Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №1

по курсу «Защита информации в сети Internet»

на тему «Разработка программы шифрования/дешифрирования текста с использованием перестановочного шифра»

Выполнили  
студенты группы 18ВВ1:

Амплеев Д.О.  
Колобов К.А.

Приняли:

к.т.н., доцент Дубравин А.В.

к.т.н., доцент Карамышева Н.С.

2020

**Цель работы:** разработать программу шифрования/дешифрирования текста с использованием перестановочного шифра.

**Задания:** разработать программу, выполняющую шифрование и расшифровывание произвольного текстового файла с использованием перестановочного шифра используя в качестве ключа последовательность, соответствующую номеру варианта. Выполнить проверку путем двоичного сравнения исходного файла и фала, полученного после расшифровывания.

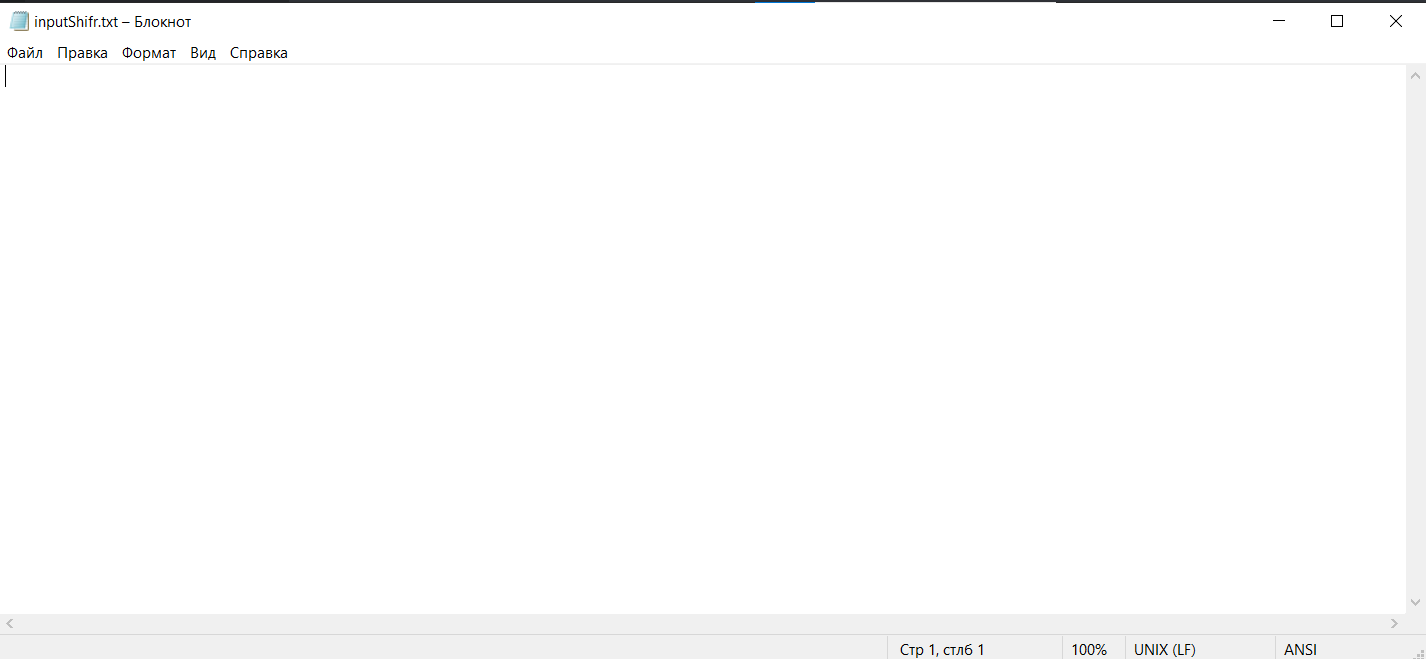
|  |  |
| --- | --- |
| Номер варианта | Последовательность перестановки |
| 1 | 3 6 5 2 4 1 7 9 8 10 |

