Министерство образования Российской Федерации  
Пензенский государственный университет  
Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №1  
по курсу «Методы защиты КС»

**Выполнили:**

студенты группы 17ВВ2

Кальнобрицкий К. В.

Аняев А. Р.

**Приняли:**

доц. Дубравин А. В.

к.т.н., доц. Карамышева Н. С.

Пенза, 2020

**Цель работы:** разработка программы шифрования/дешифрирования текста с использованием перестановочного шифра.

**Задание:** 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.

**Код программы:**

#include <iostream>

#include <fstream>

#include <cstdlib>

#include <string>

using namespace std;

struct block

{

int wholeBlock; //целый блок

int incompleteBlock; //неполный блок

};

string TextEncryption(char inputData[], int arrKey[], const int keyLength);

string TextDecryption(char inputData[], int arrKey[], const int keyLength);

void WriteToFile(const string& inputData, string directory);

void DeleteFileContents(string directory);

void WriteToFile(block& inputData);

void ReadToFile(block& inputData);

int main()

{

setlocale(LC\_ALL, "rus");

cout << "[-e] [-d] [диск1:][путь1]имя\_файла1 [диск2:][путь2]имя\_файла2\n"

<< "[имя\_файла1] [имя\_файла2] состоит из латинских символов\n"

<< "\n -e Зашифровать имя\_файла1 имя\_файла2"

<< "\n -d Расшифровать имя\_файла1 имя\_файла2\n\n";

cout << "Введите \"exit\" - для завершении работы программы\n\n";

const int N = 10;

int arrKey[N] = { 3, 9, 10, 5, 7, 1, 2, 8, 6, 4 };

char arrBuffer[N + 1];

string teamName;

do

{

cin >> teamName;

if (teamName.length() == 2)

{

if (teamName == "-e")

{

string fileName1, fileName2;

cin >> fileName1 >> fileName2;

/\* Шифрование \*/

ifstream finE;

finE.open(fileName1);

if (!finE.is\_open())

{

EXIT\_FAILURE;

}

DeleteFileContents(fileName2);

block b{};

do

{

finE.get(arrBuffer, (N + 1));

if ((int)finE.gcount())

{

for (int i = (int)finE.gcount(); i < N; i++)

{

arrBuffer[i] = ' ';

}

WriteToFile(TextEncryption(arrBuffer, arrKey, N), fileName2);

((int)finE.gcount() < N) ? b.incompleteBlock = (int)finE.gcount() : b.wholeBlock++;

}

} while (!finE.eof());

finE.close();

WriteToFile(b);

cout << "Операция завершилась успехом\n";

}

else if (teamName == "-d")

{

string fileName1, fileName2;

cin >> fileName1 >> fileName2;

/\* Расшифровывание \*/

ifstream finD;

finD.open(fileName1);

if (!finD.is\_open())

{

EXIT\_FAILURE;

}

DeleteFileContents(fileName2);

block b{};

ReadToFile(b);

do

{

finD.get(arrBuffer, (N + 1));

if (b.wholeBlock--)

{

WriteToFile(TextDecryption(arrBuffer, arrKey, N), fileName2);

if (!b.wholeBlock && b.incompleteBlock)

{

finD.get(arrBuffer, (N + 1));

string strTime = TextDecryption(arrBuffer, arrKey, N);

string strDecryption;

for (int i = 0; i < b.incompleteBlock; i++)

{

strDecryption += strTime[i];

}

WriteToFile(strDecryption, fileName2);

break;

}

}

} while (!finD.eof());

finD.close();

cout << "Операция завершилась успехом\n";

}

else

{

cout << "Ошибка ввода команды...\n";

}

}

else if (teamName != "exit")

{

cout << "Ошибка ввода команды...\n";

}

} while (teamName != "exit");

return 0;

}

/\* Шифрование \*/

string TextEncryption(char inputData[], int arrKey[], const int keyLength)

{

string textEncryption;

for (int i = 0; i < keyLength; i++)

{

textEncryption += inputData[arrKey[i] - 1];

}

return textEncryption;

}

/\* Расшифровывание \*/

string TextDecryption(char inputData[], int arrKey[], const int keyLength)

{

string textDecryption;

for (int i = 0; i < keyLength; i++)

{

int index = 0;

for (; arrKey[index] != (i + 1); index++);

textDecryption += inputData[index];

}

return textDecryption;

}

/\* Записать в файл \*/

void WriteToFile(const string& inputData, string directory)

{

ofstream fout;

fout.open(directory, ios\_base::app);

if (!fout.is\_open())

