Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №2

по курсу «Защита информации в сети Internet»

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

Выполнили  
студенты группы 18ВВ1:

Арсентьева У.С.

Горбунова В.В.

Приняли:

к.т.н., доцент Дубравин А.В.

к.т.н., доцент Карамышева Н.С.

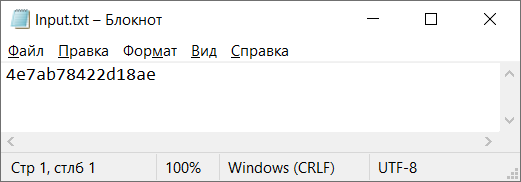
2020

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

**Задания:** Разработать программу, выполняющую шифрование и расшифровывание произвольного двоичного 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.

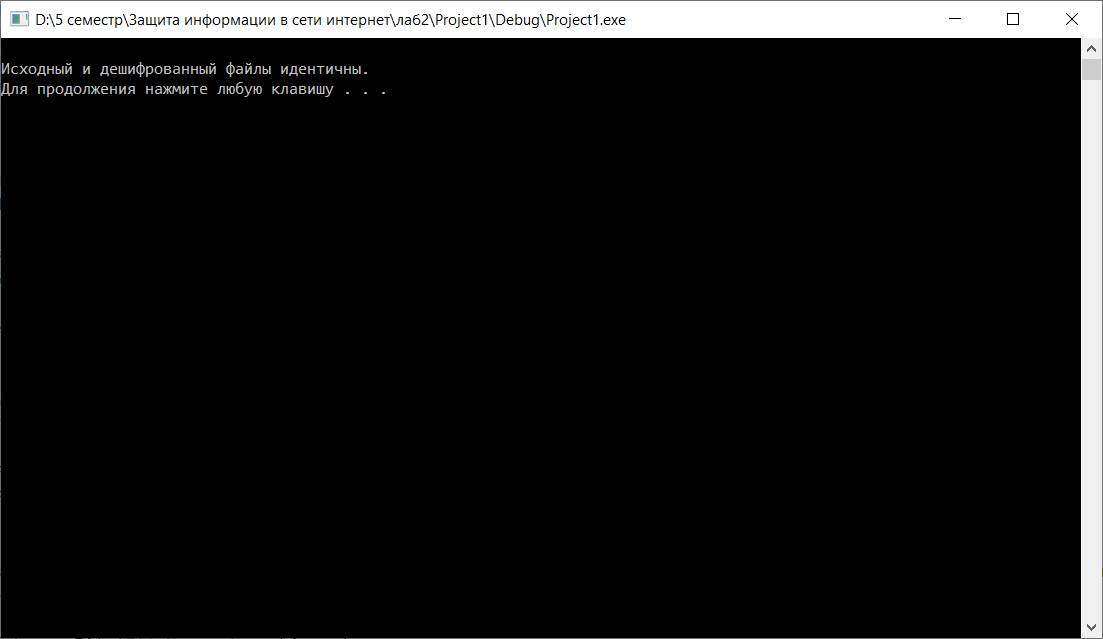
|  |  |
| --- | --- |
| Номер варианта | Последовательность перестановки |
| 4 | 12 13 30 5 27 6 11 25 3 21 22 2 23 0 8 4 18 19 10 1 14 29 9 28 20 17 26 31 7 16 15 24 |

