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/дешифрования текста»

Выполнили:  
студенты группы 18ВВ1:  
Федулов В.  
Пчелинцев А.

Приняли:

к.т.н. Дубравин А.В.

Пенза 2020

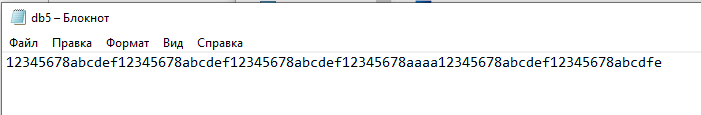
**Цель работы**: разработка программы шифрования/дешифрирования двоичных файлов с использованием перестановочного шифра.

**Задание:** 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.

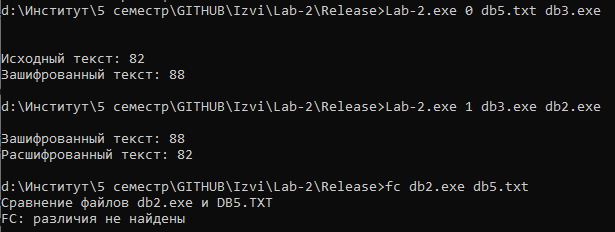
**Последовательность перестановки:** 24 10 2 8 21 26 28 0 18 12 11 25 5 22 1 16 3 9 13 4 19 17 6 15 14 20 7 23 30 31 27 29.

**Результат работы программы:**

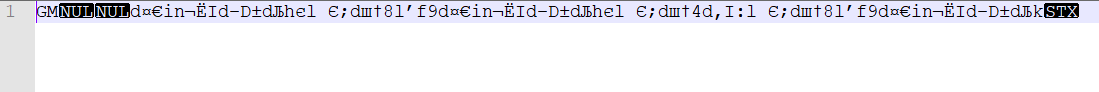
Для проверки правильности работы программы, создали файл формата txt и заполнили его числами и буквами.



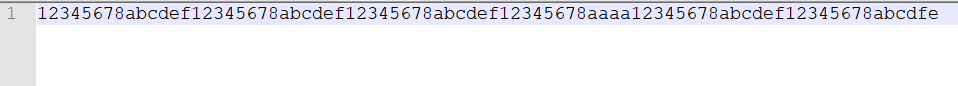
Запустили программу, по завершению она выдала следующий результат:



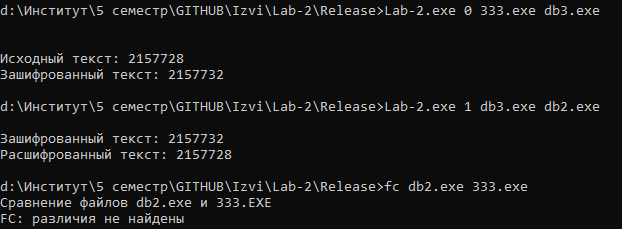
Открыли файл с зашифрованным текстом:



Открыли файл с расшифрованным текстом, он не отличается от исходного текста:



После чего запустили программу на шифрование большого exe файла содержащего 2157728 символов.



В зашифрованном тексте добавились недостающие символы.

**Листинг:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <conio.h>

#include <Windows.h>

#include <locale.h>

#include <time.h>

#include <math.h>

char key[32] = { 24, 10, 2, 8, 21, 26, 28, 0, 18, 12, 11, 25, 5, 22, 1, 16, 3, 9, 13, 4, 19, 17, 6, 15, 14, 20, 7, 23, 30, 31, 27, 29 };

void perevod(char buff[], int c, int d, int a, bool shif, char buffOut[]) {

unsigned int m = 0, out = 0;

for (char j = c; j < d; j++) {

for (char k = 0; k < 4; k++) {

m = m << 8;

if (k < a) {

m = m + (unsigned char)buff[k + j \* 4];

}

else

m = m + 48;

}

for (int i = 31; i >= 0; i--) {

if (shif) {

if (m & (1 << 31 - key[i]))

out |= (1 << 31 - i);

}

else {

for (char t = 0; t < 32; t++) {

if (key[t] == i) {

if (m & (1 << 31 - t)) {

out |= (1 << 31 - i);

}

break;

}

}

}

}

for (char k = 0; k < 4; k++) {

buffOut[3 - k + j \* 4] = out;

out = out >> 8;

}

}

}

int shifrator(const char source[], const char drain[]) {

FILE\* fp, \* f;

char buff[80], buffOut[80];

//---------------------------------------------------------------

// открытие для чтения

if ((fp = fopen(source, "rb")) == NULL) {

printf("Не возможно открыть файл.");

return 1;

}

if ((f = fopen(drain, "wb")) == NULL) {

printf("Не возможно открыть файл.");

return 1;

