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:

Арсентьева У.С.

Горбунова В.В.

Приняли:

к.т.н., доцент Дубравин А.В.

к.т.н., доцент Карамышева Н.С.

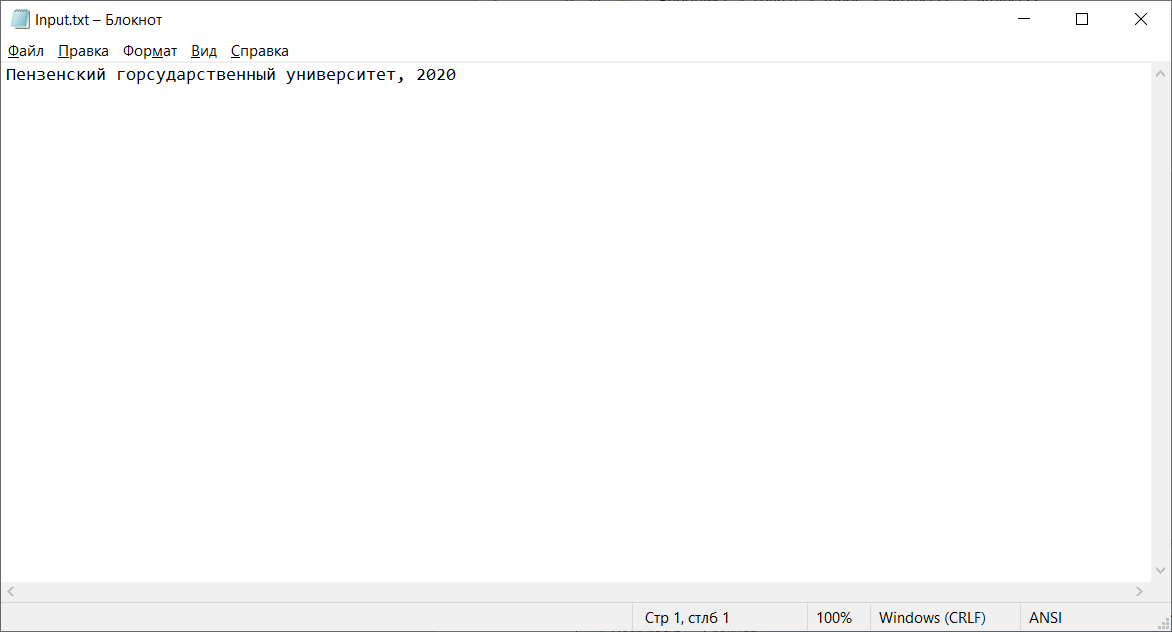
2020

**Цель работы:** разработать программу шифрования/дешифрирования 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 |
| 4 | 1 2 10 4 8 7 3 6 9 5 |

**Исходный текстовый файл:**



**Рисунок 1 – Исходный файл**

**Листинг программы:**

#include <iostream>

#include <sstream>

#include <fstream>

#include <vector>

#include <locale.h>

using namespace std;

//Возвращает файловый поток ввода (Режим открытия Mode); завершает работу приложения в случае, если файл с указанным именем не найден

ifstream OpenFile(string FileName, ios\_base::openmode Mode = ios::in)

{

ifstream Input(FileName, Mode);

if (!Input.is\_open()) //Если файл не найден

{

cout << "Файл не найден" << endl;

system("pause");

exit(0);

}

return Input;

}

//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;

}

//Удаляет пробелы в конце массива (Используется при дешифровании)

void DelSpace(vector <char> & Array)

{

while (Array.back() == ' ')

Array.pop\_back(); //Удаляет последний элемент

}

//Определяет количество символов в тексте

int LengthFile(string FileName)

{

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

int Length = file.tellg();

file.close();

return Length;

}

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

//Если Mod = "Encrypt" - выполняется шифрование,

//Если Mod = "Decrypt" - выполняется дешифрование

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

{

ifstream Input = OpenFile(InputFile); //Поток на исходный файл

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

//Определяет размер файла в байтах

int Length = LengthFile(InputFile);

int NSpaces = Size - Length % Size; //Число пробелов, добавляемых в строку для срабатывания ключа

Length += NSpaces;

vector <char> Text(Length, ' ');

