Кафедра обчислювальної техніки (ОТ)

Лабораторна робота №1

З предмету «Основи Web-програмування»

*Виконав*

*Студент другого курсу*

*групи ІП-62*

*Гук Микола*

*Номер залікової книжки - 6203*

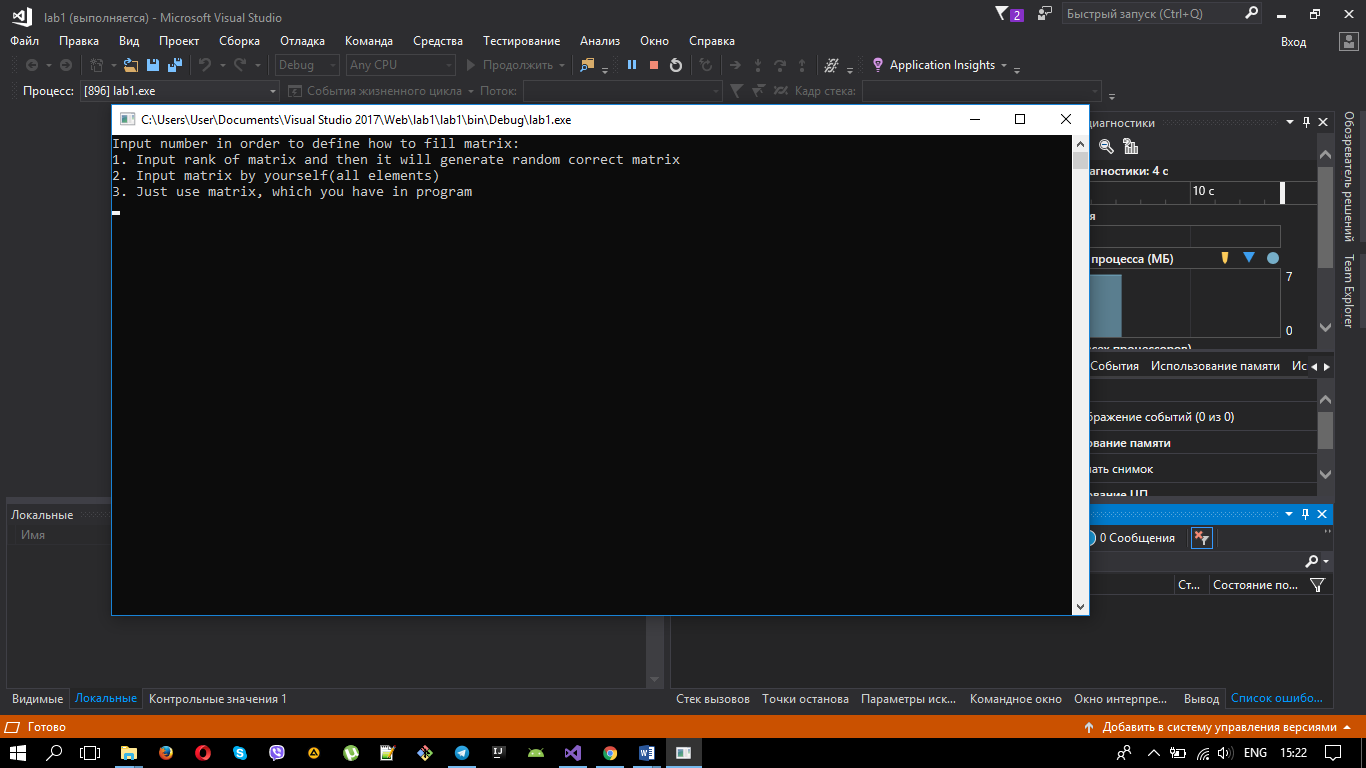
**Постановка задачі до комп’ютерного практикуму № 4**

