**Министерство образования и науки Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение

Высшего образования

**«Владимирский государственный университет**

**Имени Александра Григорьевича и Николая Григорьевича Столетовых»**

(ВлГУ)

Институт информационных технологий и электроники

Кафедра «Физика и прикладная математика»

**ЛАБОРАТОРНАЯ РАБОТА №4**

По дисциплине   
«Дискретная математика»

На тему:

**«Построение матрицы бинарного отношения»**

Выполнил:

ст. гр.ПМИ-123

Рушев А.М.

Принял:

к.т.н., доцент каф. ФиПМ

Горлов В.Н.

Владимир, 2025

**Цель работы:** изучить способы моделирования матрицы бинарного отношения множеств и написать компьютерную программу для построения матрицы бинарного отношения двух конечных множеств.

**Теоретическая часть:**

Бинарное отношение — это отношение между элементами двух множеств. Если у нас есть множество A, состоящее из n элементов, и множество B, состоящее из m элементов, то бинарное отношение R между множествами A и B можно представить в виде упорядоченной пары: R ⊆ A × B.

Построение матрицы бинарного отношения

Для представления бинарного отношения в виде матрицы, необходимо выполнить следующие шаги:

1. Определение множеств: Пусть A = { a₁, a₂, …, aₙ } и B = { b₁, b₂, …, bₘ } .

2. Создание матрицы: Создаем матрицу M размером n × m , где строки соответствуют элементам множества A , а столбцы — элементам множества B .

3. Заполнение матрицы:

Если элемент aᵢ из множества A связан с элементом bⱼ из множества B (то есть (aᵢ, bⱼ) ∈ R ), то в ячейке M[i][j] записываем 1.

Если нет связи (то есть (aᵢ, bⱼ) ∉ R ), то в ячейке записываем 0.

Таким образом, матрица бинарного отношения будет состоять только из 0 и 1.

Пример

Рассмотрим множество A = { 1, 2, 3 } и множество B = { x, y } . Пусть бинарное отношение R = { (1, x), (2, y) } .

1. Определяем размеры матрицы:

Строки: 3 (элементы множества A )

Столбцы: 2 (элементы множества B )

2. Создаем матрицу M :

M =

? | ?

? | ?

? | ?

3. Заполняем матрицу:

Для (1, x) : M[1][1] = 1

Для (2, y) : M[2][2] = 1

Все остальные элементы будут равны 0.

Итоговая матрица будет выглядеть так:

M =

1 | 0

0 | 1

0 | 0

Матрица бинарного отношения позволяет удобно представлять связи между элементами различных множеств и может быть использована для различных задач в теории графов, комбинаторике и других областях математики.

**Ход работы:**

Листинг приложения:

**MainForm.cs**

using System;

using System.Collections.Generic;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Windows.Forms;

namespace Lab4\_CM\_WinForms

{

public partial class MainForm : Form

{

#region Fields

private Random rnd = new Random();

private Int32[] massA = new Int32[0];

private Int32[] massB = new Int32[0];

private Int32[][] massAll = new Int32[0][];

private Int32 numberSelected = 0;

private DataTable dataTable = new DataTable();

#endregion

public MainForm()

{

InitializeComponent();

}

private void MainForm\_Load(object sender, EventArgs e)

{

tbxA.Text = "Введите число";

tbxA.ForeColor = Color.Gray;

tbxB.Text = "Введите число";

tbxB.ForeColor = Color.Gray;

btnCreateA.Enabled = false;

btnCreateB.Enabled = false;

}

#region ButoonsActions

private void btnCreateA\_Click(object sender, EventArgs e)

{

CreateMass(tbxA.Text, tbxA.Text.Count(), massA, "A");

}

private void btnCreateB\_Click(object sender, EventArgs e)

{

CreateMass(tbxB.Text, tbxB.Text.Count(), massB, "B");

}

private void btnClearAll\_Click(object sender, EventArgs e)

{

tbxA.Clear();

tbxB.Clear();

rTbxPrintInfo.Clear();

dgvPrintInfo.Columns.Clear();

tbxA\_Leave(sender, e);

tbxB\_Leave(sender, e);

rBtnFuncFirst.Checked = false;

rBtnFuncSecond.Checked = false;

rBtnFuncThird.Checked = false;

rBtnFuncFourth.Checked = false;

rBtnFuncFivth.Checked = false;

}

#endregion

#region Methods

//создание массивов

private void CreateMass(string tbx, int countTbx, int[] mass, string nameRTbxPrintInfo)

{

try

{

bool isValidate = true;

if (!Int32.TryParse(tbx, out countTbx))

{

if (tbx == tbxA.Text)

{

tbxA.Clear();

MessageBox.Show($"Число должно быть целым");

isValidate = false;

}

if (tbx == tbxB.Text)

{

tbxB.Clear();

MessageBox.Show($"Число должно быть целым");

isValidate = false;

}

}

if (isValidate)

{

HashSet<Int32> list = new HashSet<Int32>();

while (list.Count < countTbx)

{

list.Add(rnd.Next(1, 100));

}

mass = list.ToArray();

Array.Sort(mass);

switch (nameRTbxPrintInfo)

{

case "A":

massA = mass;

break;

case "B":

massB = mass;

break;

}

rTbxPrintInfo.AppendText($"\nМножество {nameRTbxPrintInfo}\n");

foreach (var item in mass)

{

rTbxPrintInfo.AppendText($"{item.ToString()}\t");

