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 №8

на тему: «**Изучение некоторых свойств линейных последовательных машин»**»

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) Составьте программу генерации выходных последовательностей ЛПМ.

С помощью датчика случайных чисел наработайте входную последовательность u(1), u(2), ... ,u(t) и сохраните ее для дальнейшего использования. Для простоты машинной реализации ограничим размерности векторов u(t), y(t) и s(t) восемью, и будем представлять вектора в байтовом виде.

Постройте таблицы результатов умножения характеристических матриц A=[a1,...,а8], B=[b1,...,bз], c=[c1,...,с8], D=[d1,...,d8] на всевозможные вектора (байты), где ai (bi, ci, di) есть байт, представляющий 1-ю строку матрицы А

(В, С. D), 1=1, 2, ... , 8.

Используя заданные матрицы a,b,c,d и входную последовательность из п.2.2, выработайте выходные последовательности для каждого из 255 возможных начальных состояний ЛПМ.

Отсортируйте полученные последовательности и определите число классов эквивалентных состояний ЛПМ.

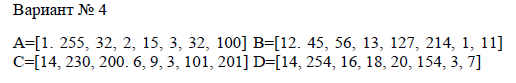
Повторите задание при другой входной последовательности и сравните полученные результаты.

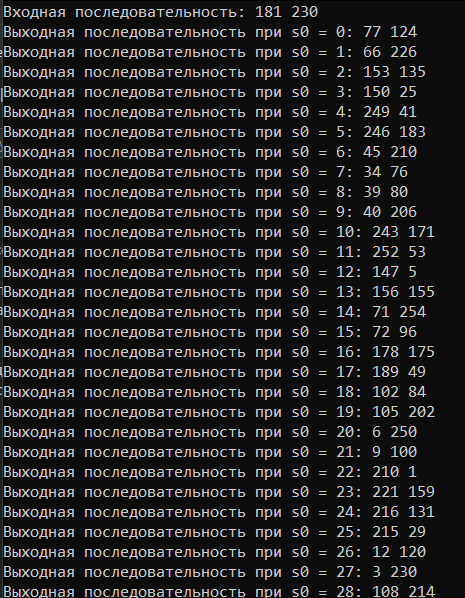
2) Выработайте выходные последовательности при каждом начальном состоянии ЛПМ и одинаковой входной последовательности.

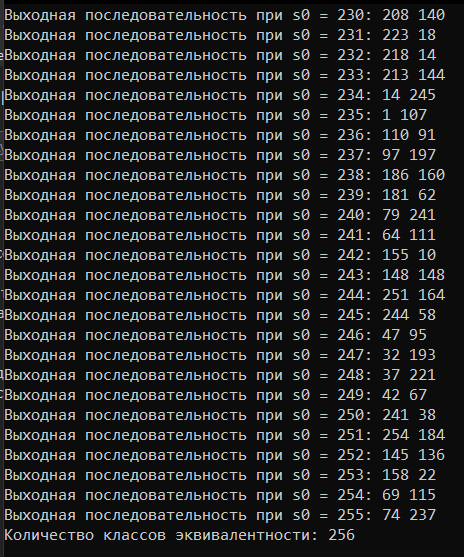
3) Определите число классов эквивалентных состояний.

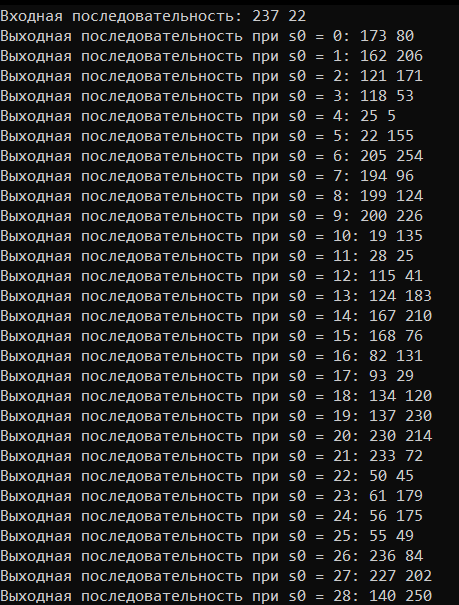
4) Повторите пункты 1.2 и 1.3 с другой входной последовательностью, проанализируйте полученные результаты