{

EXIT\_FAILURE;

}

fout << inputData;

fout.close();

}

/\* Удалить содержимое файла \*/

void DeleteFileContents(string directory)

{

ofstream fout;

fout.open(directory, ios\_base::trunc);

if (!fout.is\_open())

{

EXIT\_FAILURE;

}

fout.close();

}

/\* Запись блока в файл \*/

void WriteToFile(block& inputData)

{

ofstream fout;

fout.open("block.txt");

if (!fout.is\_open())

{

EXIT\_FAILURE;

}

fout << inputData.wholeBlock << " " << inputData.incompleteBlock;

fout.close();

}

/\* Чтение блока из файл \*/

void ReadToFile(block& inputData)

{

ifstream fin;

fin.open("block.txt");

if (!fin.is\_open())

{

EXIT\_FAILURE;

}

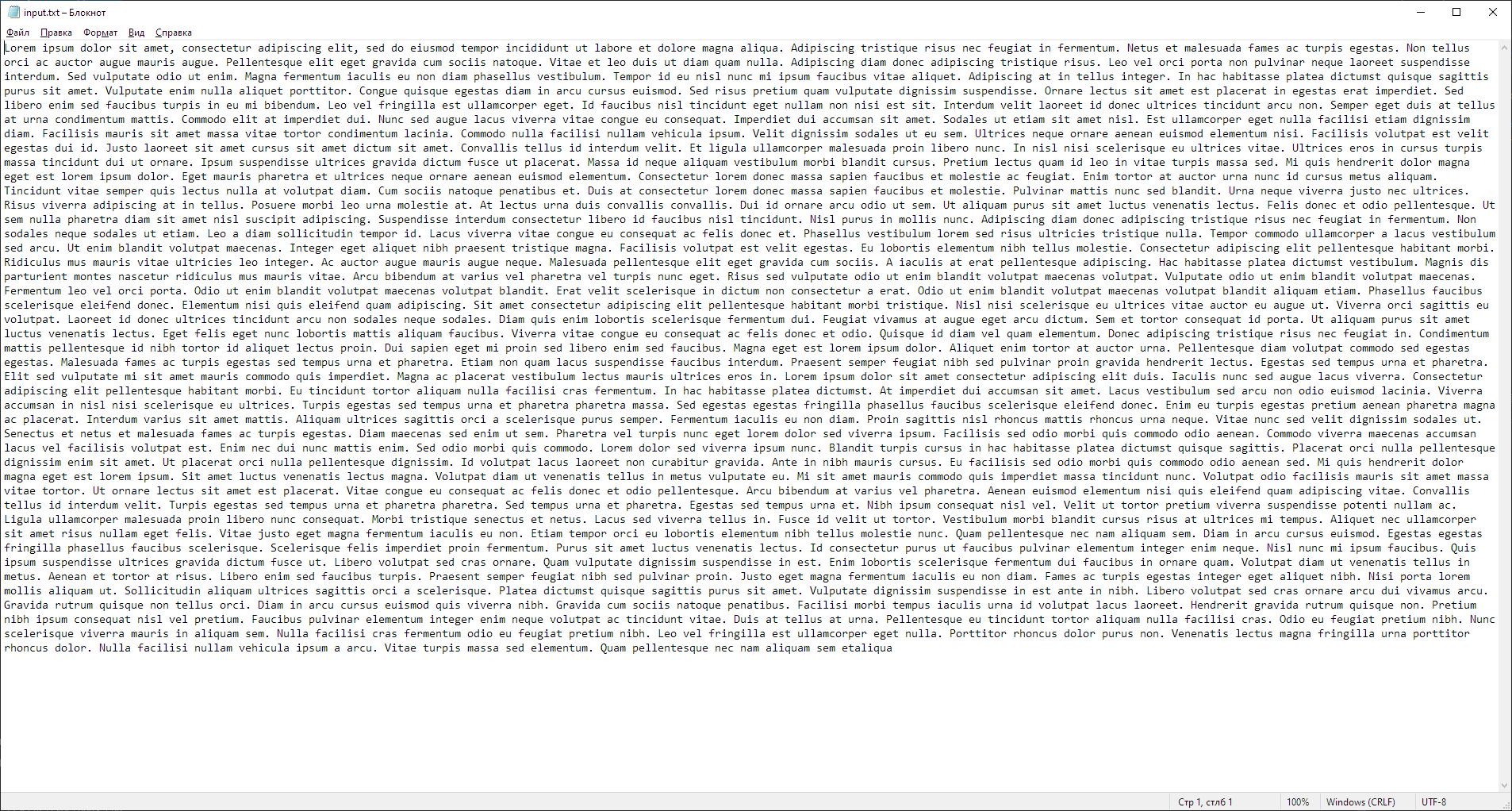
fin >> inputData.wholeBlock >> inputData.incompleteBlock;

fin.close();

