МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра информационной безопасности

**Отчет**

по лабораторной работе №6

на тему: «**Исследование эффекта вырождения схем усложнения, построенных на основе линейных регистров сдвига**»

по дисциплине «Информационная безопасность»

Выполнили: Кожухова О.А. Шифр: 170582

Шорин В.Д. Шифр: 171406

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 71-ПГ

Проверил: Еременко В.Т.

Отметка о зачете:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата «\_\_\_\_» \_\_\_\_\_\_ 2021г.

Орел, 2021 г.

**Задание**

1) Решить задачу из пункта 1 при следующих значениях параметров:

n = 11,

х = (10010010111) ,

е(х) = x1 x3 x7 x9 x10,

h(x) = x1 x3 x6 x9,

g(x) = x1 x6 x7 x9 x10.

2) Решить задачу из пункта 2 при следующих значениях параметров:

n = 11,

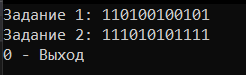
х = (11000101001),

у = (00010000111).

е(х) = x9 1.

3) Написать отчет.

**Ход работы**



**Код**

**«Program.cs»**

using System;

namespace IS\_L\_6

{

class Program

{

private static string x1 = "10010010111";

private static string x2 = "10010010111";

private static string y = "00010000111";

private static int n = 11;

static void Main(string[] args)

{

while (true)

{

Console.Clear();

Console.WriteLine($"Задание 1: {Task1()}");

Console.WriteLine($"Задание 2: {Task2()}");

Console.WriteLine("0 - Выход");

int res = Convert.ToInt32(Console.ReadLine());

switch (res)

{

case 0:

return;

default:

Console.WriteLine("Нет такой команды");

break;

}

}

}

private static string Task1()

{

string result = "";

byte currentElement, e, g, h;

byte[] Ae = new byte[n];

byte[] Ah = new byte[n];

byte[] Ag = new byte[n];

byte[] array = new byte[n];

for (int i = 0; i < n; i++)

{

Ae[i] = byte.Parse(x1[i].ToString());

Ah[i] = byte.Parse(x1[i].ToString());

Ag[i] = byte.Parse(x1[i].ToString());

}

for (int i = 0; i < n ; i++)

{

array = Ae;

e = (byte)(Ae[0] ^ Ae[2] ^ Ae[6] ^ Ae[8] ^ Ae[9]);

for (int j = n - 1; j > 1; j--)

Ae[j] = array[j - 1];

Ae[0] = e;

array = Ah;

h = (byte)(Ah[0] ^ Ah[2] ^ Ah[5] ^ Ah[8]);

for (int j = 0; j < n - 2; j++)

Ah[j + 1] = array[j];

Ah[n - 1] = h;

array = Ag;

g = (byte)(Ag[0] ^ Ag[5] ^ Ag[6] ^ Ag[8] ^ Ag[9]);

for (int j = 0; j < n - 2; j++)

Ag[j + 1] = array[j];

Ag[n - 1] = g;

currentElement = (byte)(Ae[n - 1] ^ Ah[0] ^ Ag[0]);

result += currentElement;

}

return result;

}

private static string Task2()

{

string result = "";

byte currentElement, e, h;

byte[] Ae = new byte[n];

byte[] Ah = new byte[n];

byte[] array = new byte[n];

for (int i = 0; i < n; i++)

{

Ae[i] = byte.Parse(x2[i].ToString());

Ah[i] = byte.Parse(y[i].ToString());

}

for (int i = 0; i < n; i++)

{

array = Ae;

e = (byte)(Ae[8] ^ 1);

for (int j = n - 1; j > 1; j--)

Ae[j] = array[j - 1];

Ae[0] = e;

array = Ah;

h = (byte)(Ah[8] ^ 1);

for (int j = n - 1; j > 1; j--)

Ah[j] = array[j - 1];

Ah[0] = h;

currentElement = (byte)(Ae[n - 1] ^ Ah[n - 1]);

result += currentElement;

}

return result;

}

}

}