**Исходный двоичный файл:**

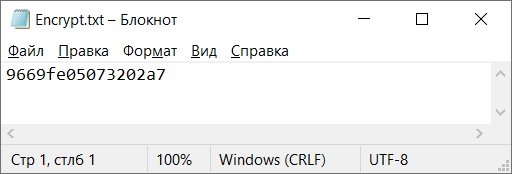


**Рисунок 1**

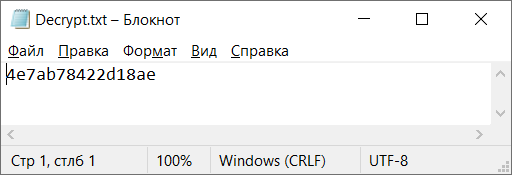
**Результаты работы программы:**



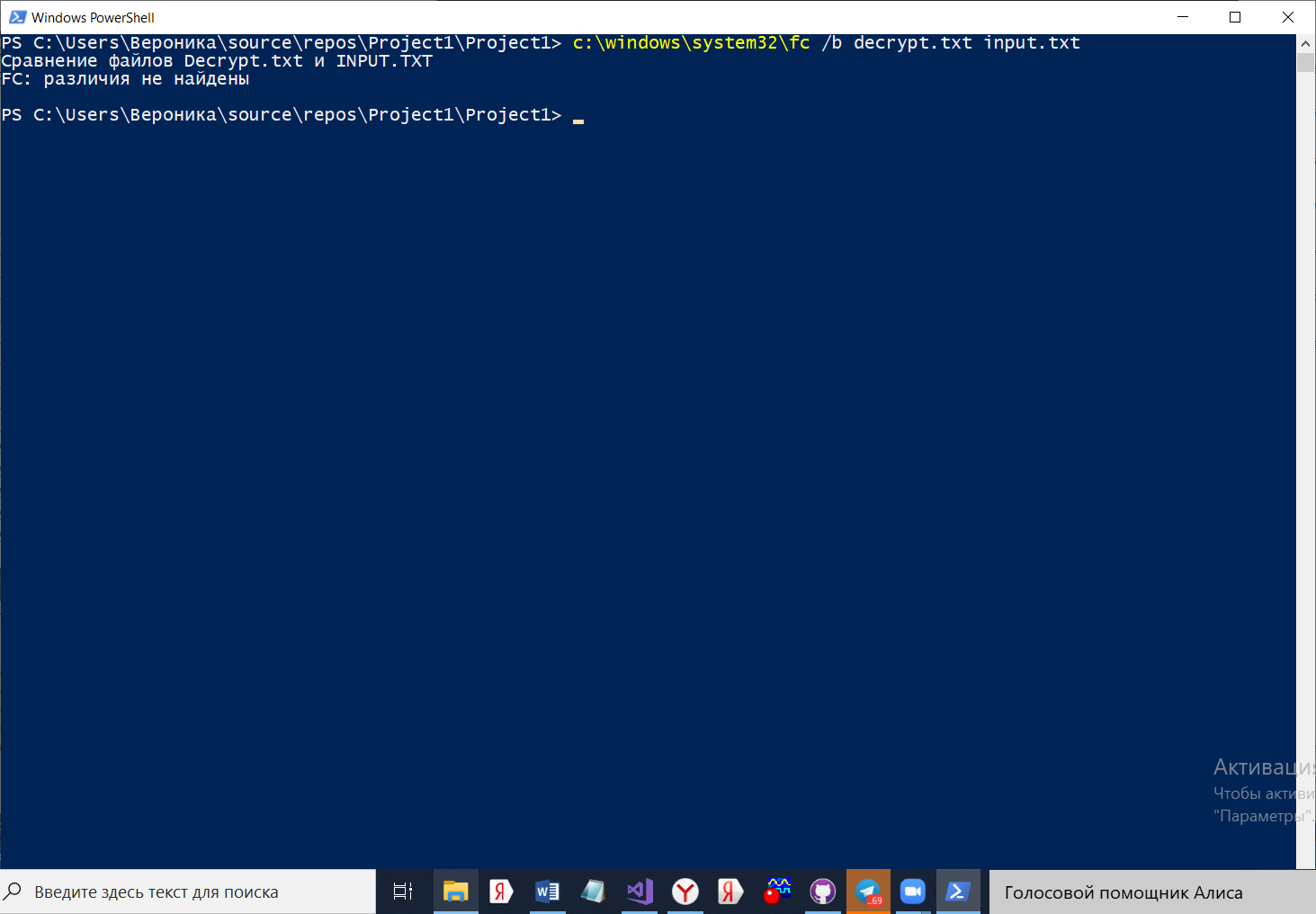
**Рисунок 2 – Результат отладки программы**



**Рисунок 3 – Зашифрованный текст**



**Рисунок 4 – Дешифрованный текст**



**Листинг программы:**

#include <iostream>

#include <fstream>

#include <sstream>

#include <vector>

using namespace std;

//Возвращает N-ый бит 1-байтового числа A, начиная с начала

bool GetBit (unsigned char A, int N)

{

return (A >> 8 - N - 1) & 1u;

}

//Вставляет в значение A указанный бит Bit в позицию N (с начала)

void ChangeBit (unsigned char & A, bool Bit, int N)

{

if (Bit == 0) A &= ~(1 << 8 - N - 1);

if (Bit == 1) A |= (1 << 8 - N - 1);

}

//Возвращает поток ввода на файл FileName; Завершает работу приложения в случае, если файл с указанным именем не найден

ifstream OpenFile (string FileName)

{

ifstream File (FileName, ios::binary);

if (!File.is\_open())

{

cout << "Файл не был открыт." << endl;

system ("pause");

exit(0);

}

return File;

}

//Возвращает индекс массива A размера Size, значение которого равно Value (Используется при дешифровании)

int FindIndex (int A [], int Size, int Value)

{

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

if (A[i] == Value)

return i;

return -1;

}

//Определяет количество нулевых байт в конце массива и удаляет их (Используется при дешифровании)

void DelZeroBytes (vector <unsigned char> & Text)

{

int Size = Text.size();

int ZeroBytes = 0;

for (int i = Size - 1; i >= 0; i --)

if (Text[i] == 0)

ZeroBytes ++;

Text.erase(Text.begin() + Size - ZeroBytes, Text.begin() + Size);

