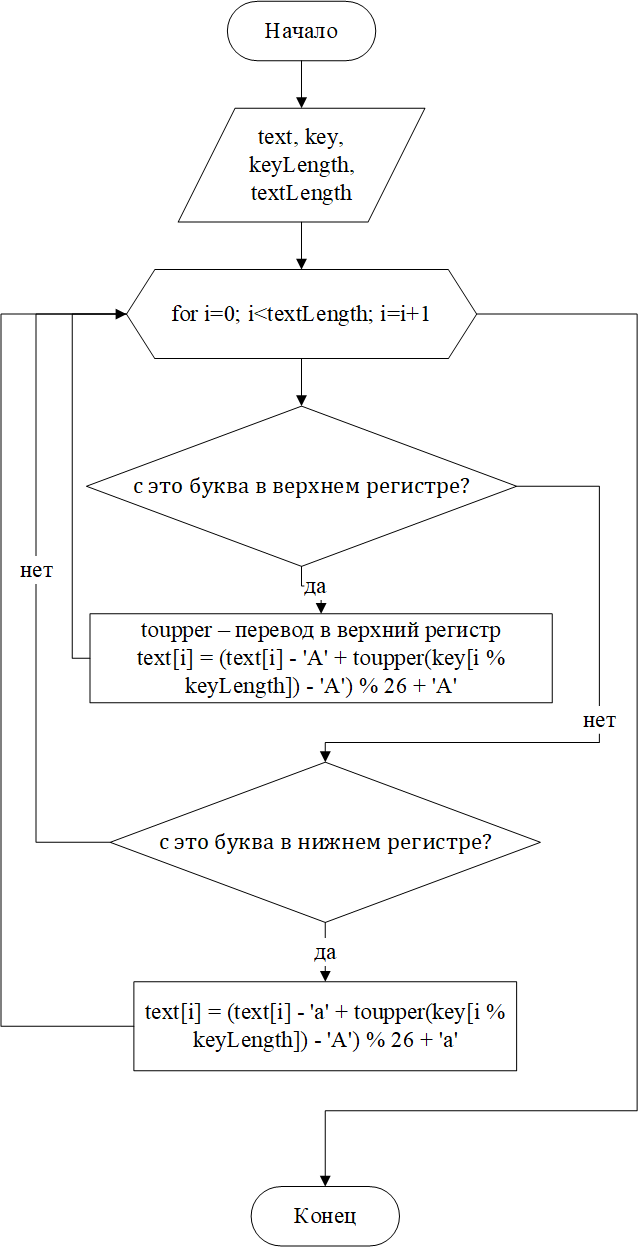


Рисунок 3 – Блок-схема функции *vigenereEncrypt*

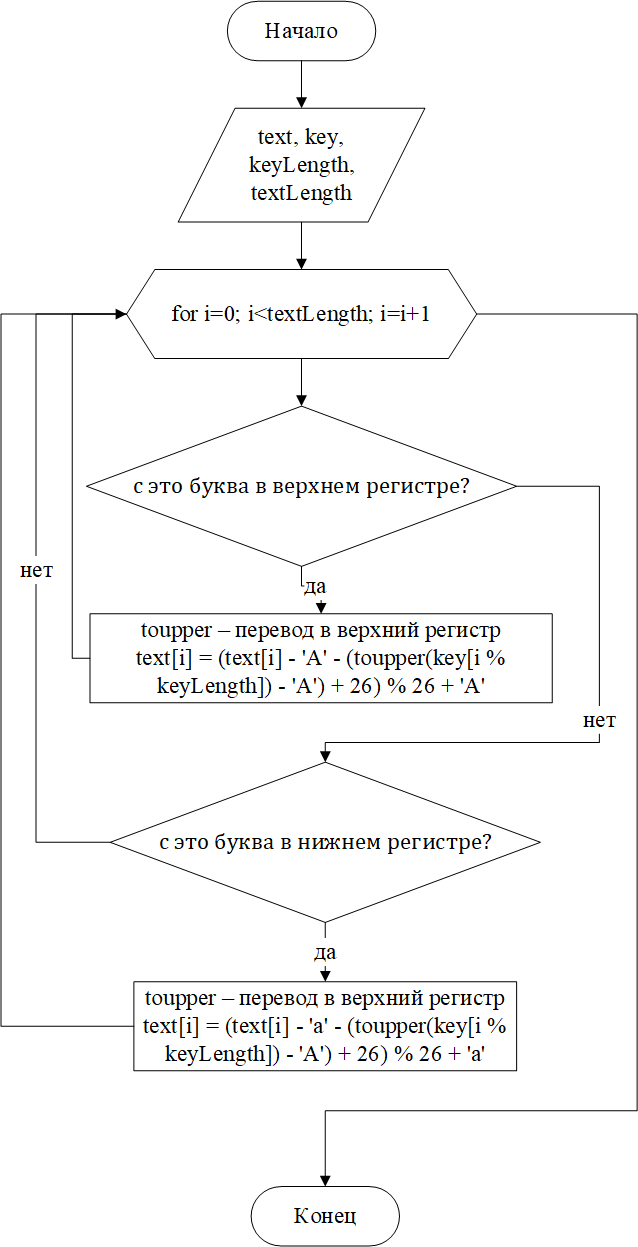


Рисунок 4 – Блок-схема функции *vigenereDecrypt*

# 3 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

Исходный текст файла для шифров – Abc def, knm.

