Державний вищий навчальний заклад

«Прикарпатський національний університет імені Василя Стефаника» Кафедра комп’ютерних наук та інформаційних систем

**ЛАБОРАТОРНА РОБОТА №1**

з предмету «Алгоритми і структури даних»

Виконав:

студент групи КН-2

Гриньків В.І.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_2020р.

к.т.н., доц. Никорак Я.Я.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_2020р.

Івано-Франківськ – 2020

Код виконання:

using System;

using System.Collections.Generic;

using System.Windows.Forms;

namespace zavd1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

//функції\\

public int firstquestion(char[] arr)

{

richTextBox1.Text += "\n" + arr[0] + arr[1] + arr[2] + arr[3] + arr[4];

if (arr[0] == '1' && arr[1] == '1' && arr[2] == '1' && arr[3] == '1' && arr[4] == '1')

{

return 0;

}

for (int i = arr.Length - 1; i >= 0; i--)

{

if (arr[i] == '0')

{

arr[i] = '1';

for (int l = i + 1; l <= arr.Length; l++)

{

if (l < arr.Length)

arr[l] = '0';

//else { }

}

return firstquestion(arr);

}

}

return firstquestion(arr);

}

public int secondquestion(char[] arr)

{

richTextBox2.Text += "\n" + arr[0] + arr[1] + arr[2] + arr[3] + arr[4];

if (arr[0] == '4' && arr[1] == '4' && arr[2] == '4' && arr[3] == '4' && arr[4] == '4')

{

return 0;

}

for (int i = arr.Length - 1; i >= 0; i--)

{

if (arr[i] != '4')

{

arr[i] = Convert.ToChar(Convert.ToInt32(arr[i]) + 1);

return secondquestion(arr);

}

else

{

for (int l = i; l >= 0; l--)

{

if (arr[l] != '4')

{

arr[l] = Convert.ToChar(Convert.ToInt32(arr[l]) + 1);

for (int h = l + 1; h < arr.Length; h++)

{

arr[h] = '0';

}

return secondquestion(arr);

}

}

}

}

return secondquestion(arr);

}

public int labirynt(int[,] A, int y, int x, int n, int ex, int ey)

{

//n = 0;

int k = 0;

if (y == ey && x == ex)

{

label1.Text = "Лабіринт пройдено за " + Convert.ToString(n) + " кроків.";

return 0;

}

//A[y, x]--;

if (k == 0)

{

try

{

if (A[y, x + 1] > A[y, x])//вправо

{

A[y, x]--;

x++;

A[y, x]--;

n++;

return labirynt(A, y, x, n, ex, ey);

}

}

catch { }

try

{

if (A[y - 1, x] > A[y, x])//vverh

{

A[y, x]--;

y--;

A[y, x]--;

n++;

return labirynt(A, y, x, n, ex, ey);

}

}

catch { }

try

{

if (A[y, x - 1] > A[y, x])//vlivo

{

A[y, x]--;

x--;

A[y, x]--;

n++;

return labirynt(A, y, x, n, ex, ey);

}

}

catch { }

try

{

if (A[y + 1, x] > A[y, x])//vnyz

{

A[y, x]--;

y++;

A[y, x]--;

n++;

return labirynt(A, y, x, n, ex, ey);

}

}

catch { }

}

k++;

if(k==1)

{

try

{

if(A[y-1,x]<A[y,x] && A[y - 1, x]!=0)//verh

{

A[y, x] = 0;

y--;

A[y, x]++;

n++;

return labirynt(A, y, x, n, ex, ey);

}

}

catch { }

try

{

if (A[y , x-1] < A[y, x] && A[y , x-1] != 0)//vlivo

{

A[y, x] = 0;

x--;

A[y, x]++;

n++;

return labirynt(A, y, x, n, ex, ey);

}

}

catch { }

try

{

if (A[y , x+1] < A[y, x] && A[y , x+1] != 0)//vpravo

{

A[y, x] = 0;

x++;

A[y, x]++;

n++;

return labirynt(A, y, x, n, ex, ey);

}

}

catch { }

try

{

if (A[y + 1, x] < A[y, x] && A[y + 1, x] != 0)//verh

{

A[y, x] = 0;

y++;

