МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра «Информатики и систем управления»

ОТЧЕТ

по лабораторной работе

по дисциплине

Методы и средства защиты информации

ПРЕПОДАВАТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Капранов С.Н. (подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гореев А.Д. (подпись) (фамилия, и.,о.)

19-ИСТ-2

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

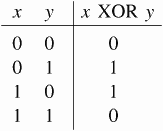
Нижний Новгород

2021

**Вариант № 7**

**Задача № 1** Реализовать алгоритм шифрования данных «Шифрование методом гаммирования» по модулю 2.

**Описание алгоритма.** Буквы исходного сообщения и гаммы представляются в виде двоичной последовательности. Далее с помощью операции XOR генерируется новое зашифрованное сообщение.



**Листинг программы**

#include <iostream>

#include <string>

using namespace std;

//Преобразование текста, с помощью исключающего или

//parameter\_1, parameter\_2 входные парамеры.parameter\_1 сообщение (стандартный вид, так и двоичный) parameter\_2 это гамма

string conversion\_XOR(string parameter\_1, string parameter\_2)

{

//Выходное значение

string parameter\_3;

int j = 0;

for (auto i = 0; i < parameter\_1.length(); i++)

{

//Преобразование символов сообщения в двоичный, с помощью исключающего или

parameter\_3 += (int)parameter\_1[i] ^ (int)parameter\_2[j];

j++;

//Если длина сообщения больше гаммы

if (j >= parameter\_2.length()) j = 0;

}

return parameter\_3;

}

int main()

{

string message;

string gamma;

cout << "Enter the message you want to encrypt : ";

getline(cin, message);

cout << "Enter the key : ";

getline(cin, gamma);

cout << "\n\t\tEncrypted text " << endl;

cout << conversion\_XOR(message, gamma) << endl;

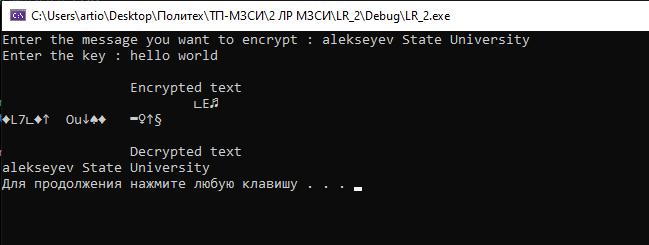
cout << "\n\t\tDecrypted text " << endl;

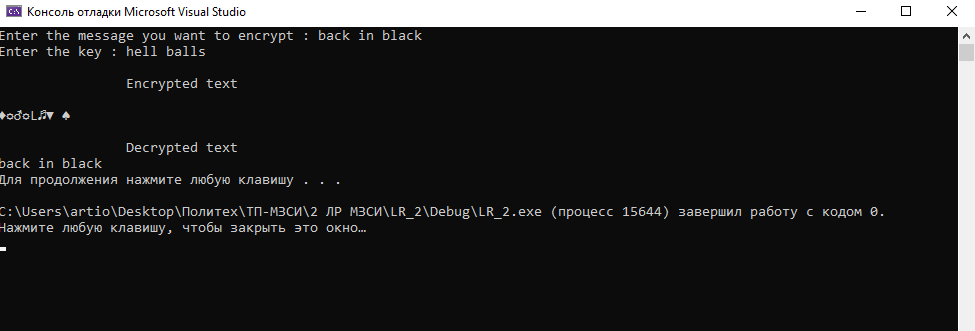
cout << conversion\_XOR(conversion\_XOR(message, gamma), gamma) << endl;

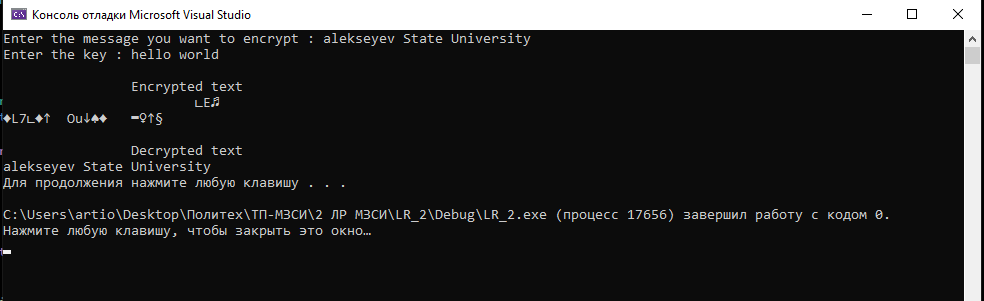
system("pause");

return 0;

}

**Примеры работы программы**

****

****