}

//Определяет количество символов в тексте

int LengthFile (string FileName)

{

ifstream file (FileName, ios::binary | ios::ate);

int Length = file.tellg();

file.close();

return Length;

}

//Возвращает текст из файла FileName и его размер Size

vector <unsigned char> GetTextInFile (string FileName, int & Size)

{

ifstream File = OpenFile (FileName);

vector <unsigned char> Text (Size, 0);

//Символы, считанные из файла, загружаются в массив

char C;

for (int i = 0; File.get(C); i ++)

Text[i] = unsigned char (C);

File.close();

return Text;

}

//Выводит элементы вектора Vector шаблонного типа Type в поток вывода Stream

//(Используется для вывода массива битов и строк в файл или на экран)

template <class Type>

ostream & operator << (ostream & Stream, vector <Type> Vector)

{

int Size = Vector.size();

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

if (Vector[i] != 13 || Vector[i + 1] != 10)

Stream << Vector [i];

return Stream;

}

//Шифрование/Дешифрование текста файла InputFile ключом Key размера Size = 4. Результат шифрования выодится в файл OutputFile.

//Если Mod = Coding - выполняется шифрование,

//Если Mod = Decode - выполняется дешифрование

void CodingFile (string InputFile, string OutputFile, int Key [], string Mode)

{

int Size = 4;

ofstream Output (OutputFile, ios::trunc); //Открывает файл для вывода в него зашифрованного текста, ios::trunc удаляет содержимое файла, если он не пуст

int Length = LengthFile (InputFile); //Находит размер исходного файла в байтах

int NZeros = Size - Length % Size; //Определяет число нулевых байт, добавляемых в массив битов для срабатывания ключа

if (Length % Size == 0) NZeros = 0;

Length += NZeros;

vector <unsigned char> Text = GetTextInFile (InputFile, Length); //Загружает текст из исхдного файла

//cout << Text << endl; //Выводит исходный текст на экран

/\*for (int i = 0; i < Length; i ++) //Выводим биты на экран

for (int j = 0; j < 8; j ++)

cout << bool(GetBit(Text[i], j));

cout << endl;\*/

vector <unsigned char> OutputText (Length, 0);

if (Mode == "Coding") //Шифрование файла

for (int i = 0; i < Length / Size; i ++) //Шифрование битов

for (int j = 0; j < Size \* 8; j ++)

ChangeBit(OutputText[Size \* i + j / 8], GetBit(Text[Size \* i + Key[j] / 8], Key[j] % 8), j % 8);

if (Mode == "Decode") //Дешифрование файла

{

for (int i = 0; i < Length / Size; i ++) //Дешифрование битов

for (int j = 0; j < Size \* 8; j ++)

ChangeBit(OutputText[Size \* i + j / 8], GetBit(Text[Size \* i + FindIndex(Key, 32, j) / 8], FindIndex(Key, 32, j) % 8), j % 8);

DelZeroBytes(OutputText);

}

//Вывод полученного текста после шифрования / дешифрования битов в файл

Output << OutputText;

Output.close();

}

//Выполняет сравнение файлов по содержимому

bool CompareFiles (string FileName1, string FileName2)

{

ifstream File1 = OpenFile (FileName1);

ifstream File2 = OpenFile (FileName2);

char ch1, ch2;

bool Result = true;

while (File1.get(ch1) && File2.get(ch2))

{

if (ch1 != ch2)

{

Result = false;

break;

}

}

File1.close();

File2.close();

return Result;

}

/\*

Input.txt - Исходный текстовый файл

Encrypt.txt - Шифрованный текстовый файл

Decrypt.txt - Дешифрованный текстовый файл

\*/

void main()

{

setlocale(LC\_ALL, "Russian");

const int N = 32; //Число значений в перестановочном ключе

int Key [N] = {12, 13, 30, 5, 27, 6, 11, 25, 3, 21, 22, 2, 23, 0, 8, 4, 18, 19, 10, 1, 14, 29, 9, 28, 20, 17, 26, 31, 7, 16, 15, 24}; //Перестановочный ключ

cout << "Исходный текстовый файл:" << endl;

CodingFile ("Input.txt", "Encrypt.txt", Key, "Coding"); //Шифрование текста

cout << endl << "Зашифрованный текстовый файл:" << endl;

CodingFile ("Encrypt.txt", "Decrypt.txt", Key, "Decode"); //Расшифрование текста

bool Compare = CompareFiles("Input.txt", "Decrypt.txt"); //Двоичное сравнение исхдного и расшифрованного файлов

if (Compare) cout << endl << "Исходный и дешифрованный файлы идентичны." << endl;

else cout << endl << "Исходный и дешифрованный файлы различны." << endl;

system ("pause");

}

**Вывод:** Была разработана программа, выполняющая шифрование, дешифрование двоичного файла с использованием перестановочного шифра.