}

}

}

catch (Exception ex)

{

MessageBox.Show($"{ex.Message}");

}

}

//создание строк и столбцов в таблице

private void CreateDataGridInfo()

{

try

{

dataTable.Clear();

dataTable.Columns.Clear();

foreach (var item in massB)

{

dataTable.Columns.Add(item.ToString());

}

dgvPrintInfo.DataSource = dataTable;

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

{

dataTable.Rows.Add();

if (i < dgvPrintInfo.Rows.Count)

{

dgvPrintInfo.Rows[i].HeaderCell.Value = massA[i].ToString();

}

}

}

catch (Exception ex)

{

MessageBox.Show($"{ex.Message}");

}

}

//расчет матрицы

public int[][] InfoDataMatrix(int[] mas1, int[] mas2, int N, int M)

{

List<int[]> res = new List<int[]>();

switch (numberSelected)

{

case 1:

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

{

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

{

if (mas1[i] < mas2[j])

{

res.Add(new int[] { mas1[i], mas2[j] });

}

}

}

break;

case 2:

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

{

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

{

if (mas1[i] == mas2[j])

{

res.Add(new int[] { mas1[i], mas2[j] });

}

}

}

break;

case 3:

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

{

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

{

if (mas1[i] % mas2[j] == 0)

{

res.Add(new int[] { mas1[i], mas2[j] });

}

}

}

break;

case 4:

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

{

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

{

if (mas1[i] % 2 == 0 || mas2[j] % 2 == 0)

{

res.Add(new int[] { mas1[i], mas2[j] });

}

}

}

break;

case 5:

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

{

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

{

if (mas1[i] % 2 == 0 && mas2[j] % 2 == 0)

{

res.Add(new int[] { mas1[i], mas2[j] });

}

}

}

break;

case 6:

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

{

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

{

if (mas1[i] != 0 && mas2[j] != 0)

{

if (GCD(mas1[i], mas2[j]) > 1)

{

res.Add(new int[] { mas1[i], mas2[j] });

}

}

}

}

break;

default: break;

}

return res.ToArray();

}

private int GCD(int a, int b)

{

while (b != 0)

{

int temp = b;

b = a % b;

a = temp;

}

return Math.Abs(a);

}

//занесение в таблицу

private void InsertInfoTable()

{

try

{

if (massA.Length == 0 || massB.Length == 0)

{

MessageBox.Show($"Нужно создать множество A и B");

}

else

{

switch (numberSelected)

{

case 1:

var resultFirstFunc = InfoDataMatrix(massA, massB, massA.Length, massB.Length);

rTbxPrintInfo.AppendText($"\nОтношение {rBtnFuncFirst.Text}\n");

foreach (var item in resultFirstFunc)

{

rTbxPrintInfo.AppendText($"<{string.Join(", ", item)}>\t");

}

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

{

for (int j = 0; j < massB.Length; j++)

{

if (massA[i] < massB[j])

{

dgvPrintInfo.Rows[i].Cells[j].Value = 1;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Green;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

else

{

dgvPrintInfo.Rows[i].Cells[j].Value = 0;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Red;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

}

}

break;

case 2:

var resultSecondFunc = InfoDataMatrix(massA, massB, massA.Length, massB.Length);

rTbxPrintInfo.AppendText($"\nОтношение {rBtnFuncSecond.Text}\n");

foreach (var item in resultSecondFunc)

{

rTbxPrintInfo.AppendText($"<{string.Join(", ", item)}>\t");

}

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

{

for (int j = 0; j < massB.Length; j++)

{

if (massA[i] == massB[j])

{

dgvPrintInfo.Rows[i].Cells[j].Value = 1;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Green;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

else

{

dgvPrintInfo.Rows[i].Cells[j].Value = 0;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Red;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

}

}

break;

case 3:

var resultThirdFunc = InfoDataMatrix(massA, massB, massA.Length, massB.Length);

rTbxPrintInfo.AppendText($"\nОтношение {rBtnFuncThird.Text}\n");

foreach (var item in resultThirdFunc)

{

rTbxPrintInfo.AppendText($"<{string.Join(", ", item)}>\t");

}

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

{

for (int j = 0; j < massB.Length; j++)

{

if (massA[i] % massB[j] == 0)

{

dgvPrintInfo.Rows[i].Cells[j].Value = 1;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Green;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

else

{

dgvPrintInfo.Rows[i].Cells[j].Value = 0;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Red;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

}

}

break;

case 4:

var resultFourthFunc = InfoDataMatrix(massA, massB, massA.Length, massB.Length);

rTbxPrintInfo.AppendText($"\nОтношение {rBtnFuncFourth.Text}\n");

foreach (var item in resultFourthFunc)

{

rTbxPrintInfo.AppendText($"<{string.Join(", ", item)}>\t");

}

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

{

for (int j = 0; j < massB.Length; j++)

{

if (massA[i] % 2 == 0 || massB[j] % 2 == 0)

{

dgvPrintInfo.Rows[i].Cells[j].Value = 1;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Green;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

else

{

dgvPrintInfo.Rows[i].Cells[j].Value = 0;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Red;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

}

}

break;

case 5:

var resultFivethFunc = InfoDataMatrix(massA, massB, massA.Length, massB.Length);

rTbxPrintInfo.AppendText($"\nОтношение {rBtnFuncFivth.Text}\n");

foreach (var item in resultFivethFunc)

{

rTbxPrintInfo.AppendText($"<{string.Join(", ", item)}>\t");