//Пока не достигнут конец файла, считывать символы в массив Text и одновременно выводить их на экран

for (int i = 0; Input.get(Text[i]); i++);

//cout << Text[i];

if (Mode == "Encrypt") //Шифрование строки

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

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

Output << Text[Size \* i + Key[j] - 1];

if (Mode == "Decrypt") //Дешифрование строки

{

vector <char> Decrypt(Length);

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

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

Decrypt[Size \* i + j] = Text[Size \* i + FindIndex(Key, Size, j + 1)];

DelSpace(Decrypt); //Удаляет пробелы в конце массива

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

Output << Decrypt[i];

}

Input.close(); //Закрыть потоки

Output.close();

}

//Выполняет сравнение файлов по содержимому

bool CompareFiles(string FileName1, string FileName2)

{

ifstream File1 = OpenFile(FileName1, ios::binary);

ifstream File2 = OpenFile(FileName2, ios::binary);

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 - Расшифрованный текстовый файл

\*/

void main()

{

setlocale(LC\_ALL,"Rus");

const int N = 10; //Число значений в перестановочном ключе

int Key[N] = { 1, 2, 10, 4, 8, 7, 3, 6, 9, 5 }; //Перестановочный ключ

//cout << "Original text file:" << endl;

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

//cout << endl << endl << "Encrypted text file:" << endl;

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

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

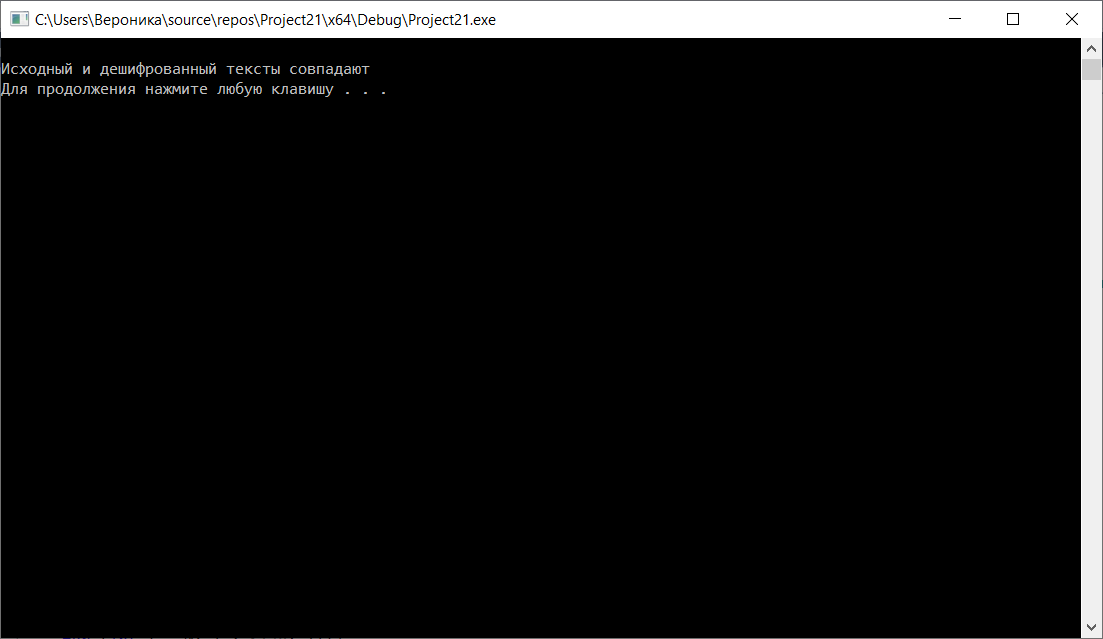
if (Compare) cout << endl << "Исходный и дешифрованный тексты совпадают" << endl;

else cout << endl << "Исходный и дешифрованный тексты не совпадают" << endl;