}

**Результат работы программы:**

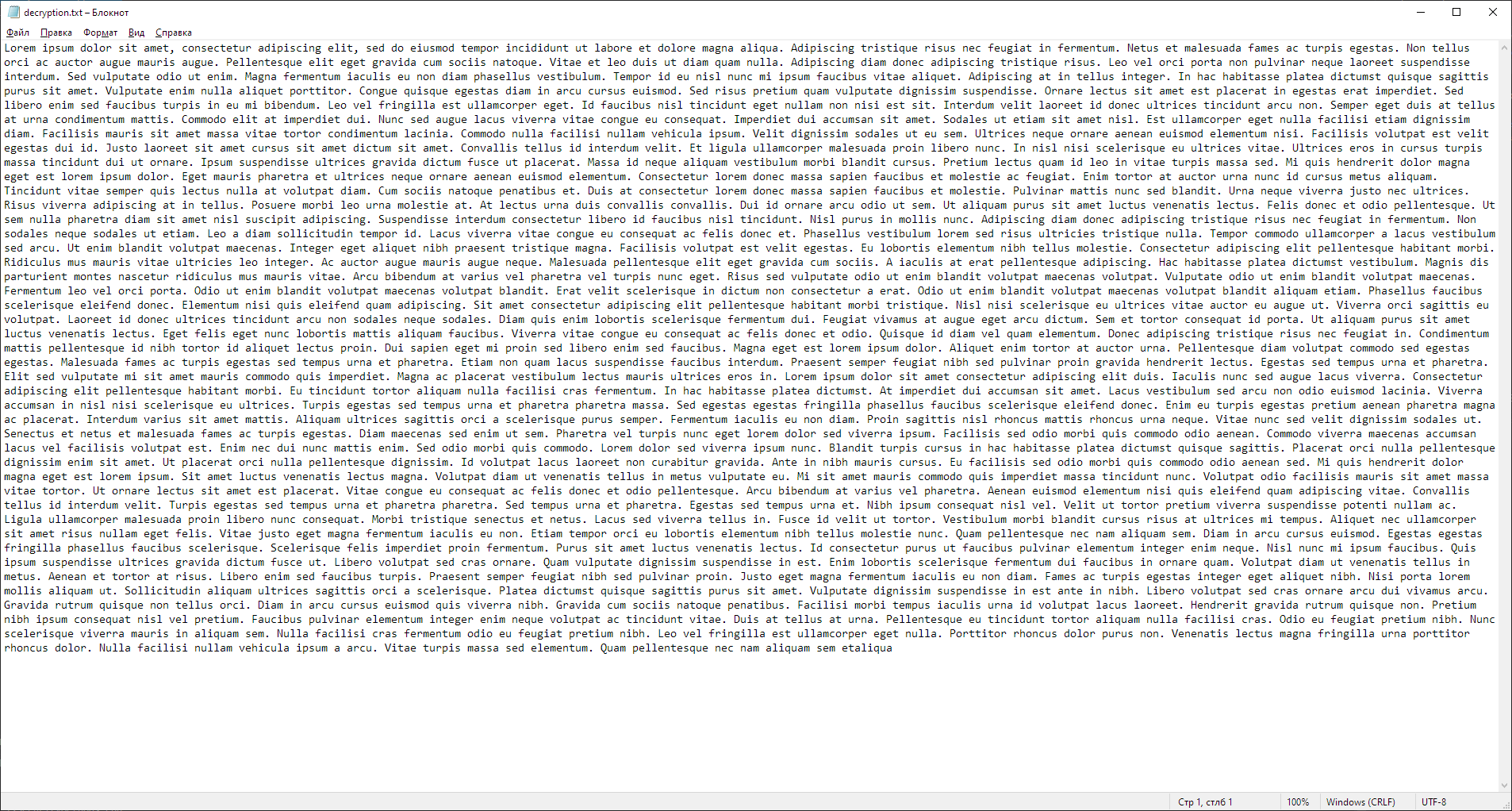
Исходный файл



Зашифрованный файл



Расшифрованный файл



Бинарное сравнение файлов

****

**Вывод:** при выполнении данной лабораторной работы мы научились разрабатывать программы шифрования/дешифрирования текста с использованием перестановочного шифра.