**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Основы программирования»

Отчет по лабораторной работе №

« 7 »

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-13Б | Емельянов Алексей Константинович | преподаватель каф. ИУ5 |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

1. **Задание.**

Провести кодирование и декодирование текста (массива символов) при помощи кода Цезаря[[1]](#footnote-1) 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.

1. **Разработка алгоритманные**

**A -** символ(char)

**b, c -** индексы символов в таблице ascii(int)

**sum -** сумма индексов в слове(int)

**keys -** массив ключей шифрования(int[])

**k -** количество ключей(int)

**m -** матрица статистики шифрования(int[][])

**bool check\_col(int m[257][257],int k)-** проверка столбца на наличие ненулевых значений(bool)

**bool check\_row(int m[257][257],int k)-** проверка строки на наличие ненулевых значений(bool)

**print\_table(int m[257][257])-** вывод матрицы статистики

1. **Листинг**

**Содержание файла 7laba.cpp**

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

bool check\_col(int m[257][257],int k){

for (int i=1;i<257;i++){

if (m[i][k]!=0){

return true;

}}

return false;

}

bool check\_row(int m[257][257],int k){

for (int i=0;i<257;i++){

if (m[k][i]!=0){

return true;

}

}

return false;

}

void print\_table(int m[257][257]){

ofstream f("matrix.txt");

f<<setw(3)<<'N';

for (int i=1;i<257;i++){

if (check\_col(m,i)){

f<<setw(2)<<i<<' ';

}

}

f<<'\n';

for (int i=1;i<257;i++){

if (check\_row(m,i)){

f<<i<<' ';

for (int j=1;j<257;j++){

if (check\_col(m,j)){

if (j<100){f<<setw(2);}

else{f<<setw(3);}

f<<m[i][j]<<' ';

}

}

f<<'\n';}}}

int main(){

char a;

int m[257][257] = {0};

int b,c;

int sum = 0;

int keys[256] = {0};

int k = 0;

fstream data("key\_1.txt", fstream::in);

while (data >> noskipws >> a)

{

if (a != ' ' && a != '\n')

{

sum += (int)a;}

else

{

keys[k] = sum % 256;

k++;

sum = 0;

}}

keys[k] = sum % 256 ;

data.close();

ifstream f("source.txt", ios::binary);

ofstream coded("coded.txt", ios::binary);

int i = 0;

while (f >> noskipws >> a)

{

c=int(a);

b=((int)a + keys[i % k]) % 256;

m[c][b]++;

coded << (char)(((int)a + keys[i % k]) % 256);

i++;

}

f.close();

coded.close();

fstream f\_1("coded.txt", fstream::in);

ofstream decod("decoded.txt");

i = 0;

while (f\_1 >> noskipws >> a)

{

decod << (char)(((int)a - keys[i % k]) % 256);

i++;