}

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

{

for (int j = 0; j < massB.Length; j++)

{

if (massA[i] % 2 == 0 && massB[j] % 2 == 0)

{

dgvPrintInfo.Rows[i].Cells[j].Value = 1;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Green;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

else

{

dgvPrintInfo.Rows[i].Cells[j].Value = 0;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Red;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

}

}

break;

case 6:

var resultSixFunc = InfoDataMatrix(massA, massB, massA.Length, massB.Length);

rTbxPrintInfo.AppendText($"\nОтношение {rBtnFuncSix.Text}\n");

var gcdResults = new List<int>();

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

{

for (int j = 0; j < massB.Length; j++)

{

var gcd = GCD(massA[i], massB[j]);

gcdResults.Add(gcd);

}

}

foreach (var item in resultSixFunc)

{

rTbxPrintInfo.AppendText($"<{string.Join(", ", item)}>\t");

}

rTbxPrintInfo.AppendText("\n");

foreach (var itemGCD in gcdResults)

{

if(itemGCD != 1)

{

rTbxPrintInfo.AppendText($"НОД: {itemGCD}\t");

}

}

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

{

for (int j = 0; j < massB.Length; j++)

{

if (GCD(massA[i], massB[j]) > 1)

{

dgvPrintInfo.Rows[i].Cells[j].Value = 1;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Green;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

else

{

dgvPrintInfo.Rows[i].Cells[j].Value = 0;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Red;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

}

}

break;

default: break;

}

}

}

catch (Exception ex)

{

MessageBox.Show($"{ex.Message}");

}

}

#endregion

#region TextBox A handlers

private void tbxA\_Enter(object sender, EventArgs e)

{

tbxA.Text = null;

tbxA.ForeColor = Color.Black;

}

private void tbxA\_Leave(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(tbxA.Text))

{

tbxA.Text = "Введите число";

tbxA.ForeColor = Color.Gray;

btnCreateA.Enabled = false;

}

}

private void tbxA\_TextChanged(object sender, EventArgs e)

{

if (!string.IsNullOrEmpty(tbxA.Text))

{

btnCreateA.Enabled = true;

}

else

{

btnCreateA.Enabled = false;

}

}

#endregion

#region TextBox B handlers

private void tbxB\_TextChanged(object sender, EventArgs e)

{

if (!string.IsNullOrEmpty(tbxB.Text))

{

btnCreateB.Enabled = true;

}

else

{

btnCreateB.Enabled = false;

}

}

private void tbxB\_Enter(object sender, EventArgs e)

{

tbxB.Text = null;

tbxB.ForeColor = Color.Black;

}

private void tbxB\_Leave(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(tbxB.Text))

{

tbxB.Text = "Введите число";

tbxB.ForeColor = Color.Gray;

btnCreateB.Enabled = false;

}

}

#endregion

#region RadioButtons handlers

private void rBtnFuncFirst\_CheckedChanged(object sender, EventArgs e)

{

if (rBtnFuncFirst.Checked)

{

numberSelected = 1;

CreateDataGridInfo();

InsertInfoTable();

}

}

private void rBtnFuncSecond\_CheckedChanged(object sender, EventArgs e)

{

if (rBtnFuncSecond.Checked)

{

numberSelected = 2;

CreateDataGridInfo();

InsertInfoTable();

}

}

private void rBtnFuncThird\_CheckedChanged(object sender, EventArgs e)

{

if (rBtnFuncThird.Checked)

{

numberSelected = 3;

CreateDataGridInfo();

InsertInfoTable();

}

}

private void rBtnFuncFourth\_CheckedChanged(object sender, EventArgs e)

{

if (rBtnFuncFourth.Checked)

{

numberSelected = 4;

CreateDataGridInfo();

InsertInfoTable();

}

}

private void rBtnFuncFivth\_CheckedChanged(object sender, EventArgs e)

{

if (rBtnFuncFivth.Checked)

{

numberSelected = 5;

CreateDataGridInfo();

InsertInfoTable();

}

}

private void rBtnFuncSix\_CheckedChanged(object sender, EventArgs e)

{

if (rBtnFuncSix.Checked)

{

numberSelected = 6;

CreateDataGridInfo();

InsertInfoTable();

}

}

#endregion

}

}

Результат выполнения приложения:

