# Звіт

Зима Варіант 13

1. Для кожного рядка знайти найменший елемент та його індекси.

Лістинг:

static void Task1\_13(int[,] matrix)

{

Console.WriteLine("Для кожного рядка знайти найменший елемент та його індекси.");

int line, min, num = 0;

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

{

line = i;

min = matrix[i, 0];

num = 0;

for (int j = 1; j < matrix.GetLength(1); j++)

{

if (matrix[i, j] <= min)

{

min = matrix[i, j];

num = j;

}

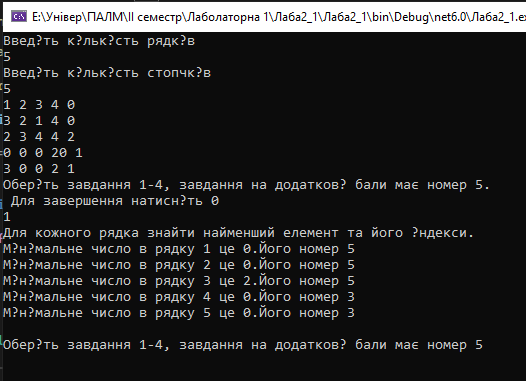
}

Console.WriteLine("Мінімальне число в рядку {0} це {1}.Його номер {2}", line+1, min, num+1);

}

}

Виконання



Програма циклічна і використовує ту ж саму матрицю

2) Якщо хоча б один з максимальних елементів матриці лежить на головній діагоналі,

то перенести його (елемент) на побічну діагональ симетрично відносно вертикалі;

якщо таких елементів кілька, то перенести їх усі.

Лістинг:

static void Task2\_13(int[,] matrix)

{

Console.WriteLine("Якщо хоча б один з максимальних елементів матриці лежить на головній діагоналі,");

Console.WriteLine("то перенести його(елемент) на побічну діагональ симетрично відносно вертикалі");

Console.WriteLine("якщо таких елементів кілька, то перенести їх усі.");

int max = matrix[0, 0];

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

{

for (int j = 0; j < matrix.GetLength(1); j++)

{

if (matrix[i, j] > max)

{

max = matrix[i, j];

}

}

}

int size;

if (matrix.GetLength(0) > matrix.GetLength(1)) size = matrix.GetLength(1);

else size = matrix.GetLength(0);

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

{

if (matrix[i, i] == max)

{

int mind = matrix[i, size - i - 1];

matrix[i, size - i - 1] = matrix[i, i];

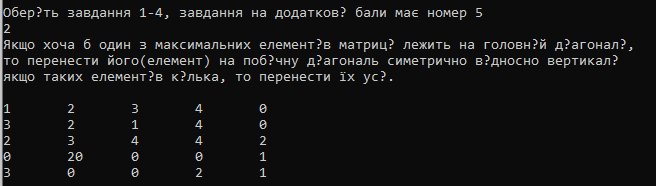
matrix[i, i] = mind;

}

}

PrintMatrix(matrix);

Виконання:



3) Упорядкувати всі рядки з парними номерами за неспаданням, а всі рядки з

непарними номерами за незростанням.

Лістинг:

static void Task3\_13(int[,] matrix)

{

Console.WriteLine("Упорядкувати всі рядки з парними номерами за неспаданням, а всі рядки " +

"\n з непарними номерами за незростанням.");

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

{

if (i % 2 == 0) even(matrix, i);

else odd(matrix, i);;

}

static void even(int[,] matrix, int line)

{

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

{

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

{

if (matrix[line,j + 1] <= matrix[line, j])

{

int mind = matrix[line, j + 1];

matrix[line, j + 1] = matrix[line, j];

matrix[line, j] = mind;

}

}

}

}

static void odd(int[,] matrix, int line)

{

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

{

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

{

if (matrix[line, j + 1] >= matrix[line, j])

{

int mind = matrix[line, j + 1];

matrix[line, j + 1] = matrix[line, j];

matrix[line, j] = mind;

}

}

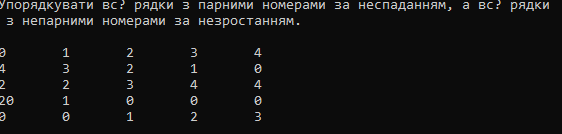
}

}

PrintMatrix(matrix);

}

Виконання:



4) Упорядкувати стовпчики матриці за незростанням кількостей нулів у цих

стовпчиках (тобто, зліва стовпчики, де нулів найбільше, потім ті де трохи менше, і

так аж до правого краю, де ті стовпчики, в яких нулів найменше).