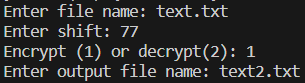


Рисунок 5 – Пример работы шифра «Цезаря»

Текст файла после применения шифра Цезаря – ZAB cde, jml.

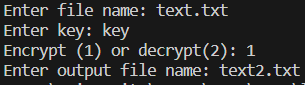


Рисунок 6 – Пример работы шифра «Виженера»

Текст файла после применения шифра Виженера – KFA hcp, urk.

# ВЫВОДЫ

Для лабораторной работы по теме “Шифр Цезаря” и “Шифр Виженера” был написан код на языке C++. Для шифра Цезаря были реализованы функции *caesarEncrypt* и *caesarDecrypt*, которые принимают строку и число *k* (сдвиг), и изменяют переданную строку, где каждая буква исходной строки сдвигается на k позиций в алфавите. Если символ не является буквой, он остается без изменений. Для шифра Виженера были реализованы аналогичные функции, которые принимают строку и ключ, который также является строкой, и изменяют переданную строку, где каждая буква исходной строки сдвигается на соответствующее букве ключа количество позиций в алфавите. Если символ не является буквой, он остается без изменений.

# ПРИЛОЖЕНИЕ А (обязательное) Листинг кода

#include <filesystem>

#include <fstream>

#include <iostream>

#include <string>

std::string readFile(const std::filesystem::path& filename) {

std::ifstream file(filename);

if (!file.is\_open()) {

throw std::invalid\_argument("Invalid file! " + filename.string());

}

std::string fileContents((std::istreambuf\_iterator<char>(file)),

std::istreambuf\_iterator<char>());

file.close();

return fileContents;

}

void writeToFile(const std::filesystem::path& filename,

const std::string& text) {

std::ofstream file(filename);

if (!file.is\_open()) {

throw std::invalid\_argument("Invalid file! " + filename.string());

}

file << text;

file.close();

}

#include <iostream>

#include "file.h"

void caesarEncrypt(std::string& text, int shift) {

for (char& c : text) {

if (std::isupper(c)) {

c = (c - 'A' + shift) % 26 + 'A';

} else if (std::islower(c)) {

c = (c - 'a' + shift) % 26 + 'a';

}

}

}

void caesarDecrypt(std::string& text, int shift) {

shift %= 26;

for (char& c : text) {

if (std::isupper(c)) {

c = (c - 'A' - shift + 26) % 26 + 'A';

} else if (std::islower(c)) {

c = (c - 'a' - shift + 26) % 26 + 'a';

}

}

}

int main() {

std::cout << "Enter file name: ";

std::filesystem::path path;

std::cin >> path;

path = std::filesystem::absolute(path);

std::string text = readFile(path);

std::cout << "Enter shift: ";

int shift;

std::cin >> shift;

if (shift < 0) {

std::cerr << "Invalid input!";

return 0;

}

std::cout << "Encrypt (1) or decrypt(2): ";

int operation;

std::cin >> operation;

switch (operation) {

case 1:

caesarEncrypt(text, shift);

break;

case 2:

caesarDecrypt(text, shift);

break;

default:

std::cerr << "Invalid input!";

return 0;

}

std::cout << "Enter output file name: ";

std::filesystem::path outPath;

std::cin >> outPath;

outPath = std::filesystem::absolute(outPath);

writeToFile(outPath, text);

}

#include <iostream>

#include "file.h"

void vigenereEncrypt(std::string& text, const std::string& key) {

int keyLength = key.size();

int textLength = text.size();

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

char& c = text[i];

if (std::isupper(c)) {

c = (c - 'A' + std::toupper(key[i % keyLength]) - 'A') % 26 + 'A';

} else if (std::islower(c)) {

c = (c - 'a' + std::toupper(key[i % keyLength]) - 'A') % 26 + 'a';

}

}

}

void vigenereDecrypt(std::string& text, const std::string& key) {

int keyLength = key.size();

int textLength = text.size();

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

char& c = text[i];

if (std::isupper(c)) {

c = (c - 'A' - (std::toupper(key[i % keyLength]) - 'A') + 26) % 26 + 'A';

} else if (std::islower(c)) {

c = (c - 'a' - (std::toupper(key[i % keyLength]) - 'A') + 26) % 26 + 'a';

}

}

}

int main() {

std::cout << "Enter file name: ";

std::filesystem::path path;

std::cin >> path;

path = std::filesystem::absolute(path);

std::string text = readFile(path);

std::cout << "Enter key: ";

std::string key;

std::cin >> key;

std::cout << "Encrypt (1) or decrypt(2): ";

int operation;

std::cin >> operation;

switch (operation) {

case 1:

vigenereEncrypt(text, key);

break;

case 2:

vigenereDecrypt(text, key);

break;

default:

std::cerr << "Invalid input!";

return 0;

}

std::cout << "Enter output file name: ";

std::filesystem::path outPath;

std::cin >> outPath;

outPath = std::filesystem::absolute(outPath);

writeToFile(outPath, text);

}