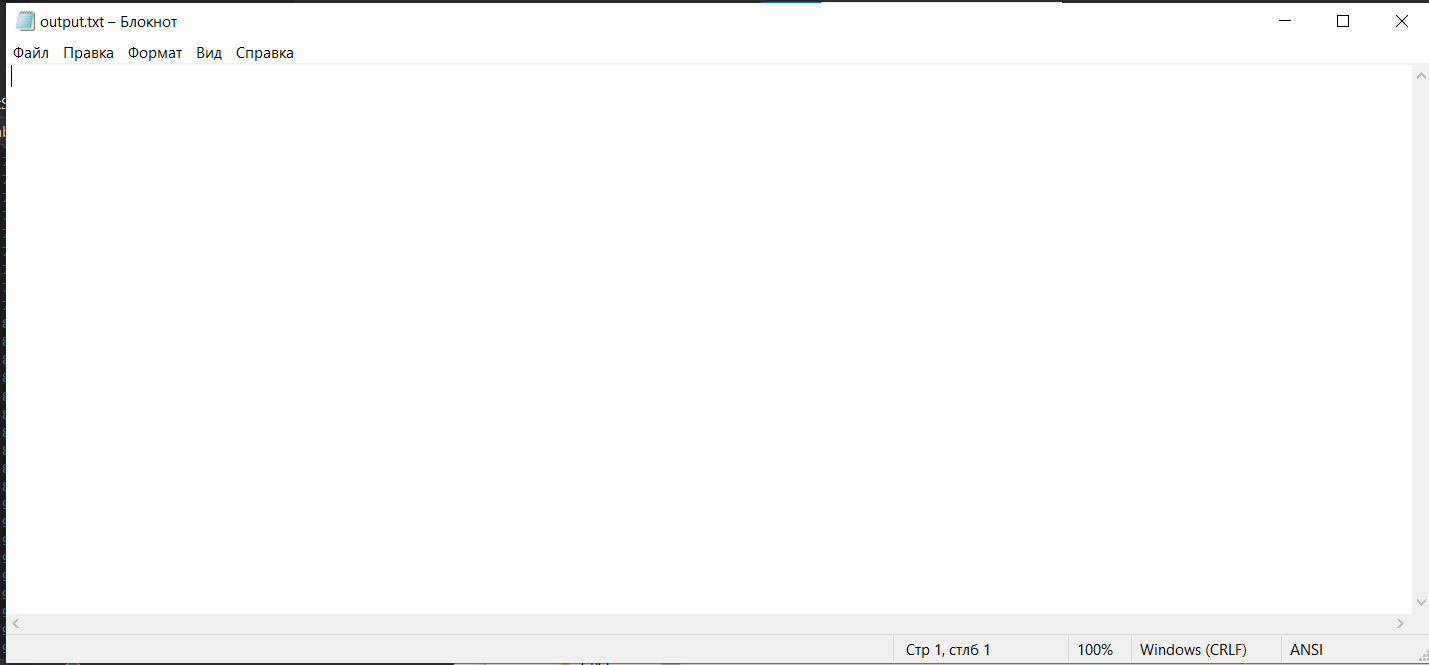
Работа программы:

Для осуществления шифрования и расшифровывания необходимо создать текстовый файл, где будет отображена последовательность перестановки. Также нужно создать файл, над которым будет проводиться шифрование.