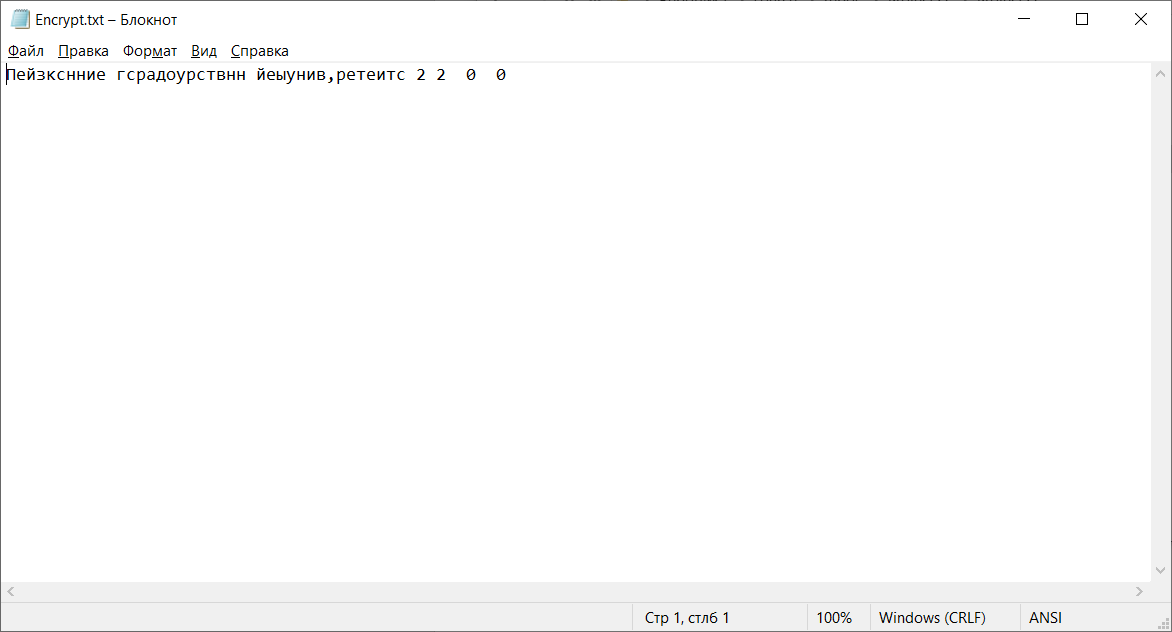
system("pause");

