

**Московский государственный технический
университет им. Н. Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управление»

Курс «Основы программирования»

Отчет по лабораторной работе №6
«Шифрование текстовых файлов»

Выполнил:

Студент группы ИУ5-11Б
Алехин Сергей

Подпись и дата:

Проверил:

Преподаватель каф. ИУ5
Правдина Анна Дмитриевна

Подпись и дата:

Москва, 2018 г.

Задание

Использование потокового ввода – вывода для работы с текстовыми файлами.
Использование библиотечных функций для обработки символьных строк.
Шифрование символьных данных с переменным ключом.
Анализ результатов шифрования текстовых файлов.

Разработка алгоритма

Входные переменные:

- 1) const int n – длина массива анализа данных;
- 2) char buff[] – массив буфера для обработки слов из кодового файла;
- 3) char cWord – считываемый символ из файлов 1.txt/2.txt;
- 4) char cWord1 – считываемый символ из файла 1.txt для заполнения массива анализа данных;
- 5) char cWord2 – считываемый символ из файла 2.txt для заполнения массива анализа данных;
- 6) char y – переменная отвечающая за выполнение кодирования/декодирования/анализа данных;
- 7) int *iCodeTable – динамический массив из кодов сдвига;
- 8) int iCountWordfilecode – количество слов в файле kod.txt;
- 9) int iLengthbuff – длина считываемого слова из файла kod.txt;
- 10) int iCodeTableIndex – номер элемента динамического массива *iCodeTable;
- 11) int iWordCode – код зашифрованного/дешифрованного символа;
- 12) int iWordIndex – номер считываемого символа;
- 13) int iSum – сумма символов в строке;
- 14) int jSum – сумма символов в столбце;
- 15) int max – максимальное значение в столбце;
- 16) int c – максимальная длина в столбце;
- 17) int iStatTable[][] – массив анализа данных;
- 18) ifstream filecode – поток чтения файла kod.txt;
- 19) ifstream file1 – поток чтения файла 1.txt;
- 20) ifstream file2in – поток чтения файла 2.txt;
- 21) ofstream file2out – поток записи файла 2.txt;
- 22) ofstream file3 – поток записи файла 3.txt.

Текст программы

```
#include <iostream>
#include <fstream>
#include <cstring>
#include <iomanip>

using namespace std;

int main()
{
    setlocale(0, "RUSSIAN");
    ifstream filecode;
    ifstream file1;
    ifstream file2in;
    ofstream file2out;
    ofstream file3;
    const int n = 256;
    char buff[256], cWord, cWord1, cWord2, y;
    int *iCodeTable, iCountWordfilecode, iLengthbuff, iCodeTableIndex, iWordCode,
    iWordIndex, iSum, jSum, max, c, iStatTable[n][n];

    //Нахождение кодовой таблицы
    filecode.open("kod.txt");
```

```

//Нахождение количества слов
filecode >> buff;
iCountWordfilecode = 1;
while (!filecode.eof())
{
    iCountWordfilecode++;
    filecode >> buff;
}

//filecode.seekg(0, ios::beg);
filecode.close();
filecode.open("kod.txt");

//Создание и обнуление массива кодов сдвига
iCodeTable = new int[iCountWordfilecode];
for (int i = 0; i < iCountWordfilecode; i++)
    iCodeTable[i] = 0;
//Заполнение массива кодов сдвига
filecode >> buff;
iCodeTableIndex = 0;
while (!filecode.eof())
{
    iLengthbuff = strlen(buff);
    for (int i = 0; i < iLengthbuff; i++)
        iCodeTable[iCodeTableIndex] += (int)buff[i];
    iCodeTableIndex++;
    filecode >> buff;
}

filecode.close();

cout << "Кодировать текст? (y/n)" << endl;
cin >> y;
if (y == 121)
{
    //Кодирование текста
    file1.open("1.txt", ios::binary);
    file2out.open("2.txt", ios_base::out | ios::binary);

    iWordIndex = 0;
    file1.get(cWord);
    while (!file1.eof())
    {
        iWordCode = (iCodeTable[iWordIndex % iCountWordfilecode] +
(int)cWord) % 256;
        file2out << (char)iWordCode;
        iWordIndex++;
        file1.get(cWord);
    }

    file1.close();
    file2out.close();
}

cout << "Декодировать текст? (y/n)" << endl;
cin >> y;
if (y == 121)

```

```

{
    //Декодирование текста
    file2in.open("2.txt", ios::binary);
    file3.open("3.txt", ios_base::out | ios::binary);

    iWordIndex = 0;
    file2in.get(cWord);
    while (!file2in.eof())
    {
        iWordCode = ((int)cWord - iCodeTable[iWordIndex %
iCountWordfilecode] % 256 + 256) % 256;
        file3 << (char)iWordCode;
        iWordIndex++;
        file2in.get(cWord);
    }

    file2in.close();
    file3.close();
}

cout << "Найти статистику? (y/n)" << endl;
cin >> y;
if (y == 121)
{
    //Нахождение статистики
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            iStatTable[i][j] = 0;

    file1.open("1.txt");
    file2in.open("2.txt");
    //Заполнение матрицы
    file1.get(cWord1);
    file2in.get(cWord2);
    while (!file1.eof())
    {
        iStatTable[(((int)cWord1 + 256) % 256) % n][(((int)cWord2 + 256) % 256)
% n]++;

        file1.get(cWord1);
        file2in.get(cWord2);
    }
    //Вывод матрицы
    cout << " ";
    for (int j = 0; j < n; j++)
    {
        jSum = 0;
        max = iStatTable[0][j];
        for (int k = 0; k < n; k++)
        {
            jSum += iStatTable[k][j];
            if (iStatTable[k][j] > max) max = iStatTable[k][j];
        }
        if (jSum == 0) continue;
        c = 0;
        while (max != 0)
        {
            max /= 10;