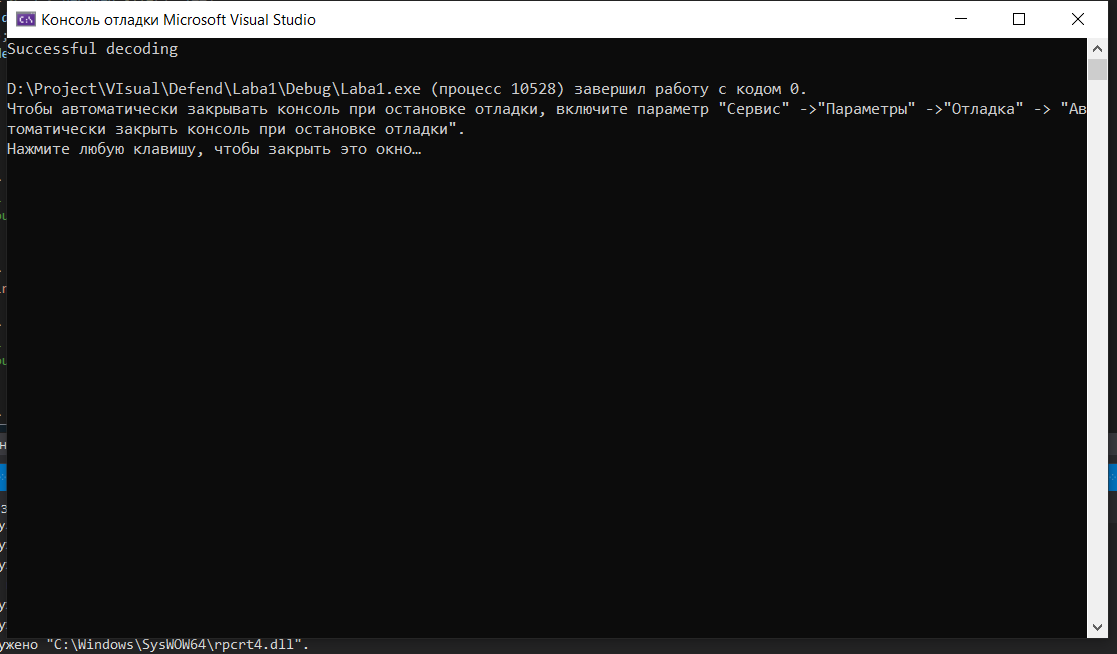
Для демонстрации работы программы создали следующий текстовый файл:







Вывод программы:





Листинг:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <string>

#include <fstream>

#include <clocale>

#include <vector>

using namespace std;

const int keySize = 10;

const int keyPosition[keySize] = { 2, 5, 4, 1, 3, 0, 6, 8, 7, 9 };

const int keySource[keySize] = { 5, 3, 0, 4, 2, 1, 6, 8, 7, 9 };

void init() {

setlocale(LC\_ALL, "ru");

}

vector<char> readText(const string& filename) {

ifstream fin(filename, ios::binary);

vector<char> text((istreambuf\_iterator<char>(fin)),

istreambuf\_iterator<char>());

fin.close();

return text;

}

void writeText(const string& filename, const vector<char>& text) {

ofstream fout("inputShifr.txt", ios::binary);

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

fout << text[i];

fout.close();

}

void encodeBlock(const vector<char>& source, vector<char>& destination, int start) {

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

destination[start + keyPosition[i]] = source[start + i];

}

}

vector<char> encodeString(const vector<char>& source) {

vector<char> encoded = source;

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

encodeBlock(source, encoded, i);

}

return encoded;

}

void decodeBlock(const vector<char>& source, vector<char>& destination, int start) {

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

destination[start + keySource[i]] = source[start + i];

}

}

vector<char> decodeString(const vector<char>& source) {

vector<char> decoded = source;

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

decodeBlock(source, decoded, i);

}

ofstream fout("output.txt", ios::binary);

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

fout << decoded[i];

fout.close();

return decoded;

}

int main() {

init();

vector<char> text = readText("vc142.pdb");

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

// cout << text[i];

//}

vector<char> encrypted = encodeString(text);

writeText("inputShifr.txt", encrypted);

vector<char> encryptedText = readText("inputShifr.txt");

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

// cout << encryptedText[i];

//}

vector<char> decodedText = decodeString(encryptedText);

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

// cout << decodedText[i];

//}

if (decodedText != text)

cout << "Unsuccessful decoding\n";

else

cout << "Successful decoding\n";

return 0;

}

Вывод: разработали программу шифрования/дешифрирования текста с использованием перестановочного шифра.