A[y, x]++;

n++;

return labirynt(A, y, x, n, ex, ey);

}

}

catch { }

}

return labirynt(A, y, x, n, ex, ey);

}

public int labirynt2(int [,] A, int y, int x, int n, int ex, int ey, int ex1, int ey1, int ex2, int ey2, int cout)

{

int k = 0;

if (y == ey && x == ex)

{

label3.Text = "Перший маршрут пройдено за " + Convert.ToString(n) + " кроків.";

cout++;

}

if (y == ey1 && x == ex1)

{

label4.Text = "Другий маршрут пройдено за " + Convert.ToString(n) + " кроків.";

cout++;

}

if (y == ey2 && x == ex2)

{

label5.Text = "Третій маршрут пройдено за " + Convert.ToString(n) + " кроків.";

cout++;

}

if(cout == 3)

{

return 0;

}

//A[y, x]--;

if (k == 0)

{

try

{

if (A[y, x + 1] > A[y, x])//вправо

{

A[y, x]--;

x++;

A[y, x]--;

n++;

return labirynt2(A, y, x, n, ex, ey, ex1, ey1, ex2, ey2,cout);

}

}

catch { }

try

{

if (A[y - 1, x] > A[y, x])//vverh

{

A[y, x]--;

y--;

A[y, x]--;

n++;

return labirynt2(A, y, x, n, ex, ey, ex1, ey1, ex2, ey2,cout);

}

}

catch { }

try

{

if (A[y, x - 1] > A[y, x])//vlivo

{

A[y, x]--;

x--;

A[y, x]--;

n++;

return labirynt2(A, y, x, n, ex, ey, ex1, ey1, ex2, ey2,cout);

}

}

catch { }

try

{

if (A[y + 1, x] > A[y, x])//vnyz

{

A[y, x]--;

y++;

A[y, x]--;

n++;

return labirynt2(A, y, x, n, ex, ey, ex1, ey1, ex2, ey2,cout);

}

}

catch { }

}

k++;

if (k == 1)

{

try

{

if (A[y - 1, x] < A[y, x] && A[y - 1, x] != 0)//verh

{

A[y, x] = 0;

y--;

A[y, x]++;

n--;

return labirynt2(A, y, x, n, ex, ey, ex1, ey1, ex2, ey2, cout);

}

}

catch { }

try

{

if (A[y, x - 1] < A[y, x] && A[y, x - 1] != 0)//vlivo

{

A[y, x] = 0;

x--;

A[y, x]++;

n--;

return labirynt2(A, y, x, n, ex, ey, ex1, ey1, ex2, ey2, cout);

}

}

catch { }

try

{

if (A[y, x + 1] < A[y, x] && A[y, x + 1] != 0)//vpravo

{

A[y, x] = 0;

x++;

A[y, x]++;

n--;

return labirynt2(A, y, x, n, ex, ey, ex1, ey1, ex2, ey2, cout);

}

}

catch { }

try

{

if (A[y + 1, x] < A[y, x] && A[y + 1, x] != 0)//verh

{

A[y, x] = 0;

y++;

A[y, x]++;

n--;

return labirynt2(A, y, x, n, ex, ey, ex1, ey1, ex2, ey2, cout);

}

}

catch { }

}

return labirynt2(A, y, x, n, ex, ey, ex1, ey1, ex2, ey2, cout);

}

/// <summary>

/// трикутник

/// </summary>

static public int count = 0;

public int len = 3;

public void triangle()

{

int[][] arr = new int[4][];

arr[0] = new int[1] { 6 };

arr[1] = new int[2] { 2, 15 };

arr[2] = new int[3] { 0, 4, 2 };

arr[3] = new int[4] { 2, 1, 4, 4 };

label7.Text = "Найбільша сума = " + Convert.ToString(tr(arr));

}

private int tr(int[][] ar, int row = 0, int col = 0, int total = 0)

{

count += 1;

if (row == ar.Length - 1)

{

return total + ar[row][col];

}

return Math.Max(tr(ar, row + 1, col, total + ar[row][col]), tr(ar, row + 1, col + 1, total + ar[row][col]));

