ФГБОУ ВПО «Чувашский Государственный Университет им. И.Н.Ульянова»

Факультет информатики и вычислительной техники

Кафедра вычислительной техники

**Лабораторная работа №3  
“ЛИНЕЙНОЕ ШИФРОВАНИЕ ДАННЫХ (ГАММИРОВАНИЕ)”**

Выполнил: студент группы ИВТ-21-12  
 Чесноков Александр  
 Проверил: Павлов П.В.

Чебоксары, 2015

**Цель работы**

Целью работы является знакомство с классическим криптографическим алгоритмом - алгоритмом линейного шифрования данных (шифрования гаммированием).

**Индивидуальное задание**

| **№ п.п.** | ***A*** | ***C*** | ***T(0)*** |
| --- | --- | --- | --- |
| **1** | **5** | **51** | **13** |

**Функция шифрования**

Задаю параметры по своему варианту:

const int T0 = 13, B = 256;

const int A = 5, C = 51;

Затем считываю посимвольно текст из файла и отправляю его в функцию шифрования:

char Encrypt(char chBuff, int Gamma);

Функция шифрования производит замену путем наложения гаммы и возвращает новый символ:

char Encrypt(char chBuff, int Gamma)

{

int symbol = ((int)chBuff + Gamma) % B;

return (char)symbol;

}

Затем гамма записывается в файл и символ также записывается в другой файл.  
Далее с помощью датчика ПСЧ генерирую новую гамму и повторяю операцию пока не пройду по всему файлу.

**Текст программы, реализующей индивидуальное задание**

#include <iostream>

#include <fstream>

using namespace std;

const int T0 = 13, B = 256;

const int A = 5, C = 51;

char chBuff;

ifstream inFile;

ofstream outFile, gamma;

int PSCh(int Gamma);

char Encrypt(char chBuff, int Gamma);

char Decrypt(char chBuff, int Gamma);

void main()

{

setlocale(LC\_ALL, "Russian");

int Gamma = T0;

inFile.open("Сообщение.txt");

outFile.open("Шифрограмма.txt");

gamma.open("Гамма.txt");

while(inFile.get(chBuff))

{

cout << chBuff;

chBuff = Encrypt(chBuff, Gamma);

gamma << Gamma << " ";

Gamma = PSCh(Gamma);

outFile << (char)chBuff;

}

outFile.close();

inFile.close();

cout << "\n\n";

Gamma = T0;

inFile.open("Шифрограмма.txt");

outFile.open("Дешифрованный текст.txt");

while (inFile.get(chBuff))

{

cout << chBuff;

chBuff = Decrypt(chBuff, Gamma);

cout << "(" << chBuff << ") ";

Gamma = PSCh(Gamma);

outFile << chBuff;

}

outFile.close();

system("pause");

}

int PSCh(int Gamma)

{

return (A\*Gamma + C) % B;

}

char Encrypt(char chBuff, int Gamma)

{

int symbol = ((int)chBuff + Gamma) % B;

return (char)symbol;

}

char Decrypt(char chBuff, int Gamma)

{

int symbol = ((int)chBuff - Gamma) % B;

if (symbol < 0)

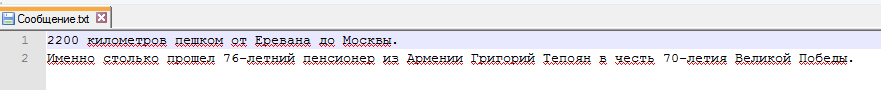
symbol += B;

return (char)symbol;

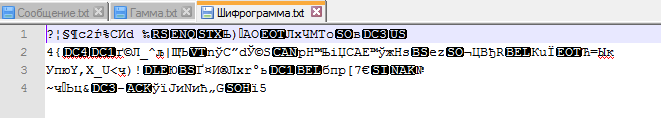
}

**Пример открытого текста и соответствующей ему шифрограммы**

Открытый текст:



Шифрограмма:



Гамма:

13 116 119 134 209 72 155 58 85 220 127 174 153 48 35 226 157 68 135 214 97 24 171 138 229 172 143 254 41 0 51 50 45 20 151 38 241 232 187 218 117 124 159 78 185 208 67 130 189 228 167 118 129 184 203 42 5 76 175 158 73 160 83 210 77 180 183 198 17 136 219 122 149 28 191 238 217 112 99 34 221 132 199 22 161 88 235 202 37 236 207 62 105 64 115 114 109 84 215 102 49 40 251 26 181 188 223 142 249 16 131 194 253 36 231 182 193 248 11 106 69 140 239 222 137 224 147 18 141 244 247 6 81 200 27 186 213 92 255 46 25 176 163 98 29 196 7

**Вывод**

Я ознакомился с классическим криптографическим алгоритмом - алгоритмом линейного шифрования данных (шифрования гаммированием) и реализовал его в среде программирования VS2013 на языке C++.