Лістинг:

Console.WriteLine("Упорядкувати стовпчики матриці за незростанням кількостей нулів у цих" +

"\n стовпчиках(тобто, зліва стовпчики, де нулів найбільше, потім ті де трохи менше," +

"\n ітак аж до правого краю, де ті стовпчики, в яких нулів найменше).");

int[] CountOfZero = new int [colms];

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

{

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

{

if (matrix[j,i]==0)

{

CountOfZero[i]++;

}

}

}

for (int i = 0; i < CountOfZero.Length - 1; ++i)

{

for (int j = 0; j < CountOfZero.Length - i - 1; ++j)

{

if (CountOfZero[j + 1] > CountOfZero[j])

{

Swap(matrix, j + 1, j);

}

}

}

PrintMatrix(matrix);

static void Swap(int[,] matrix, int FirstColm, int SecColm)

{

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

{

int mind = matrix[i, FirstColm];

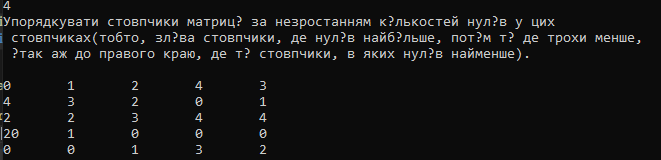
matrix[i, FirstColm] = matrix[i, SecColm];

matrix[i, SecColm] = mind;

}

}

Виконання:



Побічні функції:

Main:

Console.WriteLine("Введіть кількість рядків");

int lines = Int32.Parse(Console.ReadLine());

Console.WriteLine("Введіть кількість стопчків");

int colms = Int32.Parse(Console.ReadLine());

int[,] matrix = new int[lines, colms];

Input(matrix);

Console.WriteLine("Оберіть завдання 1-4, завдання на додаткові бали має номер 5.");

Console.WriteLine(" Для завершення натисніть 0");

int NumOfTask;

do

{

NumOfTask = Int32.Parse(Console.ReadLine());

switch (NumOfTask)

{

case 1:

Task1\_13(matrix);

break;

case 2:

Task2\_13(matrix);

break;

case 3:

Task3\_13(matrix);

break;

case 4:

Task4\_13(matrix, colms);

break;

case 5:

Console.WriteLine("Не працює");

TaskPlus(matrix);

break;

}

Console.WriteLine();

Console.WriteLine("Оберіть завдання 1-4, завдання на додаткові бали має номер 5");

} while (NumOfTask != 0);

Input:

static void Input(int[,] matrix)

{

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

{

string[] a = Console.ReadLine().Trim().Split();

for (int j = 0; j < matrix.GetLength(1); j++)

{

matrix[i, j] = Int32.Parse(a[j]);

}

}

return;

}

PrintMatrix:

static void PrintMatrix(int[,] matrix)

{

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

{

Console.WriteLine();

for (int j = 0; j < matrix.GetLength(1); j++)

{

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

}

}

Console.WriteLine();

}

Додаткове завдання (не працює):

Знайти в кожному рядку перший з максимальних і перший з мінімальних елементів

і поставити їх на першому (технічно 0-му) і останньому місцях рядка. Зверніть

увагу, що випадки «перший є мінімальним» та/або «останній є єдиним

максимальним» можуть створювати певні проблеми.

Лістинг

static void TaskPlus(int[,] matrix)

{

Console.WriteLine("Знайти в кожному рядку перший з максимальних і перший з мінімальних елементів" +

"\nі поставити їх на першому(технічно 0 - му) і останньому місцях рядка.Зверніть" +

"\nувагу, що випадки «перший є мінімальним» та / або «останній є єдиним" +

"\nмаксимальним» можуть створювати певні проблеми.");

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

{

int[] Max = new int[3] { i, 0, matrix[i,0]};

int[] Min = new int[3] { i, matrix.GetLength(1) - 1, matrix[i, matrix.GetLength(1) - 1] };

for (int j = 0; j <= matrix.GetLength(1); j++)

{

if (matrix[i, j] > Max[3])

{

Max[3] = matrix[i, j];

Max[0] = i;

Max[1] = j;

}

else

{

if (matrix[i, j] <= Min[3])

{

Min[3] = matrix[i, j];

Min[0] = i;

Min[1] = j;

}

}

}

SwapMAX(Max, matrix);

SwapMIN(Min, matrix);

PrintMatrix(matrix);

}

static void SwapMAX(int[] MAX, int[,] matrix)

{

int mind = matrix[MAX[0], 0];

matrix[MAX[0], 0] = MAX[3];

matrix[MAX[0], MAX[1]] = mind;

return;

}

static void SwapMIN(int[]MIN, int[,] matrix)

{

int mind = matrix[MIN[0], matrix.GetLength(1)-1];

matrix[MIN[0], matrix.GetLength(1) - 1] = MIN[3];

matrix[MIN[0], MIN[1]] = mind;

return;

}

}

Програма пропускає частину коду. На це вказує пропуск точок зупинки програми.