Московский государственный технический

университет им. Н.Э. Баумана

Курс «Основы информатики»

Отчёт по лабораторной работе №7

Обработка символьных строк.

Проверил:

Преподаватель каф. ИУ5

Аксёнова М.В.

Подпись и дата:

Факультет «Информатика и система управления»

Кафедра ИУ5 «Система обработки информации и управления»

Выполнил:

Студент группы ИУ5-13Б

Пермяков Дмитрий Кириллович

Подпись и дата:

**Постановка задачи.**

Провести кодирование и декодирование текста (массива символов) при помощи кода Цезаря с переменным сдвигом по таблице ASCII-кодов. Величина сдвига для каждой позиции в исходном тексте - сумма (по модулю 256) кодов символов слова кодового блокнота, стоящего в блокноте на той же позиции. Если кодовый блокнот имеет слов меньше, чем количество символов в исходном тексте, то по исчерпании слов в нём перейти к первому слову и продолжить. (На основе кодового блокнота целесообразно сначала сформировать по заданному правилу целочисленный массив ключей, который затем использовать при кодировании. Эти действия оформить в виде отдельной функции.)

Исследовать повторяемость символов в закодированном тексте (сколько каких кодов одного и того же исходного символа получено) в зависимости от кодового блокнота и длины исходного текста. Результаты исследования представить в виде таблицы (продумать формат таблицы). Исследование и вывод таблицы результатов следует выполнять в режиме диалога, последовательно вычисляя и выводя результаты для запрашиваемого символа. Статистические данные хранить в массиве int stat[256], Для большей достоверности статистических результатов в качестве исходного текста и кодового блокнота использовать текстовые файлы размером около 1 Кбайта.

**Разработка алгоритма**

Переменные:

int\* key- массив ключей

char ch – символ, пробегающий по файлу

i-индекс для заполнения массива ключей

unsigned char letter – буква, введённая с клавиатуры для аналитики

int ArrLetter[256][256]- массив, хранящий символ До и После шифра

int num – кол-во ключей

ifstream text – читаем файл text.txt для будущего шифрования

ofstream fin22 – открываем и добавляем в файл encrypt.txt зашифрованный текст из text.txt

ifstream fin2 – читаем зашифрованный файл

ofstream fin3 – открываем и закидываем в файл "decrypt.txt расшифрованный текст

Функции:

void encrypt(char& text, int\* key, int& num, int j, ofstream& file, unsigned char letter, int ArrLetter[256][256]) – принимаем текущий символ, переводим в тип int и прибавляем ключ. Далее обратно переводим в тип char и записываем в файл encrypt.txt

void decrypt(char& text, int\* key, int& num, int j, ofstream& file) – принимаем текущий символ, переводим в тип int и отнимаем ключ. Далее обратно переводим в тип char и записываем в файл decrypt.txt

**Код программы.**

**Заголовки:**

#include <iostream>

#include <fstream>

#include <iomanip>

#include <clocale>

#include <Windows.h>

#include <stdio.h>

#include <cstdint>

using namespace std;

void encrypt(char& text, int\* key, int& num, int j, ofstream& file, unsigned char letter, int ArrLetter[256][256]);

void decrypt(char& text, int\* key, int& num, int j, ofstream& file);

**Функции**

#include "Header.h"

void encrypt(char& text, int\* key, int& num, int j, ofstream& file, unsigned char letter, int ArrLetter[256][256]){

// Аналитика

unsigned char text1 = text;

if (letter == text1)

++ArrLetter[(int)letter][((int)text1 + key[num % j]) % 256];

// кек

text1 = static\_cast<char>((static\_cast<int>(text1) + key[num % j]) % 256);

file << text1;

++num;

}

void decrypt(char& text, int\* key, int& num, int j, ofstream& file){

text = static\_cast<char>((static\_cast<int>(text) - key[num % j]) % 256);

file << text;

++num;

}

#include "Header.h"

**Main:**

int main()

{

setlocale(LC\_ALL, "rus");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

// получаем сумму ключей

int\* key = new int[1000]{ 0 };

char ch;

int i = -1;

ifstream fin("key.txt", ios::binary);

bool newLetter = true;

while (fin.get(ch))

{

unsigned char ch\_old = static\_cast<unsigned char>(ch);

if (ispunct(ch\_old) || isspace(ch\_old)){

newLetter = true;

continue;

}

if (newLetter){

++i;

newLetter = false;

}

key[i] += static\_cast<int>(ch\_old);

if (i == 1000)

break;

}

fin.close();

unsigned char letter;

cout << "Enter the letter for analytic: ";

cout << "\x1b[35m";

cin >> letter;

cout << "\x1b[0m\n";

int ArrLetter[256][256]{ 0 };