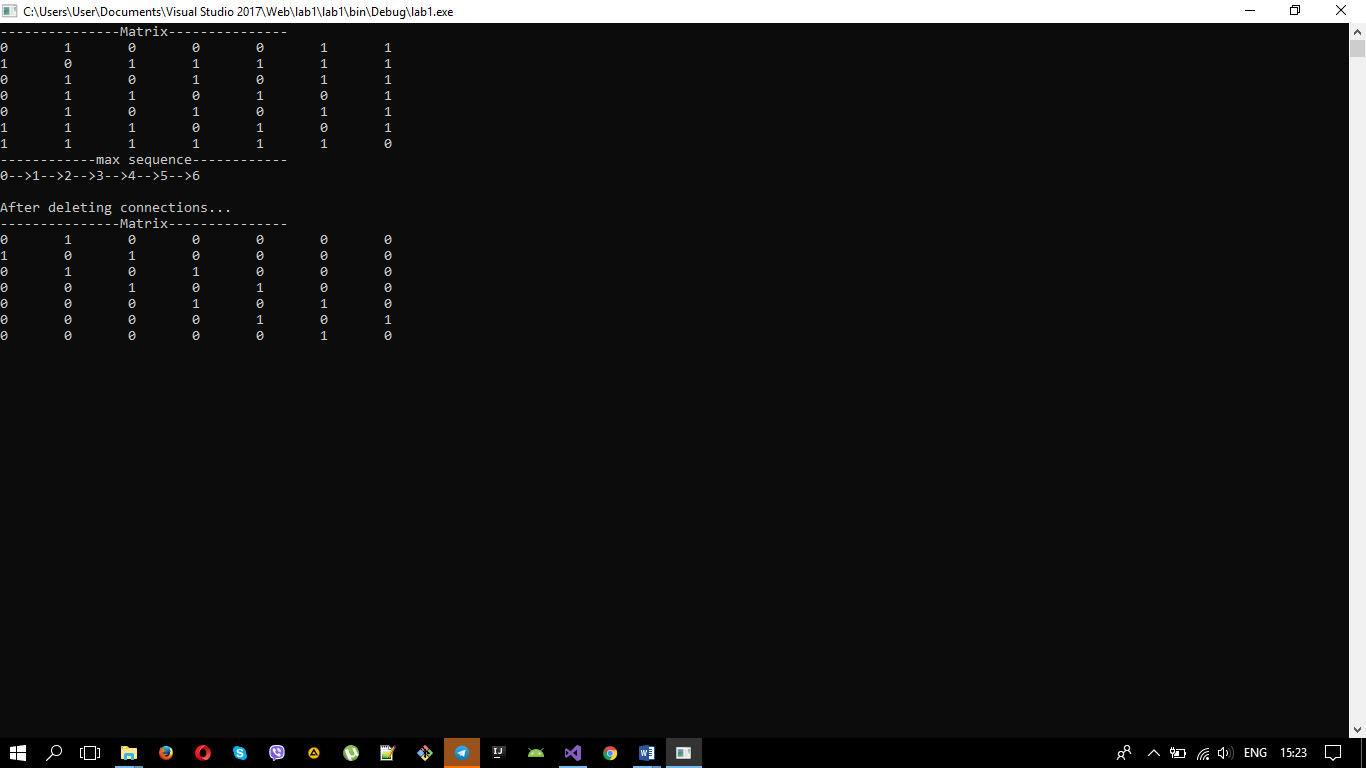
При виконанні комп’ютерого практикуму слід реалізувати наступні задачі:

1. дозволяти користувачу визначати кількість вершин графа самостійно з консолі;
2. дозволяти користувачу вводити довільні матриці (списки суміжності) різної розмірності самостійно з консолі;
3. мати можливість генерації довільної матриці (списку суміжності) з консолі;
4. виводи на екран результат

**Варіант 27**

N колец сцеплены между собой (задана матрица A(n\*n), A(i,j)=1 в случае, если кольца i и j сцеплены друг с другом и A(i,j)=0 иначе). Удалить минимальное количество колец так, чтобы получилась цепочка.





**Код програми**

using System;

using System.Collections;

namespace lab1

{

class Program

{

static int[,] matrix1 = new int [,]{

{ 0, 1, 0, 0, 0, 1, 1},

{ 1, 0, 1, 1, 1, 1, 1},

{ 0, 1, 0, 1, 0, 1, 1},

{ 0, 1, 1, 0, 1, 0, 1},

{ 0, 1, 0, 1, 0, 1, 1},

{ 1, 1, 1, 0, 1, 0, 1},

{ 1, 1, 1, 1, 1, 1, 0}

};

static int[,] matrix2 = new int[,]{

{ 0, 1, 0, 0, 0, 1, 0},

{ 0, 0, 0, 0, 0, 0, 1},

{ 0, 0, 0, 1, 0, 1, 0},

{ 0, 0, 1, 0, 1, 0, 0},

{ 0, 0, 0, 1, 0, 1, 0},

{ 0, 0, 1, 0, 1, 0, 0},

{ 0, 1, 0, 0, 0, 0, 0}

};

static int[,] matrix3 = new int[,]{

{ 0, 1, 0, 0, 0},

{ 1, 0, 0, 0, 0},

{ 0, 0, 0, 0, 0},

{ 0, 0, 0, 0, 0},

{ 0, 0, 0, 0, 0}

};

static void Main(string[] args)

{

ArrayList arraylist = new ArrayList();

ArrayList maxSequence = new ArrayList();

int [,] myMatrix = info();

Console.Clear();

makeCorrectMatrix(myMatrix);

output(myMatrix);

int[,] c = myMatrix;

for (int i = 0; i < myMatrix.GetLength(0); i++)

{

arraylist.Add(i);

find(c, i, arraylist, maxSequence);

}

Console.WriteLine("------------max sequence------------");

for(int i=0; i<maxSequence.Count;i++)