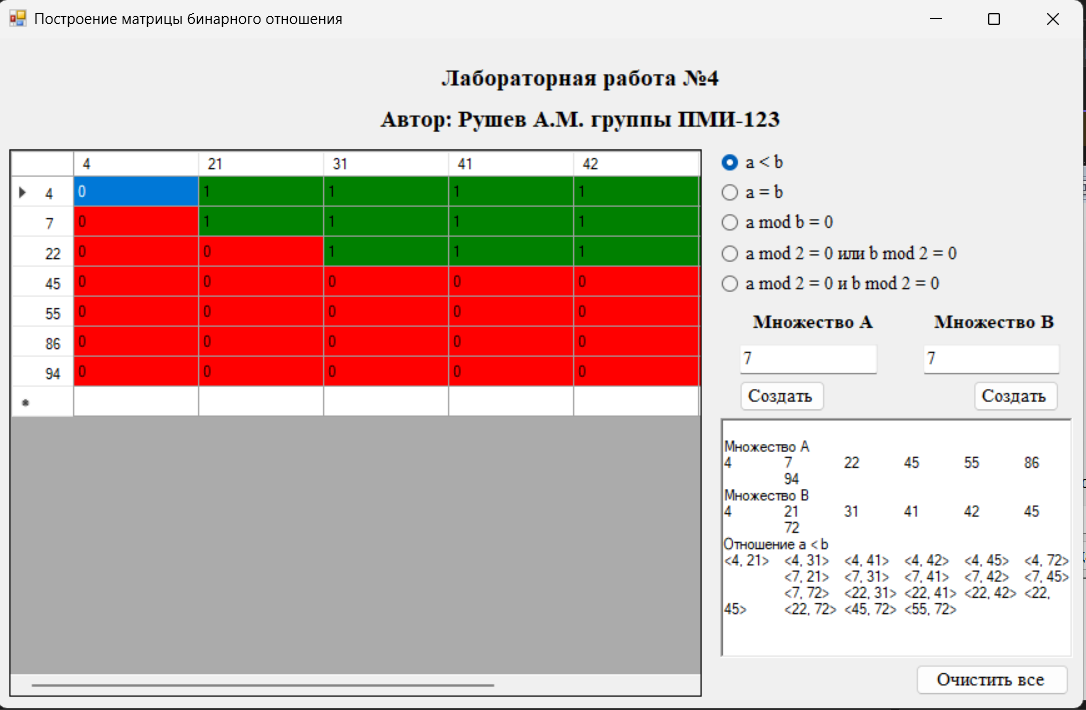


Рисунок 1 – Результат 1 отношения

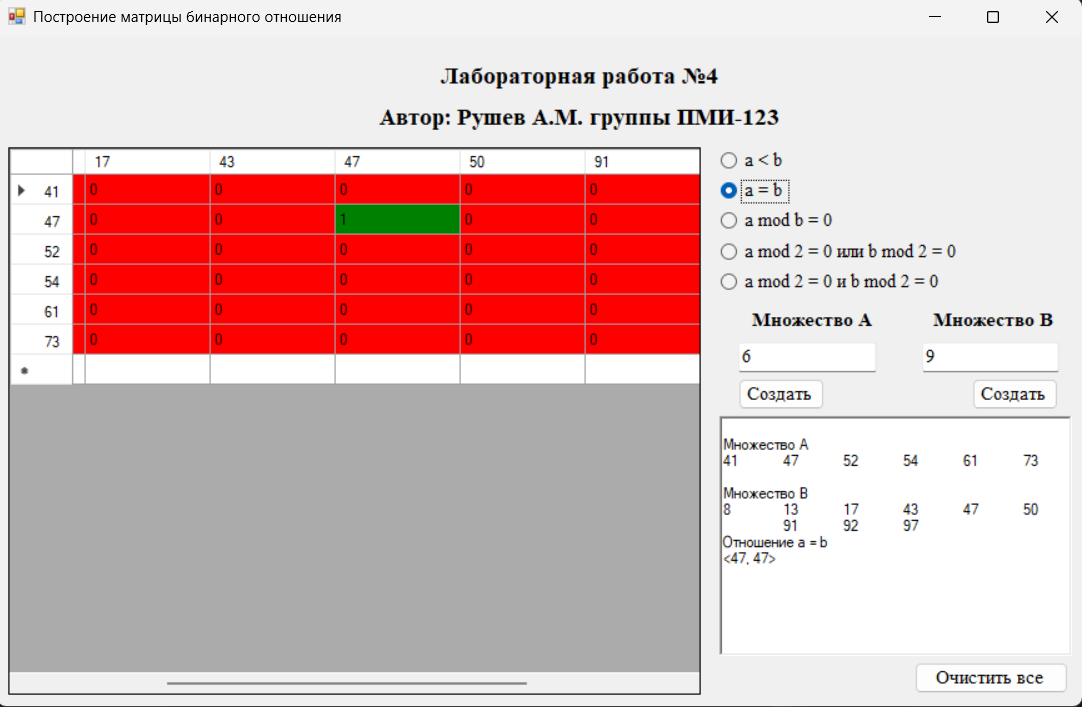


Рисунок 2 – Результат 2 отношения

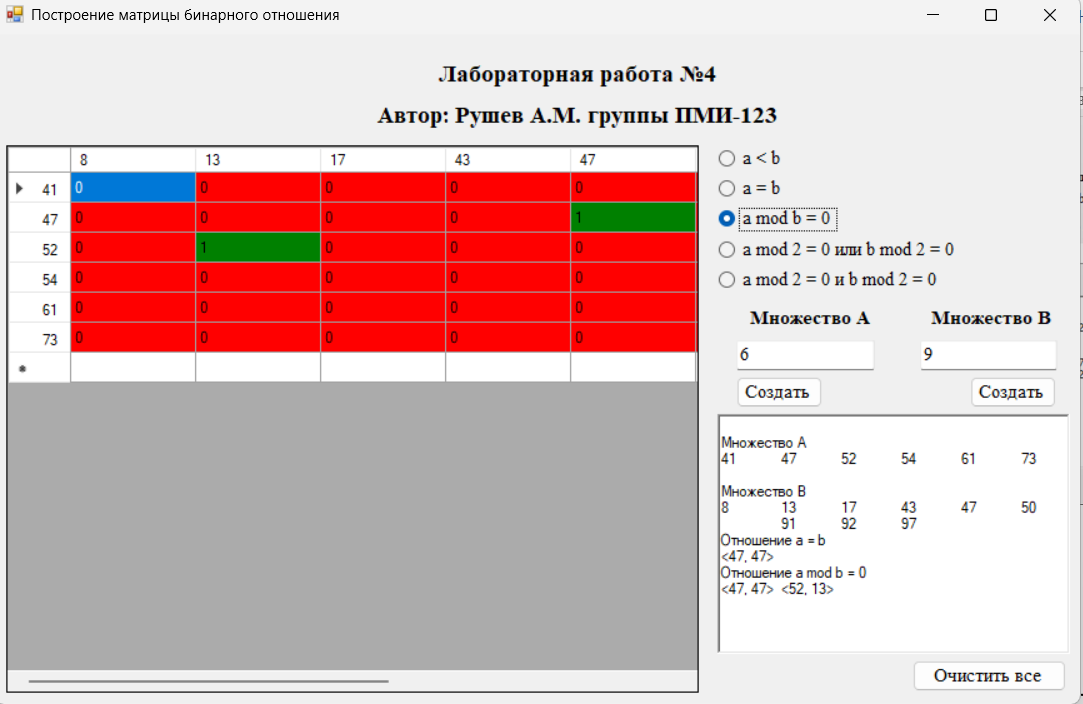


Рисунок 3 – Результат 3 отношения

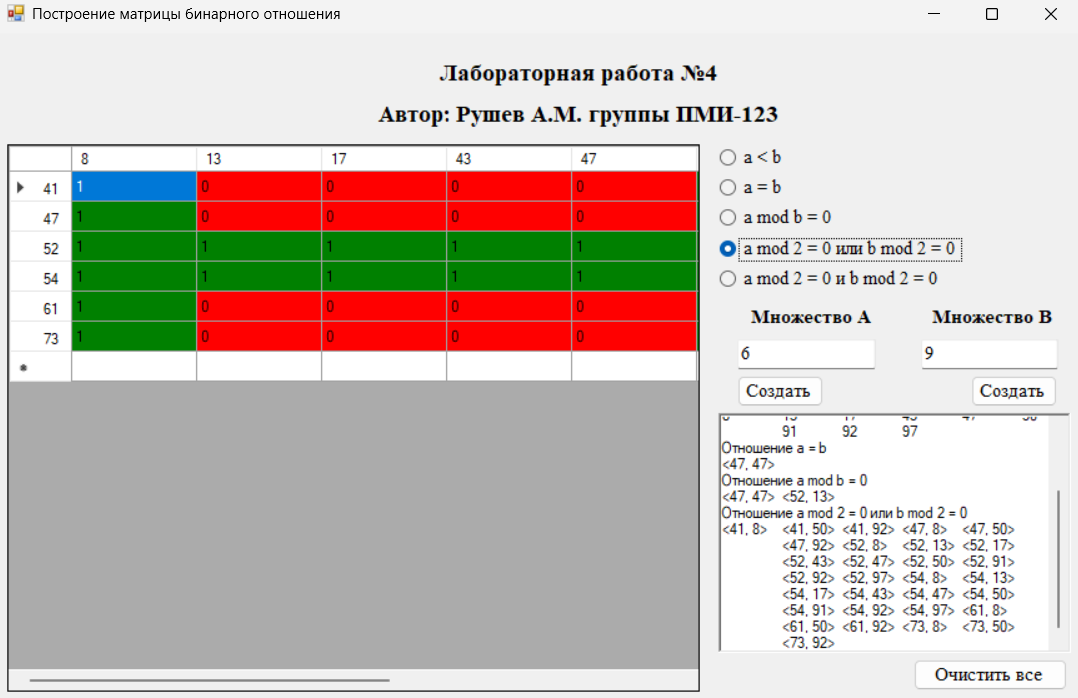


Рисунок 4 – Результат 4 отношения

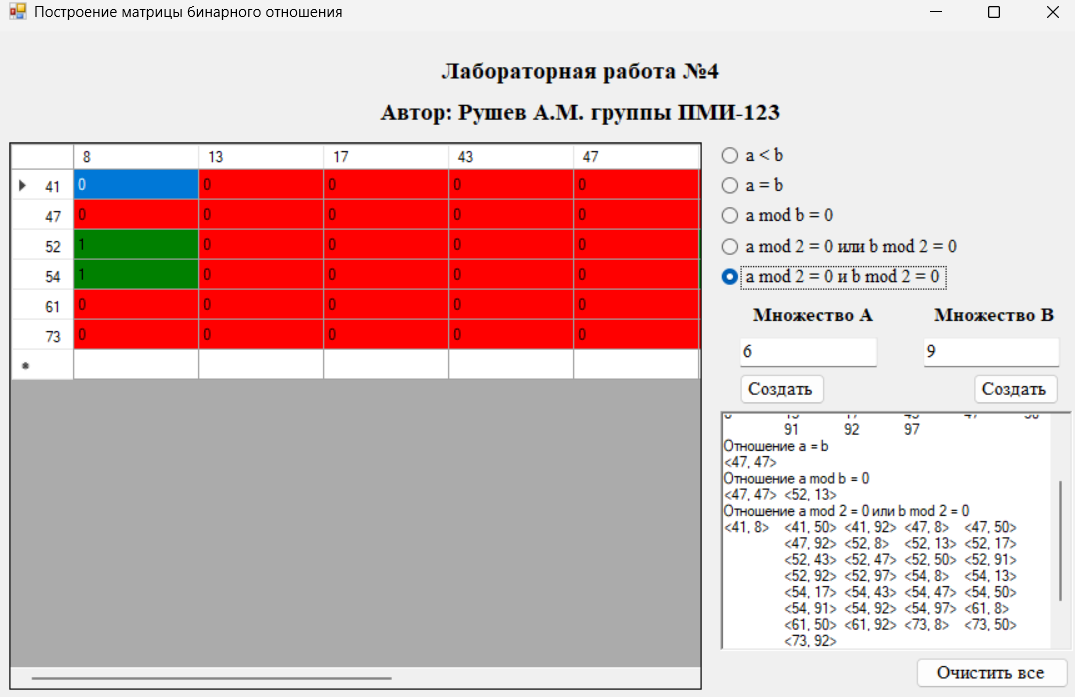


Рисунок 5 – Результат 5 отношения

**Вывод:** данная программа демонстрирует простой способ моделирования и визуализации бинарного отношения между двумя конечными множествами. Используя этот подход, можно легко изменять множества и отношения для получения различных матриц. Это может быть полезно в задачах теории графов, комбинаторики и других областях, где необходимо анализировать связи между элементами.

Защита отчета

using System;

using System.Collections.Generic;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Windows.Forms;

namespace Lab4\_CM\_WinForms

{

public partial class MainForm : Form

{

#region Fields

private Random rnd = new Random();

private Int32[] massA = new Int32[0];

private Int32[] massB = new Int32[0];

private Int32[][] massAll = new Int32[0][];

private Int32 numberSelected = 0;

private DataTable dataTable = new DataTable();

#endregion

public MainForm()

{

InitializeComponent();

}

private void MainForm\_Load(object sender, EventArgs e)

{

tbxA.Text = "Введите число";

tbxA.ForeColor = Color.Gray;

tbxB.Text = "Введите число";

tbxB.ForeColor = Color.Gray;

btnCreateA.Enabled = false;

btnCreateB.Enabled = false;

}

#region ButoonsActions