}

//---------------------------------------------------------------

strncpy(buffOut, " ", 8);

strncpy(buff, "", 80);

fwrite(buffOut, 4, 1, f);

//---------------------------------------------------------------

int pos1 = ftell(fp);

while (fread(buff, 80, 1, fp) != 0) {

pos1 = ftell(fp);

perevod(buff, 0, 20, 8, true, buffOut);

fwrite(buffOut, sizeof buffOut, 1, f);

}

//---------------------------------------------------------------

fseek(fp, 0, SEEK\_END);

int pos2 = ftell(fp) - pos1;;

fseek(fp, pos1, SEEK\_SET);

if (pos2 != 0) {

int a = pos2 % 4, b = pos2 / 4, c = (b + (a == 0 ? 0 : 1)) \* 4;

strncpy(buffOut, "", 80);

fread(buff, pos2, 1, fp);

perevod(buff, 0, b, 8, true, buffOut);

if (a != 0)

perevod(buff, b, b + 1, a, true, buffOut);

fwrite(buffOut, c, 1, f);

//---------------------------------------------------------------

for (char i = 0; i < a;) {

buffOut[i] = rand() % 60 + 30;

if (buffOut[i] != ' ')

i++;

}

fseek(f, 0, SEEK\_SET);

fwrite(buffOut, a, 1, f);

}

//---------------------------------------------------------------

fseek(fp, 0, SEEK\_END);

fseek(f, 0, SEEK\_END);

printf("\n\nИсходный текст: %d ", ftell(fp));

printf("\nЗашифрованный текст: %d \n", ftell(f));

fclose(fp);

fclose(f);

return 0;

}

int deshifrator(const char source[], const char drain[]) {

FILE\* fp, \* deshif;

char buff[80], buffOut[80], plicity = 0;

//---------------------------------------------------------------

if ((fp = fopen(source, "rb")) == NULL) {

printf("Не возможно открыть файл.");

return 1;

}

if ((deshif = fopen(drain, "wb")) == NULL) {

printf("Не возможно открыть файл.");

return 1;

}

//---------------------------------------------------------------

fread(buffOut, 4, 1, fp);

while (plicity < 4 && buffOut[plicity] != '\0' && buffOut[plicity] != ' ') {

plicity++;

}

strncpy(buff, "", 80);

strncpy(buffOut, "", 80);

//---------------------------------------------------------------

int pos1 = ftell(fp);

while (fread(buff, 80, 1, fp) != 0) {

pos1 = ftell(fp);

perevod(buff, 0, 20, 8, false, buffOut);

fseek(fp, 0, SEEK\_END);

if (ftell(fp) - pos1 > 0)

fwrite(buffOut, 80, 1, deshif);

fseek(fp, pos1, SEEK\_SET);

}

//---------------------------------------------------------------

fseek(fp, 0, SEEK\_END); // устанавливаем позицию в конец файла

int pos2 = ftell(fp) - pos1; // получаем размер в байтах

fseek(fp, pos1, SEEK\_SET);

if (pos2 == 0) {

if (plicity != 0) {

pos2 = 72 + plicity;

fseek(deshif, pos1 - 88, SEEK\_SET);

}

else

pos2 = 80;

fwrite(buffOut, pos2, 1, deshif);

}

else if (pos2 == 4 && plicity != 0) {

perevod(buff, 0, plicity, 8, false, buffOut);

fwrite(buffOut, pos2 + plicity - 4, 1, deshif);

}

else if (pos2 != 0) {

perevod(buff, 0, pos2 / 4, 8, false, buffOut);

if (plicity != 0)

pos2 = pos2 - 4 + plicity;

fwrite(buffOut, pos2, 1, deshif);

}

//---------------------------------------------------------------

fseek(fp, 0, SEEK\_END);

fseek(deshif, 0, SEEK\_END);

printf("\nЗашифрованный текст: %d ", ftell(fp));

printf("\nРасшифрованный текст: %d \n", ftell(deshif));

fclose(fp);

fclose(deshif);

return 0;

}

int main(int argc, char\* argv[]) {

setlocale(0, "Rus");

//char mas1[] = { "mi.txt" };

//char mas2[] = { "db2.exe" };

//char mas3[] = { "db3.exe" };

//

//system("\n fc mi.txt db3.exe");

int d = atoi(argv[1]);

if (d == 0) {

shifrator(argv[2], argv[3]);

}

else if (d == 1) {

deshifrator(argv[2], argv[3]);

}

\_getch();

return 0;

}

**Вывод:** в ходе данной лабораторной работы, мы создали программу шифрования/дешифрования двоичного файла с использованием перестановочного шифра.