БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Лабораторная работа № 12

по дисциплине «Структуры и алгоритмы обработки данных»

Вариант № 4

Выполнил студент: Ефименко Павел Викторович,

группа 981063

Минск 2020

**Условие:**

*Задание.* Представить ориентированный граф, состоящий из 7-10 вершин, с помощью матрицы смежности, а затем выполнить следующие операторы над его элементами.

1. Указать вершину *v* и определить список вершин, смежных с вершиной *v*. Если *v* не имеет смежных вершин, то возвращается «нулевая» вершина

2. Указать вершину *v* и определить список вершин, из которых можно попасть в вершину  *v*. Если таких вершин на орграфе нет, то возвращается

3. Указать вершину *v* и индекс *i* интересующей вершины из списка вершин, смежных с *v*. Определить вершину с индексом *i*. Если таких вершин на орграфе нет, то возвращается.

**Листинг:**

using System;

using System.Collections.Generic;

using System.Linq;

namespace Lr12

{

class Program

{

public static int[,] matrix;

public List<int> pathes = new List<int>();

// Рекурсивный метод для поиска путей.

public static void FindWays(int elem, int n)

{

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

{

if (matrix[elem - 1, j] == 1)

{

Console.Write($"Переход от {elem} к {j + 1}\n");

if (j != n - 1)

{

FindWays(j + 1, n);

}

}

}

}

static void Main(string[] args)

{

Console.WriteLine("Введите кол-во ветвей: ");

var n = int.Parse(Console.ReadLine());

matrix = new int[n, n];

// Cоздаем матрицу смежности.

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

{

Console.WriteLine($"Элемент {i + 1} имеет связь с: ");

var elements = Console.ReadLine();

var links = elements.Split(' ').Select(x => int.Parse(x)).ToArray();

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

{

if (links.Contains(j + 1))

{

matrix[i, j] = 1;

}

}

}

// Вывод матриы смежности.

for (int i = 0, j = 0; i < n && j < n; j++)

{

Console.Write(matrix[i, j] + "\t");

if (j == n - 1)

{

i++;

j = -1;

Console.WriteLine();

}

}

// Ввод элемента для поиска.

Console.WriteLine("Элемент для поиска:");

var elementForSearch = int.Parse(Console.ReadLine());

List<int> relatedElements = new List<int>();

// Поиск смежных элементов.

Console.WriteLine("Элемент смежен с:");

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

{

if (matrix[elementForSearch - 1, j] == 1)

{

Console.Write($"Элемент {j + 1} смежен с {elementForSearch}\n");

}

}

// Поиск пути.

Console.WriteLine($"Путь из {elementForSearch}:");

FindWays(elementForSearch, n);

Console.Read();

}

}

}

**Выполнение:**