private void btnCreateA\_Click(object sender, EventArgs e)

{

CreateMass(tbxA.Text, tbxA.Text.Count(), massA, "A");

}

private void btnCreateB\_Click(object sender, EventArgs e)

{

CreateMass(tbxB.Text, tbxB.Text.Count(), massB, "B");

}

private void btnClearAll\_Click(object sender, EventArgs e)

{

tbxA.Clear();

tbxB.Clear();

rTbxPrintInfo.Clear();

dgvPrintInfo.Columns.Clear();

tbxA\_Leave(sender, e);

tbxB\_Leave(sender, e);

rBtnFuncFirst.Checked = false;

rBtnFuncSecond.Checked = false;

rBtnFuncThird.Checked = false;

rBtnFuncFourth.Checked = false;

rBtnFuncFivth.Checked = false;

}

#endregion

#region Methods

//создание массивов

private void CreateMass(string tbx, int countTbx, int[] mass, string nameRTbxPrintInfo)

{

try

{

bool isValidate = true;

if (!Int32.TryParse(tbx, out countTbx))

{

if (tbx == tbxA.Text)

{

tbxA.Clear();

MessageBox.Show($"Число должно быть целым");

isValidate = false;

}

if (tbx == tbxB.Text)

{

tbxB.Clear();

MessageBox.Show($"Число должно быть целым");

isValidate = false;

}

}

if (isValidate)

{

HashSet<Int32> list = new HashSet<Int32>();

while (list.Count < countTbx)

{

list.Add(rnd.Next(1, 100));

}

mass = list.ToArray();

Array.Sort(mass);

switch (nameRTbxPrintInfo)

{

case "A":

massA = mass;

break;

case "B":

massB = mass;

break;

}

rTbxPrintInfo.AppendText($"\nМножество {nameRTbxPrintInfo}\n");

foreach (var item in mass)

{

rTbxPrintInfo.AppendText($"{item.ToString()}\t");

}

}

}

catch (Exception ex)

{

MessageBox.Show($"{ex.Message}");

