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/дешифрования текста»

Выполнили:  
студенты группы 18ВВ1:  
Федулов В.,

Пчелинцев А.

Приняли:

к.т.н. Дубравин А.В.

Пенза 2020

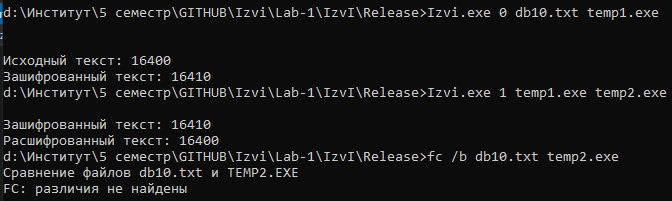
**Цель работы**: разработка программы шифрования/дешифрирования текста с использованием перестановочного шифра.

**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 файла, полученного после расшифровывания.

**Последовательность перестановки:** 2 8 10 4 3 7 1 6 9 5.

**Результат работы программы:**

Запустили программу на шифрование большого exe файла содержащего 16400 символов.



В зашифрованном тексте добавились недостающие символы, так как ключ содержит 10 символов.

**Листинг:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <conio.h>

#include <Windows.h>

#include <locale.h>

#include <time.h>

char key[10] = { 2, 8, 10, 4, 3, 7, 1, 6, 9, 5 };

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

FILE\* fp, \* f;

char buff[100], buffOut[100];

srand(time(NULL));

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

// открытие для чтения

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

printf("cannot open file");

return 1;

}

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

printf("Cannot open file.");

return 1;

}

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

for (int j = 0; j < 10; j++)

buffOut[j] = ' ';

fwrite(buffOut, 10, 1, f);

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

int pos1 = ftell(fp);

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

pos1 = ftell(fp);

for (int j = 0; j < 10; j++)

for (int r = 0; r < 10; r++)

buffOut[r + j \* 10] = buff[key[r] - 1 + j \* 10];

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

}

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

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) {

int a = pos2 % 10, b = pos2 / 10, c = (b + (a == 0 ? 0 : 1)) \* 10, d = b \* 10 + a;

fread(buff, pos2, 1, fp);

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

for (int r = 0; r < 10; r++)

buffOut[r + j \* 10] = buff[key[r] - 1 + j \* 10];

if (a != 0) {

for (int r = 0; r < 10; r++) {

if (key[r] - 1 < a)

buffOut[r + b \* 10] = buff[key[r] - 1 + b \* 10];

else

buffOut[r + b \* 10] = ' ';

}

}

fwrite(buffOut, c, 1, f);

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

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

buffOut[i] = buff[(rand() % 9) + (rand() % 9) \* 10] + rand() % 9 + 1;

if (buffOut[i] != ' ')

i++;

}

fseek(f, 0, SEEK\_SET);

fwrite(buffOut, a, 1, f);

}

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

fseek(fp, 0, SEEK\_END);

fseek(f, 0, SEEK\_END);

setlocale(0, "Rus");

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

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

fclose(fp);

fclose(f);

return 0;

}

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

FILE\* fp, \* deshif;

char buff[100], buffOut[100];

char plicity = 0;

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

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

printf("Cannot open file.");

return 1;

}

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

printf("Cannot open file.");

return 1;

}

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

fread(buffOut, 10, 1, fp);

while (plicity < 10 && buffOut[plicity] != ' ') {

plicity++;

}

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

int pos1 = ftell(fp);

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

pos1 = ftell(fp);

for (int j = 0; j < 10; j++)

for (int r = 0; r < 10; r++)

buffOut[key[r] - 1 + j \* 10] = buff[r + j \* 10];

fseek(fp, 0, SEEK\_END);

if (ftell(fp) - pos1 > 0)

fwrite(buffOut, sizeof buffOut, 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 = 90 + plicity;

fseek(deshif, pos1 - 110, SEEK\_SET);

}

else

pos2 = 100;

fwrite(buffOut, pos2, 1, deshif);

}

else if (pos2 == 10 && plicity != 0) {

pos2 = pos2 + plicity - 10;

fwrite(buffOut, pos2, 1, deshif);

}

else if (pos2 != 0) {

if (plicity != 0)

pos2 = pos2 - 10 + plicity;

fwrite(buff, pos2, 1, deshif);

}

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

fseek(fp, 0, SEEK\_END);

fseek(deshif, 0, SEEK\_END);

setlocale(0, "Rus");

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

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

fclose(fp);

fclose(deshif);

return 0;

}

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

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 текста с использованием перестановочного шифра.