ЗАШИФРОВКА

#include <iostream>

#include <Windows.h>

#include <string>

#include <fstream>

using namespace std;

//функция возвращает индекс буквы в алфавите

int find\_number(char l, string alp)

{

for (int a = 0; a < alp.length(); ++a)

{

if (alp[a] == l)

return a;

}

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

// открываем файл и считываем открытый текст

ifstream in("D:\\source\\Crypto\_lab2\_part1\\Text.txt");

string line, line1;

if (in.is\_open())

{

while (getline(in, line1))

line += line1;

}

else

{

cout << "Не удалось открыть файл...\n";

return 0;

}

cout << "Исходный текст: " << endl << endl;

cout << line << endl << endl;

string alphabet = "абвгдежзийклмнопрстуфхцчшщъыьэюя";

string keys[5] = { "ку", "мда", "слон", "крыса", "попугайвернулся" }; //ключи разной длины

for (int a = 0; a < 5; ++a) //проходим по всем ключам

{

string r\_s;

int len = 0; //переменная запоминает индекс буквы секретного ключа

cout << "Длина ключа: " << keys[a].length() << endl << endl;

for (int b = 0; b < line.length(); ++b) //проходим по всей строке

{

if (len >= keys[a].length()) //дойдя до конца строки с секретным ключем, начинаем заново

len = 0;

int num = find\_number(line[b], alphabet) + find\_number(keys[a][len], alphabet); //ищем индекс новой буквы

if (num < 0)

num += 32;

if (num > 31)

num -= 32;

r\_s += alphabet[num]; //формируем шифротекст

++len;

}

cout << r\_s << endl << endl;

}

}

РАСШИФРОВКА

#include <fstream>

#include <iostream>

#include <Windows.h>

#include <string>

#include <vector>

using namespace std;

class Letter

{

private:

char letter;

int number;

public:

Letter(char l) : letter(l), number(0) {}

char getLetter() { return letter; }

int getNumber() { return number; }

void incNumber() { ++number; }

};

// функция определяет индекс буквы в алфавите

int find\_number(char l, string alp)

{

for (int a = 0; a < alp.length(); ++a)

{

if (alp[a] == l)

return a;

}

}

// функция возвращает массив с данными о всех буквах и кол-ве их повторений в строке

vector<Letter> count\_letters(string str)

{

vector<Letter>letters; //массив со всеми найденными буквами и кол-вом их повторений

for (int a = 0; a < str.length(); ++a) //проходим по всей строке

{

if (letters.size() == 0)

letters.push\_back(str[a]); //добавляем первую букву

else

{

int d = 0;

for (int b = 0; b < letters.size(); ++b) //сверяемся со всеми найденными ранее буквами

{

if (letters[b].getLetter() == str[a]) //если такая буква уже есть в массиве, увеличиваем ее счетчик на 1 и выходим из цикла

{

letters[b].incNumber();

d = 1;

break;

}

}

if (d == 0) //если такой буквы нет в массиве, добавляем ее

letters.push\_back(str[a]);

}

}

return letters;

}

// функция ищет наиболее частую букву в строке

char find\_letter(string str)

{

vector<Letter>letters = count\_letters(str); //получаем массив для обработки

char let = '0';

int c = 0;

for (int a = 0; a < letters.size(); ++a) //проходим по всему массиву и выискиваем букву с наибольшим счетчиком

{

if (letters[a].getNumber() > c)

{

c = letters[a].getNumber();

let = letters[a].getLetter();

}

}

return let;

}

//определяет энтропию строки (для распознавателя русского языка)

double entrophy(string line)

{

int a = 0;

vector<Letter>let = count\_letters(line); //получаем массив для обработки

double ent = 0;

for (int c1 = 0; c1 < let.size(); ++c1) //высчитываем значение Н (аналогично 1-й лабе)

{

double d = (double)let[c1].getNumber() / line.length();

if (d != 0)

ent += (d \* log2(d));