}

//

static int n = 4;

static public int kilkist = 0;

static int ft = 0;

static int[,] ar2 = new int[roww, column];

static List<int> Result = new List<int>();

public static bool temp;

const int roww = 10;

const int column = 10;

//

static private int zmija(int[,] ar, int row = 1, int col = 1, int total = 0)

{

// 100 - # поле , 0 - G ігра

// 10 - o камінь , 1 - X тіло

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

{

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

{

ar2[i, j] = ar[i, j];

}

}

total = kilkist;

if (((row > 8 || row < 1) || (col > 8 || col < 1)) || (ar[row, col] != 0 && kilkist > 0))

{

return 0;

}

if (ar2[row + 1, col] != 0 && ar2[row - 1, col] != 0 && ar2[row, col + 1] != 0 && ar2[row, col - 1] != 0)

{

ar2[row, col] = 0;

return total;

}

else

{

ar2[row, col] = 1;

kilkist++;

}

return Math.Max(Math.Max(zmija(ar2, row + 1, col, total), zmija(ar2, row - 1, col, total)),

Math.Max(zmija(ar2, row, col + 1, total), zmija(ar2, row, col - 1, total)));

}

//кнопки\\

private void label1\_Click(object sender, EventArgs e)

{

}

public char[] arr = "00000".ToCharArray();

public char[] arr2 = "00000".ToCharArray();

public char[] arr3 = "00000".ToCharArray();

private void button1\_Click(object sender, EventArgs e)

{

firstquestion(arr);

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

secondquestion(arr2);

}

private void button4\_Click(object sender, EventArgs e)

{

int ex = 1;

int ey = 0 ;

int n = 0;

int[,] Lab = new int[6, 6]

{{0, 10, 0, 0, 0, 0},

{0, 10, 10, 10, 10, 0},

{0, 0, 0, 10, 0, 0},

{0, 0, 10, 10, 10, 0},

{0, 10, 10, 0, 10, 0},

{9, 10, 0, 0, 0, 0} };

labirynt(Lab, 5, 0, n, ex, ey);

label2.Text = "# E # # # #\n# 0 0 0 0 #\n# # # 0 # #\n# # 0 0 0 #\n# 0 0 # 0 #\nS 0 # # # #";

}

private void button3\_Click(object sender, EventArgs e)

{

int cout = 0;

int n = 0;

//

int ex = 5;

int ey = 5;

//

int ex1 = 5;

int ey1 = 2;

//

int ey2 = 0;

int ex2 = 0;

//

int[,] Lab1 = new int[6, 6]

{{ 10, 10, 0, 0, 0, 0},

{0, 10, 10, 10, 10, 10},

{0, 0, 0, 10, 0, 10},

{0, 0, 10, 10, 10, 0},

{0, 10, 10, 0, 10, 10},

{9, 10, 0, 0, 0, 10} };

//public int labirynt2(int[,] A, int y, int x, int n, int ex, int ey, int ex1, int ey1, int ex2, int ey2, int cout)

labirynt2(Lab1, 5, 0, n, ey, ey, ex1, ey1, ex2, ey2, cout);

label6.Text = "3 0 # # # #\n# 0 0 0 0 0\n# # # 0 # 2\n# # 0 0 0 #\n# 0 0 # 0 0\nS 0 # # # 1";

}

private void button5\_Click(object sender, EventArgs e)

{

triangle();

}

private void button6\_Click(object sender, EventArgs e)

{

int[,] arr = new int[roww, column];

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

{

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

{

if (i == 0 || i == roww - 1)

{

arr[i, j] = 100;

}

else if (j == 0 || j == roww - 1)

{

arr[i, j] = 100;

}

}

}

arr[1, 1] = 1;

arr[2, 3] = 10;

arr[3, 3] = 10;

arr[1, 7] = 10;

arr[4, 1] = 10;

arr[8, 3] = 10;

label8.Text = "Максимальна довжина удава: " +Convert.ToString(zmija(arr) + n);

label9.Text = "# # # # # # # # # #\n# X G G G G G o G #\n# G G o G G G G G #\n# G G o G G G G G #\n# G G o G G G G G #\n# o G G G G G G G #\n# G G G G G G G G #\n# G G G G G G G G #\n# G G G G G G G G #\n# G G o G G G G G #\n# # # # # # # # # #";

}

}

}