}

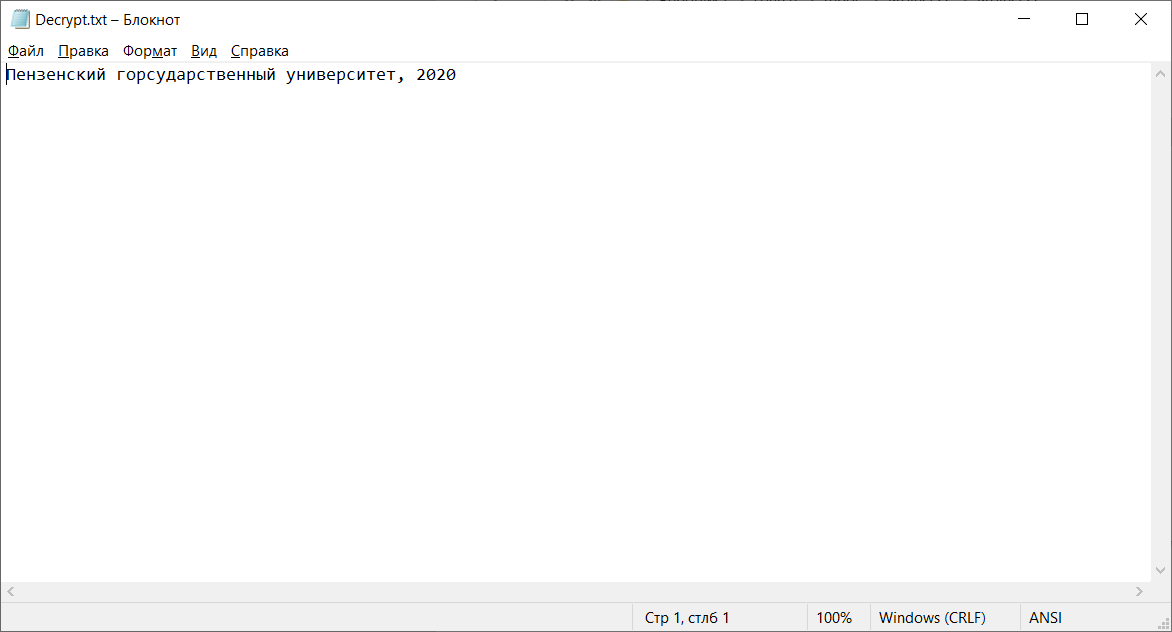
**Результаты работы программы:**



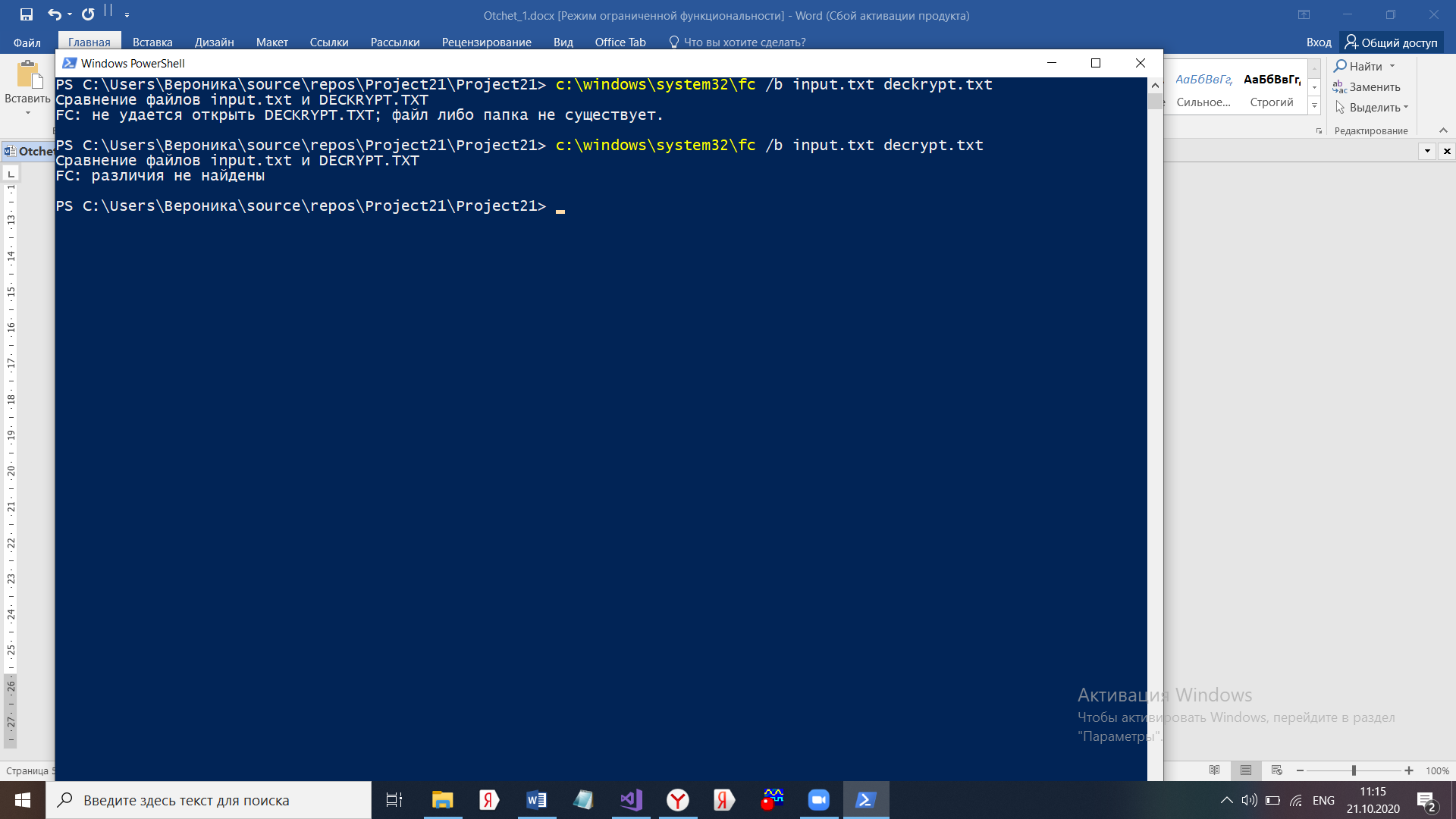
**Рисунок 2 – Результат отладки программы**



**Рисунок 3 – Зашифрованный текст**



**Рисунок 4 – Дешифрованный текст**



**Рисунок 5 – Проверка на различия исходного и дешифрованного файлов**

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