// шифруем

int num = 0;

ifstream text("text.txt", ios::binary);

ofstream fin22("encrypt.txt", ios::binary);

while (text.get(ch))

encrypt(ch, key, num, i+1, fin22, letter, ArrLetter);

cout << endl;

text.close();

fin22.close();

// читаем файл и расшифровываем

num = 0;

ifstream fin2("encrypt.txt", ios::binary);

ofstream fin3("decrypt.txt", ios::binary);

while (fin2.get(ch))

decrypt(ch, key, num, i + 1, fin3);

fin2.close();

fin3.close();

cout << endl;

cout << endl << "Analytics" << endl;

cout << setw(8) << "symbol " << '|' << setw(12) << "its kod " << '|' << setw(10) << "numbers " << '|' << endl;

for (int j = 0; j < 256; j++) {

if (ArrLetter[(int)letter][j] > 0)

cout << setw(4) << (char)(j) << setw(5) << '|'

<< setw(8) << j << setw(5) << '|'

<< setw(6) << ArrLetter[(int)(letter)][j] << setw(5) << '|' << endl;

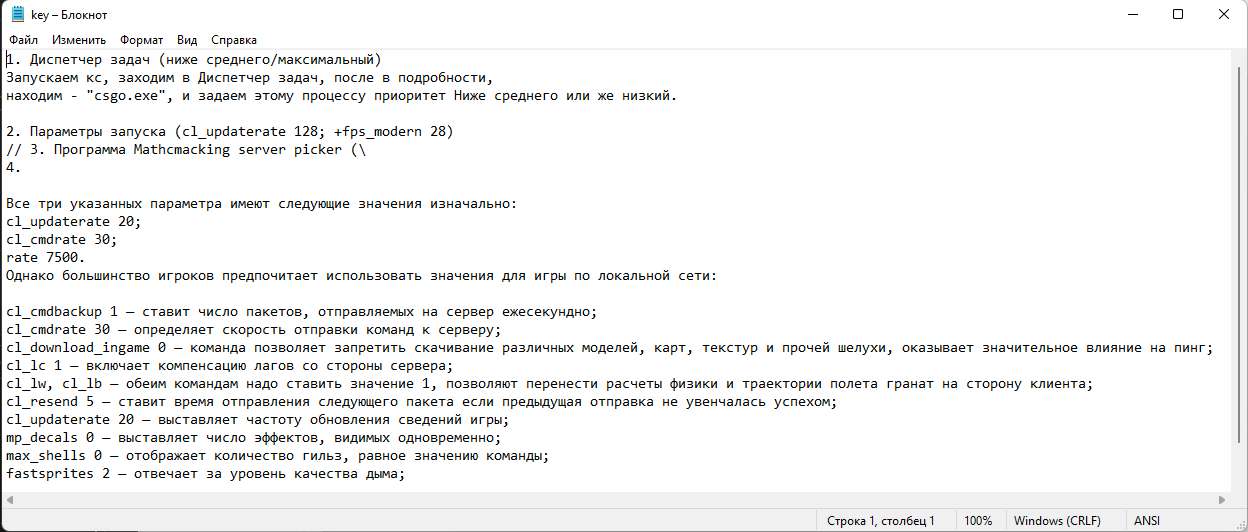
}

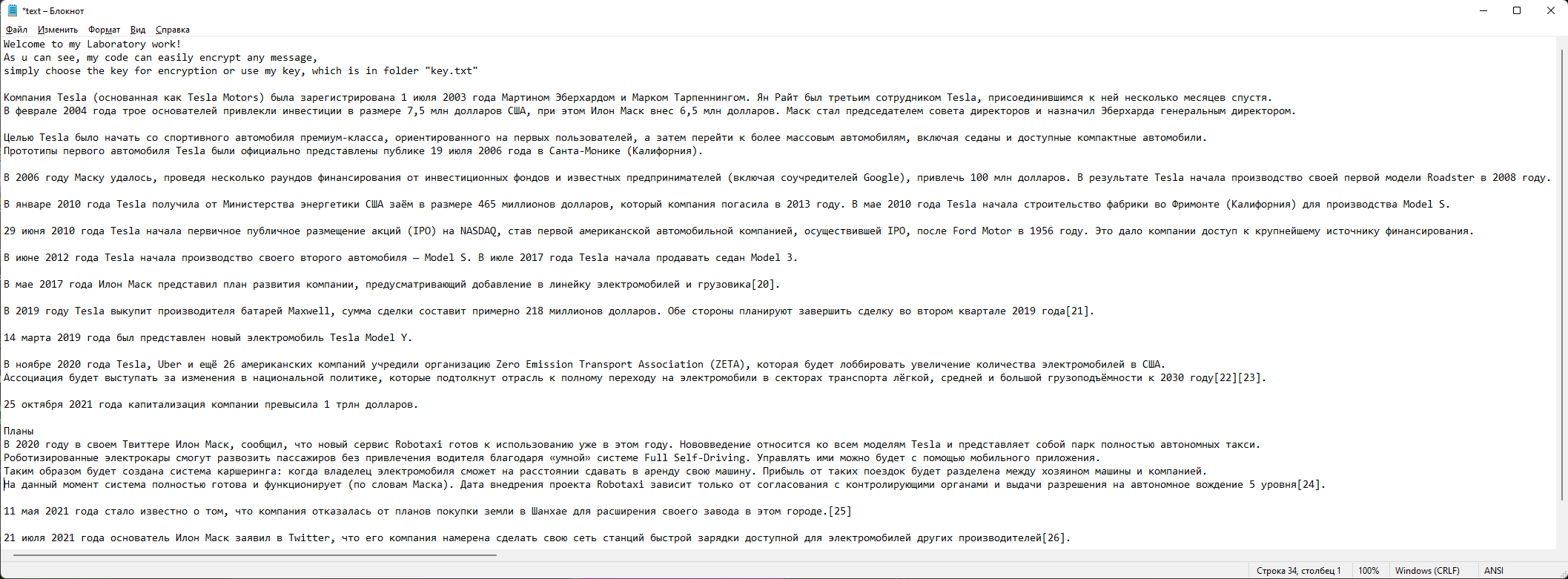
delete[] key;

return 0;

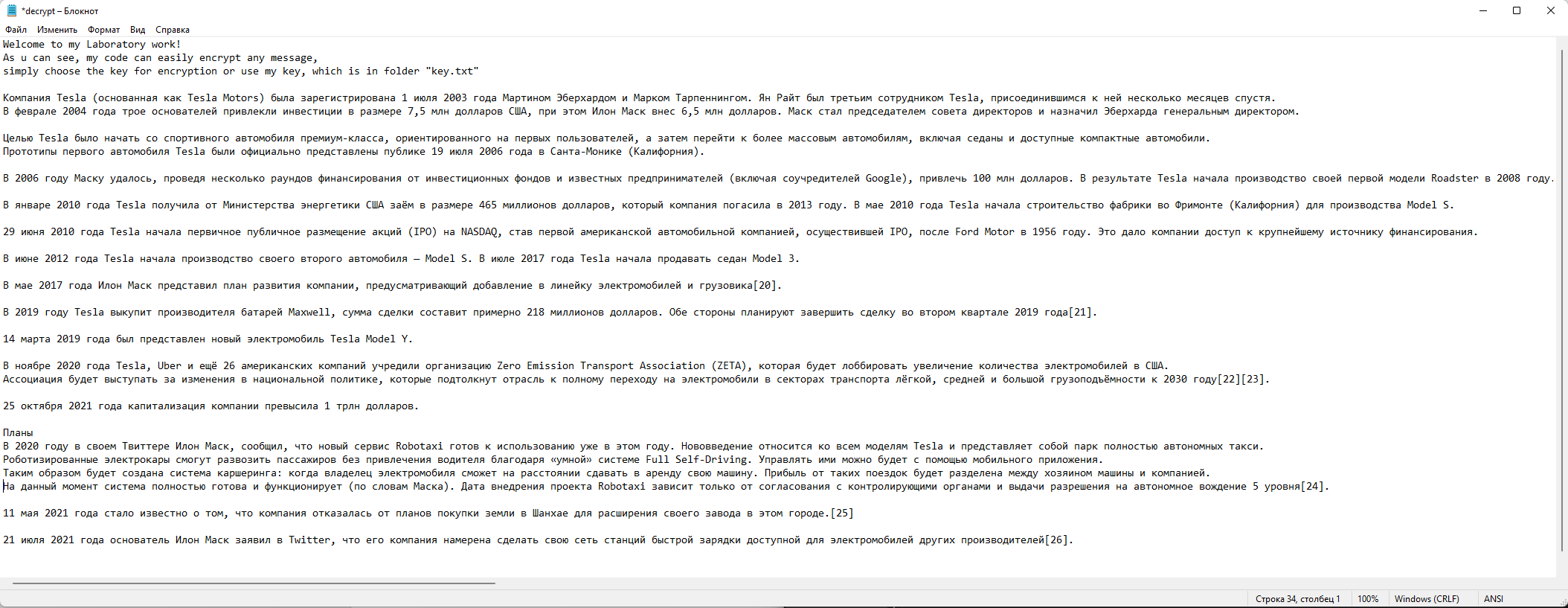
}

**Тестирование:**

****

****



****