}

ent = -ent;

return ent;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

// СЧИТЫВАЕМ ШИФРОВАННЫЙ ТЕКСТ ИЗ ФАЙЛА

ifstream in("D:\\source\\Crypto\_lab2\\Text.txt");

string line, line1;

if (in.is\_open())

{

while (getline(in, line1))

line +=line1;

}

else

{

cout << "Не удалось открыть файл...\n";

return 0;

}

// ПОИСК КЛЮЧА R

cout << "Начинаю поиск длины ключа...\n\n";

int a = 0, b = 0, counter = 0, max\_counter = 0, r = 6, max\_r = 6;

vector<string>main\_; //здесь будут хранится все фрагменты текста, расшифрованного правильным ключом

for (; r < 31; ++r) //проверяем все ключи длиной от 6 до 30 символов

{

cout << "Проверяю длину ключа: " << r << endl;

vector<string>fr; // сразу сохраняем в векторный массив все фрагменты текса для дальнейшей расшифровки

while (b < r)

{

fr.push\_back("");

fr[b] += line[a + b];

while (a + b + r < line.length())

{

if (line[a + b] == line[a + b + r]) //считаем кол-во идущих подряд повторяющихся символов

++counter;

fr[b] += line[a + b + r]; //записываем текст из символов, находящихся на расстоянии r

a += r;

}

a = 0; ++b;

}

cout << "Кол-во повторений: " << counter << endl << endl;

if (counter > max\_counter) //если счетчик повторяющихся символов больше - обновляем данные

{

max\_counter = counter; //сохраняем максимальное количество повторений

max\_r = r; //ключ

main\_ = fr; //и фрагментированный текст

}

counter = 0;

b = 0;

}

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n";

cout << "ОПРЕДЕЛЕНА ДЛИНА КЛЮЧА: " << max\_r << endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n";

// РАСШИФРОВКА ЦЕЗАРЯ

string alphabet = "абвгдежзийклмнопрстуфхцчшщъыьэюя";

int main\_letters[3] = {14, 5, 0}; //индексы самых распостраненных букв русского языка: о (14), е (5), а (0)

string en\_text; //здесь будет хранится расшифрованный текст

for (int c1 = 0; c1 < max\_r; ++c1) //проходим по всем фрагментам

{

char l1 = find\_letter(main\_[c1]); //находим самую частую букву во фрагменте

int num = find\_number(l1, alphabet); //определяем ее индекс

double e\_n = 10; string e\_str;

for (int c2 = 0; c2 < 3; ++c2) //проверяем все 3 буквы (о, е, а)

{

int c3 = 0;

int num1 = -(num - main\_letters[c2]); //находим предполагаемый ключ Цезаря

string str; //здесь будет хранится расшифрованная строка

for (c3 = 0; c3 < main\_[c1].length(); ++c3) //расшифровываем строку предполагаемым ключем

{

int num2 = find\_number(main\_[c1][c3], alphabet);

num2 += num1;

if (num2 >= 32)

num2 -= 32;

else if (num2 < 0)

num2 += 32;

str += alphabet[num2];

}

if (c1 == 0) //если это 1-я строка - оставляем ее с ключем "о" и переходим к следующей

{

en\_text += str;

break;

}

double entr = entrophy(en\_text + str); //считаем энтропию полученной строки вместе с расшифрованным текстом

if (entr < e\_n) // строка с наименьшим значением энтропии наиболее вероятно будет русским текстом

{

e\_n = entr;

e\_str = str;

}

}

string txt, txt1;

int e1 = 0; int e2 = 0;

for (int c4 = 0; c4 < (en\_text + e\_str).length(); ++c4) //склеиваем расшифрованную строку с предыдущим текстом, сразу расставляя буквы на свои места

{

if (((c4 + 1) % (c1 + 1)) == 0 && e2 < e\_str.length())

{

txt += e\_str[e2];

++e2;

}

else

{

txt += en\_text[e1];

++e1;

}

}

en\_text = txt;

}

cout << "Полученный текст: \n";

cout << en\_text << endl << endl;

}