{

if(i==maxSequence.Count-1)

Console.WriteLine(maxSequence[i]);

else Console.Write(maxSequence[i]+ "-->");

}

myMatrix = deleteConnection(myMatrix, maxSequence);

/\*\*

\* If you want you can just delete whole useless ring

\*/

/\*\*

for (int i= myMatrix.GetLength(0)-1; i >= 0 ; i--)

{

if (!maxSequence.Contains(i))

{

myMatrix = deleteRing(myMatrix, i);

output(myMatrix);

}

} \*/

Console.ReadLine();

}

static int[,] info()

{

Console.Clear();

int key;

Console.WriteLine("Input number in order to define how to fill matrix:");

Console.WriteLine("1. Input rank of matrix and then it will generate random correct matrix");

Console.WriteLine("2. Input matrix by yourself(all elements)");

Console.WriteLine("3. Just use matrix, which you have in program");

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

switch (key)

{

case 1: return input1();

case 2: return input2();

case 3: return matrix1;

default: return info();

}

}

static int[,] input1()

{

Console.Clear();

Random rnd = new Random();

Console.Write("Input rank of matrix: ");

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

int[,] matrix = new int[rank, rank];

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

{

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

matrix[i, j] = rnd.Next(2);

}

return matrix;

}

static int[,] input2()

{

Console.Clear();

Console.WriteLine("WARNING! You need understand several rules: ");

Console.WriteLine("1) If i-ring has connection with j-ring then j-ring has connection" +

" with i-ring, so if matrix[i,j]==1 then matrix[j,i]=1");

Console.WriteLine("2) i-ring can't has connection with i-ring, so matrix[i,i] must be equal 0");

Console.WriteLine("3) all indexes start from 0");

Console.Write("Please input rank of matrix: ");

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

int[,] matrix = new int[rank, rank];

Console.WriteLine("Please input element by element:");

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

{

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

{

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

matrix[i, j] = Convert.ToInt32(Console.ReadLine());

}

Console.WriteLine();

}

return matrix;

}

static void output(int [,] matrix)

{

Console.WriteLine("---------------Matrix---------------");

for(int i=0; i < matrix.GetLength(0); i++)

{

for(int j=0; j < matrix.GetLength(0); j++)

{

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

}

Console.WriteLine();

}

}

static void find(int[,] c, int l, ArrayList arraylist, ArrayList maxSequence)

{

for (int j = 0; j < c.GetLength(0); j++)

if (c[l, j] == 1)

{

if (!arraylist.Contains(j))

{

arraylist.Add(j);

find(c, j, arraylist, maxSequence);

}

}

if (arraylist.Count > maxSequence.Count)

{

maxSequence.Clear();

for (int i = 0; i < arraylist.Count; i++)

{

maxSequence.Add(arraylist[i]);

}

}

if (arraylist.Count > 0)

arraylist.RemoveAt(arraylist.Count - 1);

}

static void makeCorrectMatrix(int [,] matrix)

{

/\*\*

If we have connection i,j then we have connection j,i

Ring can't be connected with itself

\*/

for (int b = 0; b < matrix.GetLength(0); b++)

{

for (int v = 0; v < matrix.GetLength(0); v++)

{

if (matrix[b, v] == 1)

matrix[v, b] = 1;

if (b == v) matrix[b, v] = 0;

}

}

}

static int [,] deleteRing(int[,] matrix, int i)

{

if (i != matrix.GetLength(0)-2)

{

for (int k = i; k < matrix.GetLength(0)-1; k++)

{

for (int m = 0; m < matrix.GetLength(0); m++)

{

matrix[k, m] = matrix[k + 1, m];

matrix[m, k] = matrix[m, k + 1];

}

}

}

int[,] newMatrix = new int[matrix.GetLength(0) - 1, matrix.GetLength(0) - 1];

for(int l=0; l < newMatrix.GetLength(0); l++)

{

for(int j=0; j < newMatrix.GetLength(0); j++)

{

newMatrix[l, j] = matrix[l, j];

}

}

Console.WriteLine("\n---------------------\n"+i+"-row and column were deleted");

return newMatrix;

}

static int [,] deleteConnection(int [,] matrix, ArrayList maxSequence)

{

for(int i = 0; i < maxSequence.Count-1; i++)

{

int n = (int)maxSequence[i];

int m = (int)maxSequence[i + 1];

matrix[n, m] = -1;

matrix[m, n] = -1;

}

for (int i = 0; i < matrix.GetLength(0); i++)

for (int j = 0; j < matrix.GetLength(0); j++)

if (matrix[i, j] != -1)

matrix[i, j] = 0;

for (int i = 0; i < matrix.GetLength(0); i++)

for (int j = 0; j < matrix.GetLength(0); j++)

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

matrix[i, j] = 1;

Console.WriteLine("\nAfter deleting connections...");

output(matrix);

return matrix;

}

}

}