}

}

//создание строк и столбцов в таблице

private void CreateDataGridInfo()

{

try

{

dataTable.Clear();

dataTable.Columns.Clear();

foreach (var item in massB)

{

dataTable.Columns.Add(item.ToString());

}

dgvPrintInfo.DataSource = dataTable;

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

{

dataTable.Rows.Add();

if (i < dgvPrintInfo.Rows.Count)

{

dgvPrintInfo.Rows[i].HeaderCell.Value = massA[i].ToString();

}

}

}

catch (Exception ex)

{

MessageBox.Show($"{ex.Message}");

}

}

//расчет матрицы

public int[][] InfoDataMatrix(int[] mas1, int[] mas2, int N, int M)

{

List<int[]> res = new List<int[]>();

switch (numberSelected)

{

case 1:

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

{

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

{

if (mas1[i] < mas2[j])

{

res.Add(new int[] { mas1[i], mas2[j] });

}

}

}

break;

case 2:

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

{

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

{

if (mas1[i] == mas2[j])

{

res.Add(new int[] { mas1[i], mas2[j] });

}

}

}

break;

case 3:

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

{

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

{

if (mas1[i] % mas2[j] == 0)

{

res.Add(new int[] { mas1[i], mas2[j] });

}

}

}

break;

case 4:

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

{

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

{

if (mas1[i] % 2 == 0 || mas2[j] % 2 == 0)

{

res.Add(new int[] { mas1[i], mas2[j] });

}

}

}

break;

case 5:

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

{

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

{

if (mas1[i] % 2 == 0 && mas2[j] % 2 == 0)

{

res.Add(new int[] { mas1[i], mas2[j] });

}

}

}

break;

case 6:

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

{

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

{

if (mas1[i] != 0 && mas2[j] != 0)

{

if (GCD(mas1[i], mas2[j]) > 1)

{

res.Add(new int[] { mas1[i], mas2[j] });

}

}

}

}

break;

default: break;

}

return res.ToArray();

}

private int GCD(int a, int b)

{

while (b != 0)

{

int temp = b;

b = a % b;

a = temp;

}

return Math.Abs(a);

}

//занесение в таблицу

private void InsertInfoTable()

{

try

{

if (massA.Length == 0 || massB.Length == 0)

{

MessageBox.Show($"Нужно создать множество A и B");

}

else

{

switch (numberSelected)

{

case 1:

var resultFirstFunc = InfoDataMatrix(massA, massB, massA.Length, massB.Length);

rTbxPrintInfo.AppendText($"\nОтношение {rBtnFuncFirst.Text}\n");

foreach (var item in resultFirstFunc)

{

rTbxPrintInfo.AppendText($"<{string.Join(", ", item)}>\t");

}

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

{

for (int j = 0; j < massB.Length; j++)

{

if (massA[i] < massB[j])

{

dgvPrintInfo.Rows[i].Cells[j].Value = 1;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Green;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

else

{

dgvPrintInfo.Rows[i].Cells[j].Value = 0;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Red;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

}

}

break;

case 2:

var resultSecondFunc = InfoDataMatrix(massA, massB, massA.Length, massB.Length);

rTbxPrintInfo.AppendText($"\nОтношение {rBtnFuncSecond.Text}\n");

foreach (var item in resultSecondFunc)

{

rTbxPrintInfo.AppendText($"<{string.Join(", ", item)}>\t");

}

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

{

for (int j = 0; j < massB.Length; j++)

{

if (massA[i] == massB[j])

{

dgvPrintInfo.Rows[i].Cells[j].Value = 1;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Green;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

else

{

dgvPrintInfo.Rows[i].Cells[j].Value = 0;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Red;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

}

}

break;

case 3:

var resultThirdFunc = InfoDataMatrix(massA, massB, massA.Length, massB.Length);

rTbxPrintInfo.AppendText($"\nОтношение {rBtnFuncThird.Text}\n");

foreach (var item in resultThirdFunc)

{

rTbxPrintInfo.AppendText($"<{string.Join(", ", item)}>\t");

}

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

{

for (int j = 0; j < massB.Length; j++)

{

if (massA[i] % massB[j] == 0)

{

dgvPrintInfo.Rows[i].Cells[j].Value = 1;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Green;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

else

{

dgvPrintInfo.Rows[i].Cells[j].Value = 0;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Red;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

}

}

break;

case 4:

var resultFourthFunc = InfoDataMatrix(massA, massB, massA.Length, massB.Length);

rTbxPrintInfo.AppendText($"\nОтношение {rBtnFuncFourth.Text}\n");

foreach (var item in resultFourthFunc)

{

rTbxPrintInfo.AppendText($"<{string.Join(", ", item)}>\t");

}

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

{

for (int j = 0; j < massB.Length; j++)

{

if (massA[i] % 2 == 0 || massB[j] % 2 == 0)

{

dgvPrintInfo.Rows[i].Cells[j].Value = 1;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Green;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

else

{

dgvPrintInfo.Rows[i].Cells[j].Value = 0;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Red;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

}

}

break;

case 5:

var resultFivethFunc = InfoDataMatrix(massA, massB, massA.Length, massB.Length);

rTbxPrintInfo.AppendText($"\nОтношение {rBtnFuncFivth.Text}\n");

foreach (var item in resultFivethFunc)

{

rTbxPrintInfo.AppendText($"<{string.Join(", ", item)}>\t");

}

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

{

for (int j = 0; j < massB.Length; j++)

{

if (massA[i] % 2 == 0 && massB[j] % 2 == 0)

{

dgvPrintInfo.Rows[i].Cells[j].Value = 1;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Green;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

else

{

dgvPrintInfo.Rows[i].Cells[j].Value = 0;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Red;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

}

}

break;

case 6:

var resultSixFunc = InfoDataMatrix(massA, massB, massA.Length, massB.Length);

rTbxPrintInfo.AppendText($"\nОтношение {rBtnFuncSix.Text}\n");

var gcdResults = new List<int>();

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

{

for (int j = 0; j < massB.Length; j++)

{

var gcd = GCD(massA[i], massB[j]);

gcdResults.Add(gcd);

}

}

foreach (var item in resultSixFunc)

{

rTbxPrintInfo.AppendText($"<{string.Join(", ", item)}>\t");

}

rTbxPrintInfo.AppendText("\n");

foreach (var itemGCD in gcdResults)

{

if(itemGCD != 1)

{

rTbxPrintInfo.AppendText($"НОД: {itemGCD}\t");

}

}

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

{

for (int j = 0; j < massB.Length; j++)

{

if (GCD(massA[i], massB[j]) > 1)

{

dgvPrintInfo.Rows[i].Cells[j].Value = 1;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Green;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

else

{

dgvPrintInfo.Rows[i].Cells[j].Value = 0;

dgvPrintInfo.Rows[i].Cells[j].Style.BackColor = Color.Red;

dgvPrintInfo.Columns[j].Width = 50;

dgvPrintInfo.Rows[i].Height = 20;