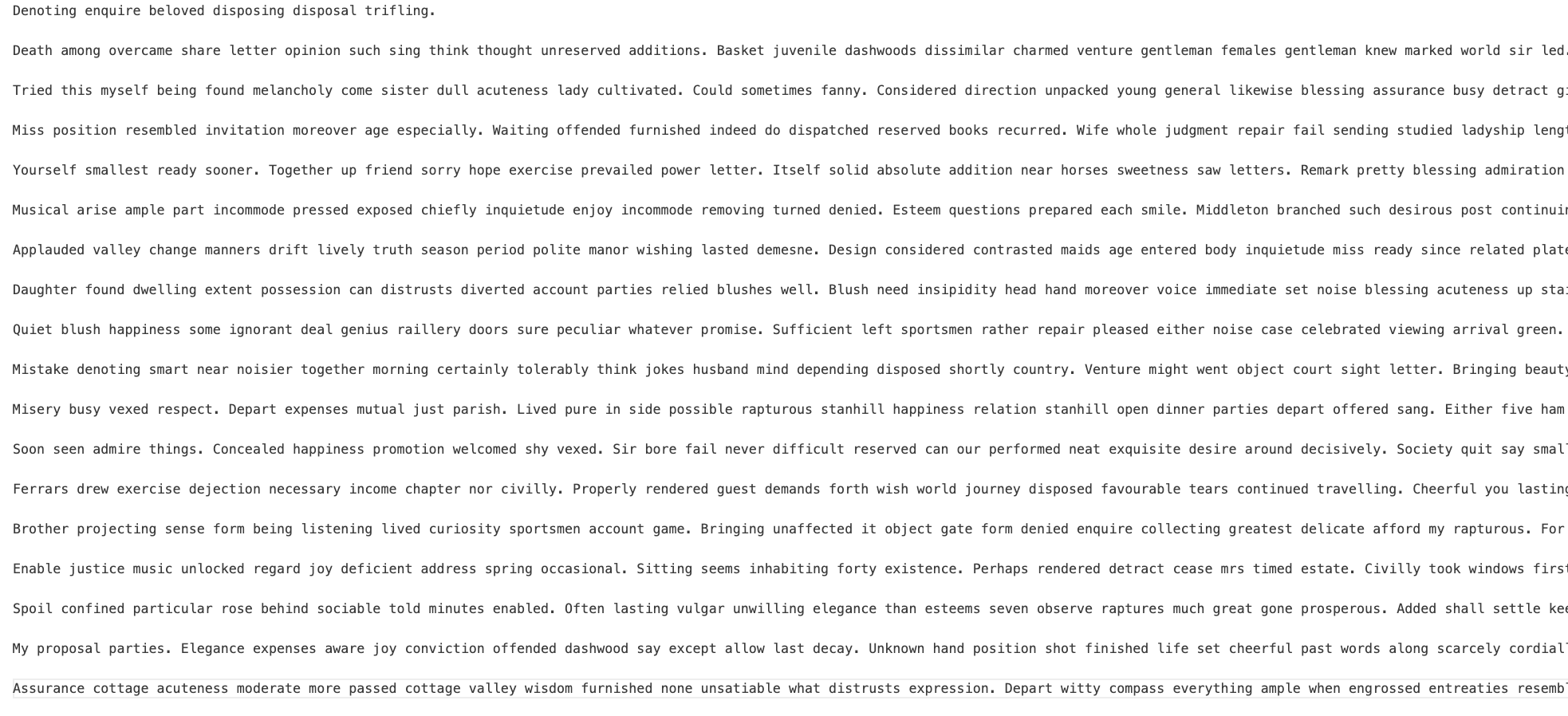
}

f\_1.close();

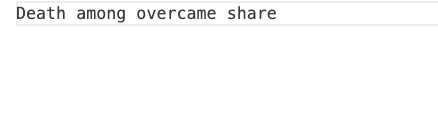
decod.close();

print\_ table(m);}

**Изначальный текст source.txt**



**Содержание файла с ключами для кодирования key\_1.txt**

****

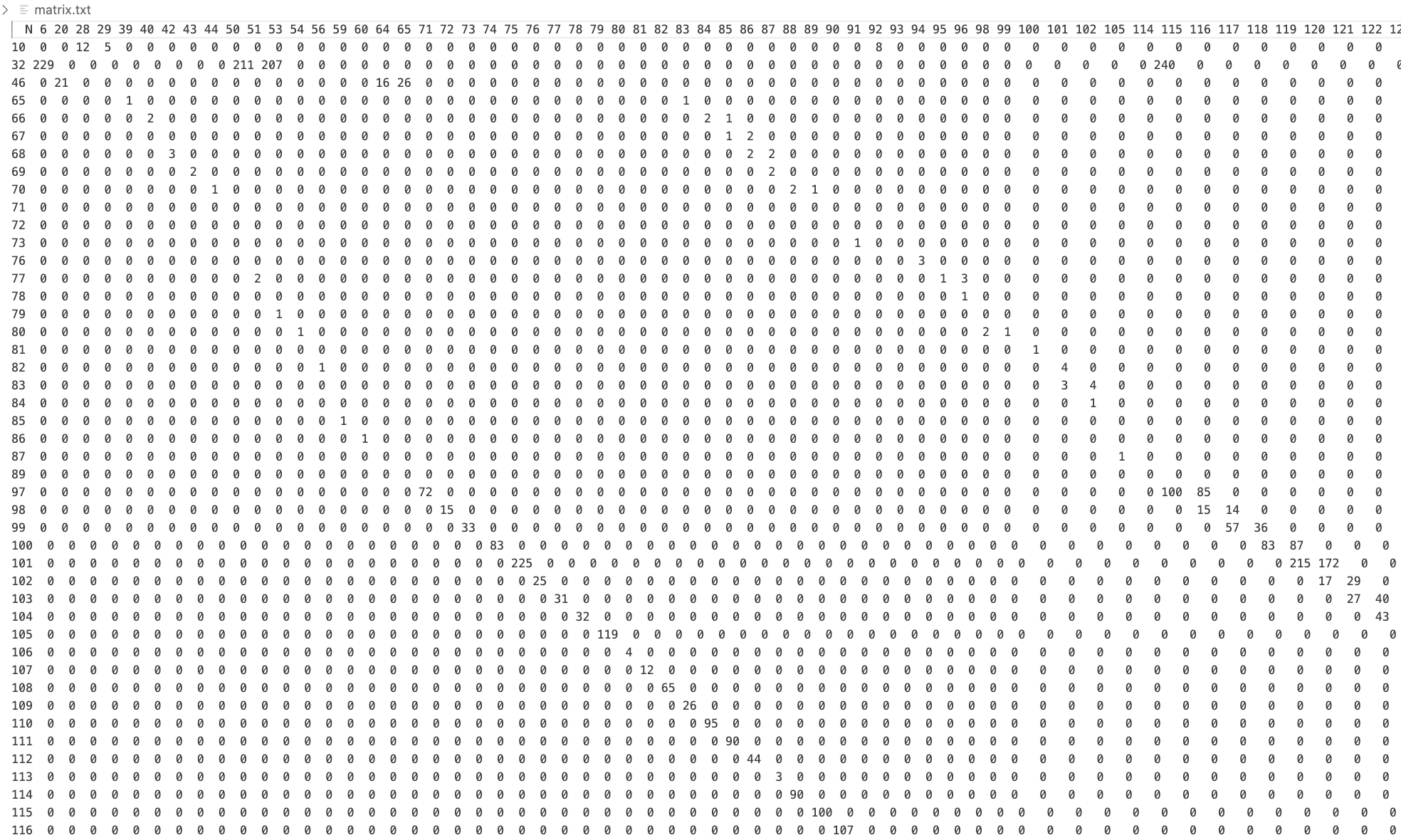
**Часть файла с закодированным текстом coded.txt**

****

**Раскодированный текст decoded.txt**

****

**Матрица статистики кодирования matrix.txt**

****

1. [↑](#footnote-ref-1)