```

```

        c++;
    }
    cout << setw(c) << (char)j << ' ';
}
//cout << endl;

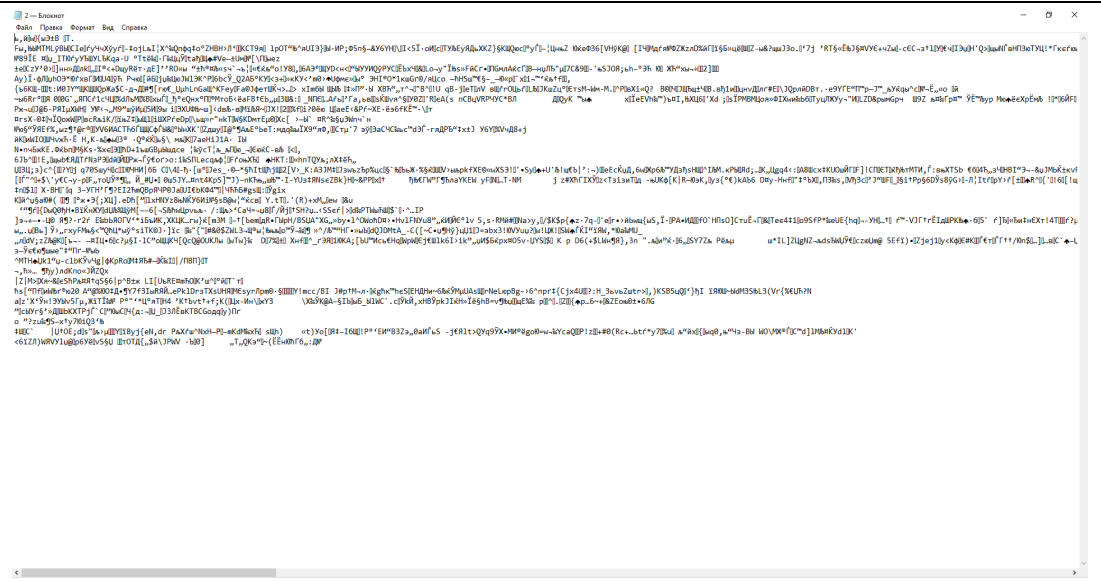
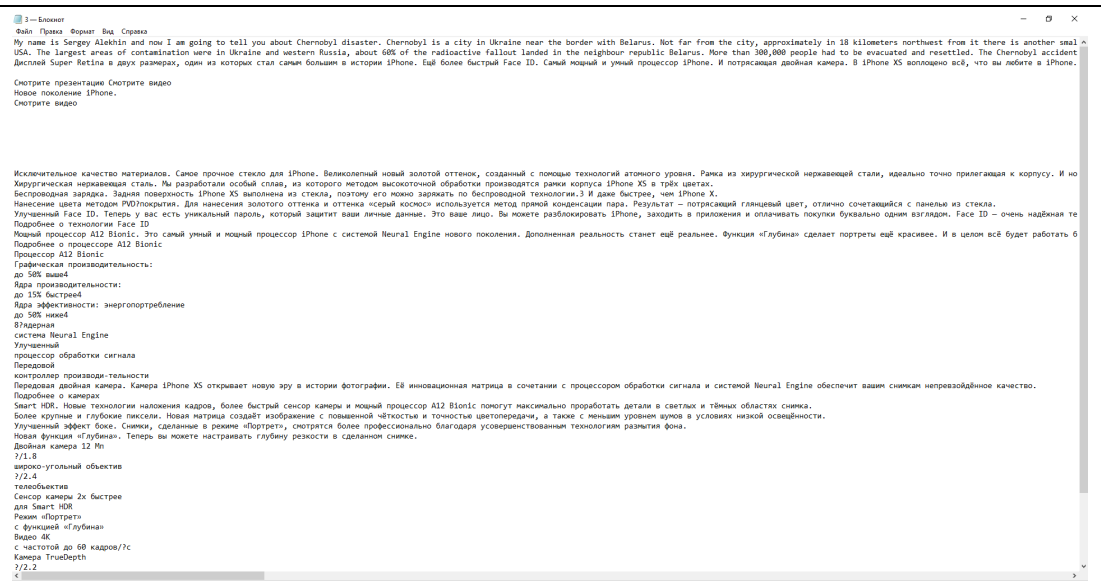

for (int i = 0; i < n; i++)
{
    iSum = 0;
    for (int k = 0; k < n; k++)
        iSum += iStatTable[i][k];
    if (iSum == 0) continue;
    cout << (char)i << ' ';
    for (int j = 0; j < n; j++)
    {
        jSum = 0;
        max = iStatTable[0][j];
        for (int k = 0; k < n; k++)
        {
            jSum += iStatTable[k][j];
            if (iStatTable[k][j] > max) max = iStatTable[k][j];
        }
        if (jSum == 0) continue;
        c = 0;
        while (max != 0)
        {
            max /= 10;
            c++;
        }
        cout << setw(c) << iStatTable[i][j] << ' ';
    }
    cout << endl;
}


return 0;
}

```

Анализ результатов

[illegible]

2	<h2>Закодированный файл</h2>	
3	<h2>Декодированный файл</h2>	
4	<h2>Кодовый файл</h2>	

5	Статистика	
---	------------	--

В этой лабораторной мы научились кодировать и декодировать файл с помощью модернизированного метода Цезаря используя кодовый блокнот. В работе использовалась работа с файлами, все тексты писались в файлы.