}

}

}

break;

default: break;

}

}

}

catch (Exception ex)

{

MessageBox.Show($"{ex.Message}");

}

}

#endregion

#region TextBox A handlers

private void tbxA\_Enter(object sender, EventArgs e)

{

tbxA.Text = null;

tbxA.ForeColor = Color.Black;

}

private void tbxA\_Leave(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(tbxA.Text))

{

tbxA.Text = "Введите число";

tbxA.ForeColor = Color.Gray;

btnCreateA.Enabled = false;

}

}

private void tbxA\_TextChanged(object sender, EventArgs e)

{

if (!string.IsNullOrEmpty(tbxA.Text))

{

btnCreateA.Enabled = true;

}

else

{

btnCreateA.Enabled = false;

}

}

#endregion

#region TextBox B handlers

private void tbxB\_TextChanged(object sender, EventArgs e)

{

if (!string.IsNullOrEmpty(tbxB.Text))

{

btnCreateB.Enabled = true;

}

else

{

btnCreateB.Enabled = false;

}

}

private void tbxB\_Enter(object sender, EventArgs e)

{

tbxB.Text = null;

tbxB.ForeColor = Color.Black;

}

private void tbxB\_Leave(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(tbxB.Text))

{

tbxB.Text = "Введите число";

tbxB.ForeColor = Color.Gray;

btnCreateB.Enabled = false;

}

}

#endregion

#region RadioButtons handlers

private void rBtnFuncFirst\_CheckedChanged(object sender, EventArgs e)

{

if (rBtnFuncFirst.Checked)

{

numberSelected = 1;

CreateDataGridInfo();

InsertInfoTable();

}

}

private void rBtnFuncSecond\_CheckedChanged(object sender, EventArgs e)

{

if (rBtnFuncSecond.Checked)

{

numberSelected = 2;

CreateDataGridInfo();

InsertInfoTable();

}

}

private void rBtnFuncThird\_CheckedChanged(object sender, EventArgs e)

{

if (rBtnFuncThird.Checked)

{

numberSelected = 3;

CreateDataGridInfo();

InsertInfoTable();

}

}

private void rBtnFuncFourth\_CheckedChanged(object sender, EventArgs e)

{

if (rBtnFuncFourth.Checked)

{

numberSelected = 4;

CreateDataGridInfo();

InsertInfoTable();

}

}

private void rBtnFuncFivth\_CheckedChanged(object sender, EventArgs e)

{

if (rBtnFuncFivth.Checked)

{

numberSelected = 5;

CreateDataGridInfo();

InsertInfoTable();

}

}

private void rBtnFuncSix\_CheckedChanged(object sender, EventArgs e)

{

if (rBtnFuncSix.Checked)

{

numberSelected = 6;

CreateDataGridInfo();

InsertInfoTable();

}

}

#endregion

}

}

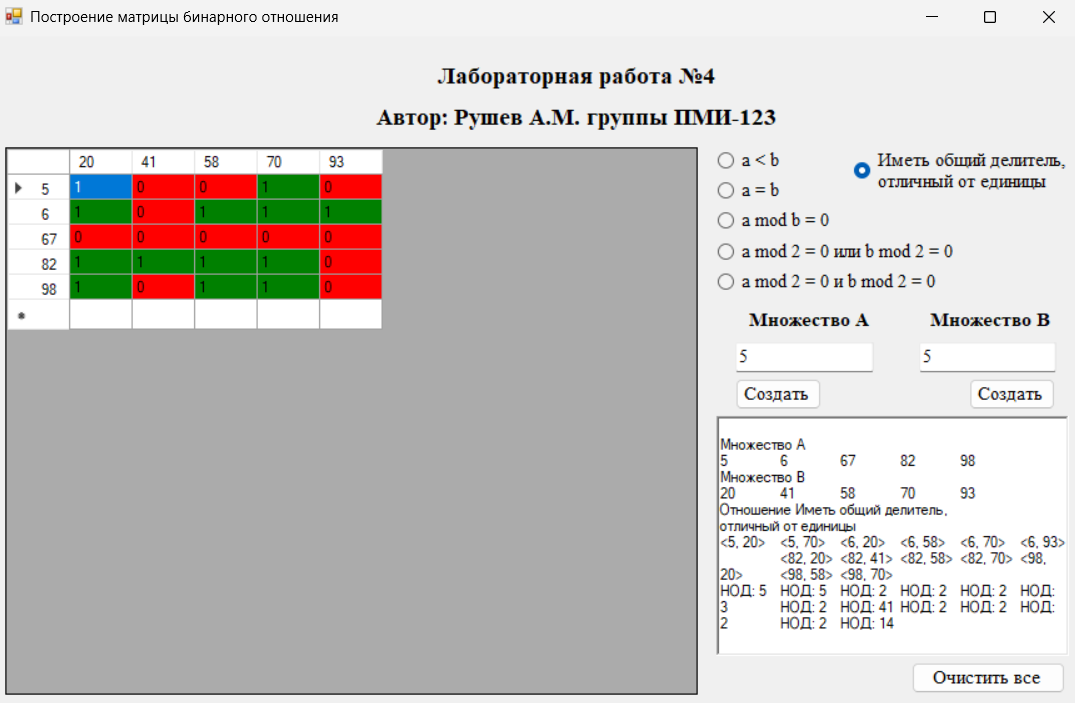


Рисунок 6 – Результат нового отношения