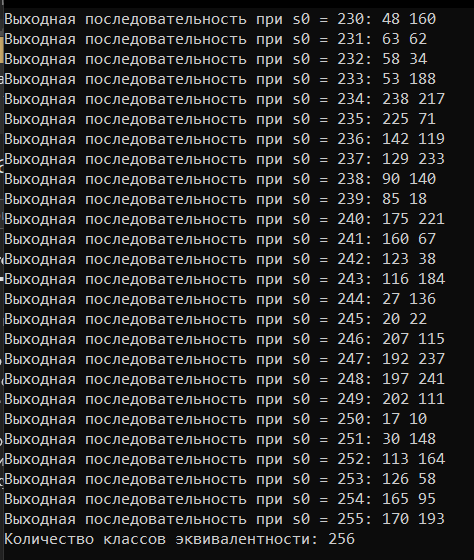
**Ход работы**











**Код**

using System;

using System.Collections.Generic;

namespace IS\_L\_8

{

class Program

{

private static int t = 2;

private static int n = 8;

static void Main(string[] args)

{

int[] a = new int[] { 1, 255, 32, 2, 15, 3, 32, 100 };

int[] b = new int[] { 12, 45, 56, 13, 127, 214, 1, 11 };

int[] c = new int[] { 14, 230, 200, 6, 9, 3, 101, 201 };

int[] d = new int[] { 14, 254, 16, 18, 20, 154, 3, 7 };

int[] u = GenerateU(n, t);

Console.Write("Входная последовательность: ");

foreach (var item in u)

{

Console.Write(item + " ");

}

Console.WriteLine();

Dictionary<int[], int> m0 = new Dictionary<int[], int>();

for (int i = 0; i < 256; i++)

{

int[] answer = Solve(t, i, u, a, b, c, d);

if (m0.ContainsKey(answer))

{

Console.WriteLine("Contains");

m0[answer]++;

}

else

{

m0.Add(answer, 0);

}

Console.Write($"Выходная последовательность при s0 = {i}: ");

foreach (var item in answer)

Console.Write(item + " ");

Console.WriteLine();

}

Console.WriteLine("Количество классов эквивалентности: " + m0.Count);

}

private static int[] Solve(int t, int s0, int[] u, int[] a, int[] b, int[] c, int[] d)

{

int[] s = new int[t + 1];

s[0] = s0;

int[] answer = new int[t];

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

{

s[i + 1] = T(Foo(

Dot(a, T(new int[] { s[i] }, n), 1),

Dot(b, T(new int[] { u[i] }, n), 1),

1), 1

)[0];

answer[i] = T(Foo(

Dot(c, T(new int[] { s[i] }, n), 1),

Dot(d, T(new int[] { u[i] }, n), 1),

1), 1

)[0];

}

return answer;

}

private static int[] T(int[] a, int n)

{

int[] result = new int[n];

for (int i = 0; i < a.Length; i++)

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

result[j] |= (Get(a[i], j) << i);

return result;

}

private static int[] Foo(int[] a, int[] b, int n)

{

int[] result = new int[a.Length];

for (int i = 0; i < result.Length; i++)

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

result[i] |= ((Get(a[i], j) ^ Get(b[i], j)) << j);

return result;

}

private static int[] Dot(int[] a, int[] b, int n)

{

int[] result = new int[a.Length];

for (int i = 0; i < a.Length; i++)

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

for (int k = 0; k < b.Length; k++)

result[i] ^= ((Get(a[i], k) \* Get(b[k], j)) << j);

return result;

}

private static int Get(int a, int i) => ((a & (1 << i)) != 0) ? 0 : 1;

private static int[] GenerateU(int n, int t)

{

int[] result = new int[t];

Random random = new Random();

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

result[i] = random.Next(0, 1 << n);

return result;

}

}

}