Міністерство освіти та науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

ЗВІТ

до лабораторної роботи №1

з дисципліни «Основи Web-програмування»

Виконав: студент 2 курсу групи ІП-64 , ФІОТ

Євтушенко Ілля

Номер залікової: 6406 Номер в списку: 7

Київ – 2017

**Код програми:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static int size;

static int[,] matrix;

static int[] enters;

static int[] exits;

static int[,] paths1;

static int[,] paths2;

static void Main(string[] args)

{

getMatrix();

calc1Answer();

calc2Answer();

Console.ReadLine();

}

static void getMatrix() {

getSize();

matrix = new int[size, size];

if (isAutoMatrix())

{

AutoMatrix();

}

else

{

UserMatrix();

}

CheckInputs();

}

static void AutoMatrix() {

Random random = new Random();

enters = new int[size/5 + 1];

exits = new int[size/5 + 1];

for (int i = 1; i < size; i++)

{

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

{

matrix[i, j] = matrix[j, i] = (random.NextDouble() < 0.65) ? 1 : 0;

}

}

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

{

matrix[i, i] = 0;

}

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

{

enters[i] = random.Next(size);

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

{

if(enters[i] == enters[j])

{

i--;

break;

}

}

}

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

{

exits[i] = random.Next(size);

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

{

if (exits[i] == exits [j])

{

i--;

break;

}

}

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

{

if (exits[i] == enters[k])

{

i--;

break;

}

}

}

}

static void UserMatrix() {

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

{

string[] listLine = Console.ReadLine().Split(' ');

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

{

matrix[i, j] = Convert.ToInt32(listLine.ElementAt(j));

}

}

Console.WriteLine("Введите все входы: ");

string[] entersString = Console.ReadLine().Split(' ');

enters = new int[entersString.Length];

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

{

enters[i] = Convert.ToInt32(entersString[i]);

}

Console.WriteLine("Введите все выходы: ");

string[] exitsString = Console.ReadLine().Split(' ');

exits = new int[exitsString.Length];

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

{

exits[i] = Convert.ToInt32(exitsString[i]);

}

}

static void CheckInputs()

{

Console.WriteLine("Матрица: ");

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

{

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

{

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

}

Console.WriteLine();

}

Console.WriteLine("Входы: ");

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

{

Console.Write(enters[i] + " ");

}

Console.WriteLine("\nВыходы: ");

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

{

Console.Write(exits[i] + " ");

}

Console.WriteLine();

}

static void getSize() {

Console.WriteLine("Введите размерность матрицы: ");

size = Convert.ToInt32(Console.ReadLine());

}

static bool isAutoMatrix()

{

Console.WriteLine("Сгенерировать матрицу автоматически?[y/n]");

char answer = Console.ReadLine().ElementAt(0);

if (answer.Equals('y')) {

return true;

}

else

{

return false;

}

}

static void calc1Answer() {

int answers = 0;

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

Queue<int> currNodes = new Queue<int>();

List<int> entersList = enters.ToList();

while(entersList.Count > 0)

{

int currEnter = entersList.ElementAt(0);

entersList.Remove(currEnter);

currNodes.Enqueue(currEnter);

while(currNodes.Count > 0)

{

int currNode = currNodes.Dequeue();

usedNodes.Add(currNode);

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

{

if (matrix[currNode, i] == 1)

{

if (exits.Contains(i))

{

//Console.WriteLine("exit: " + i);

answers++;

if (enters.Contains(currNode))

{

continue;

}

else

{

break;

}

}

if (usedNodes.Contains(i))

{

//Console.WriteLine("used node: " + i);

continue;

}

if (entersList.Contains(i))

{

//Console.WriteLine("enter: " + i);

//currNodes.Enqueue(i);

entersList.Remove(i);

}

//Console.WriteLine("add node: " + i);

currNodes.Enqueue(i);

//usedNodes.Add(i);

if (enters.Contains(currNode))

{

continue;

}

else

{

break;

}

}

}

}

}

Console.WriteLine("Ответ1: " + answers);

}

static void calc2Answer() {

int answers = 0;

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

Queue<int> currNodes = new Queue<int>();

List<int> entersList = enters.ToList();

while (entersList.Count > 0)

{

int currEnter = entersList.ElementAt(0);

entersList.Remove(currEnter);

currNodes.Enqueue(currEnter);

while (currNodes.Count > 0)

{

int currNode = currNodes.Dequeue();

usedNodes.Add(currNode);

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

{

if (matrix[currNode, i] == 1)

{

if (exits.Contains(i))

{

//Console.WriteLine("exit: " + i);

answers++;

if (enters.Contains(currNode))

{

continue;

}

else

{

break;

}

}

if (usedNodes.Contains(i))

{

//Console.WriteLine("used node: " + i);

continue;

}

if (entersList.Contains(i))

{

//Console.WriteLine("enter: " + i);

//currNodes.Enqueue(i);

entersList.Remove(i);

}

//Console.WriteLine("add node: " + i);

currNodes.Enqueue(i);

usedNodes.Add(i);

if (enters.Contains(currNode))

{

continue;

}

else

{

break;

}

}

}

}

}

Console.WriteLine("Ответ2: " + answers);

}

}

}

**Скріншот програми:**

