**Лабораторная работа №5**

«*Прямоугольные массивы*»

**Требования:**

1. Выполнять задания строго в соответствии со своим вариантом в списке журнала.
2. Выполнять все поставленные задачи в задании.
3. В случае, если данные обозначены буквами или не даны в явном виде, то они вводятся с клавиатуры.
4. Обязательно выводить в консоли начальные значения задачи, если они заданы.
5. Обязательно выводить в консоли результат работы программы с соответствующим обозначением того, что этот результат означает.
6. Если пользователь должен что-то ввести с консоли, обязательно вывести на консоль сообщение, что это конкретно должно быть.

*Вариант 1*

1. Задана целочисленная матрица размера nxn. Поменять местами строку, содержащую элемент с наиболь­шим значением в матрице, со строкой, содержащей элемент с наименьшим значением.

static int[,] SwapLines(int[,] matrix, int n)

{

int minIdx = FindMinIdx(matrix, n);

int maxIdx = FindMaxIdx(matrix, n);

int[,] newMatrix = new int[n, n];

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

{

newMatrix[maxIdx, i] = matrix[minIdx, i];

}

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

{

newMatrix[minIdx, i] = matrix[maxIdx, i];

}

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

{

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

{

if (i != minIdx && i != maxIdx)

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

}

}

return newMatrix;

}

static int FindMinIdx(int[,] arr, int n)

{

int idx = 0;

int tmp = arr[0, 0];

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

{

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

{

if (arr[i, j] < tmp)

{

idx = i;

tmp = arr[i, j];

}

}

}

return idx;

}

static int FindMaxIdx(int[,] arr, int n)

{

int idx = 0;

int tmp = arr[0, 0];

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

{

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

{

if (arr[i, j] > tmp)

{

idx = i;

tmp = arr[i, j];

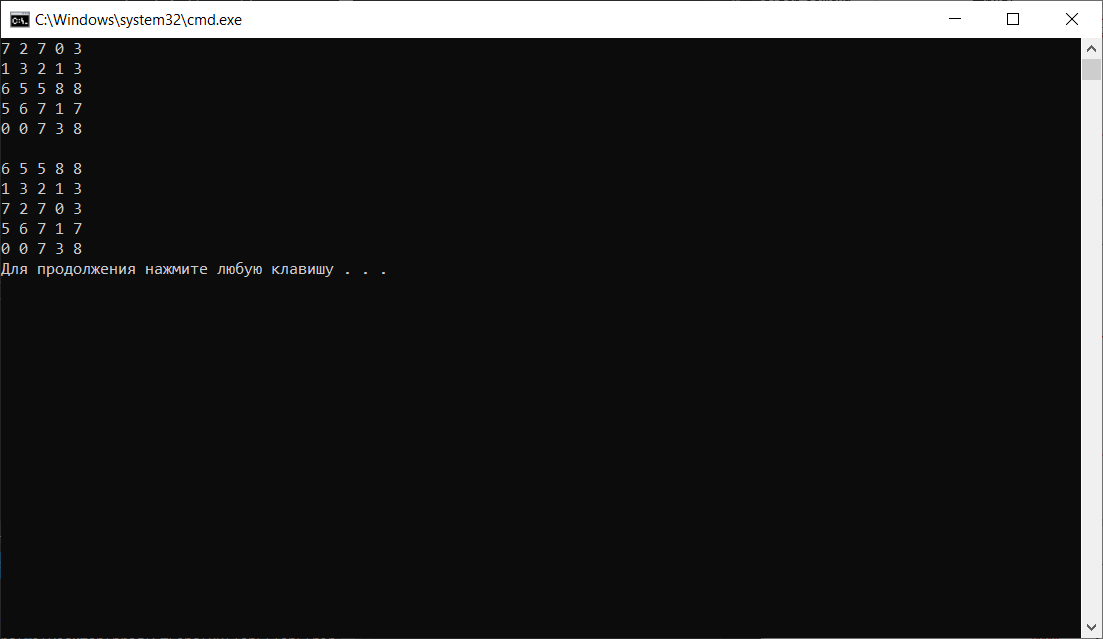
}

}

}

return idx;

}



1. Найти сумму элементов, расположенных на закрашенной части матрицы.



static int SumTriangle(int[,] matrix, int n)

{

int sum = 0;

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

{

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

{

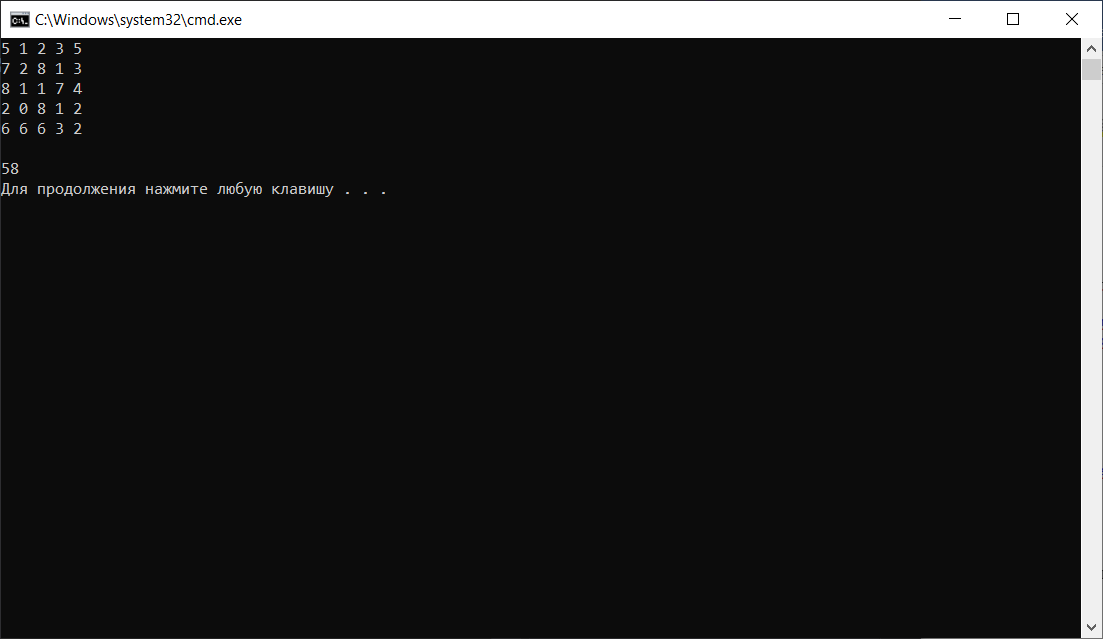
sum += matrix[i, j];

}

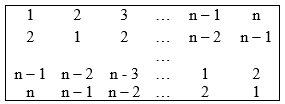
}

return sum;

}



1. Получить заданную матрицу порядка n



static int[,] fillMatrix(int n)

{

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

int k = n;

for (int i = n - 1; i >= 0; i--, k--)

{

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

{

matrix[i, j] = k - j;

}

}

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

{

k = 1;

for (int j = i+1; j < n; j++)

{

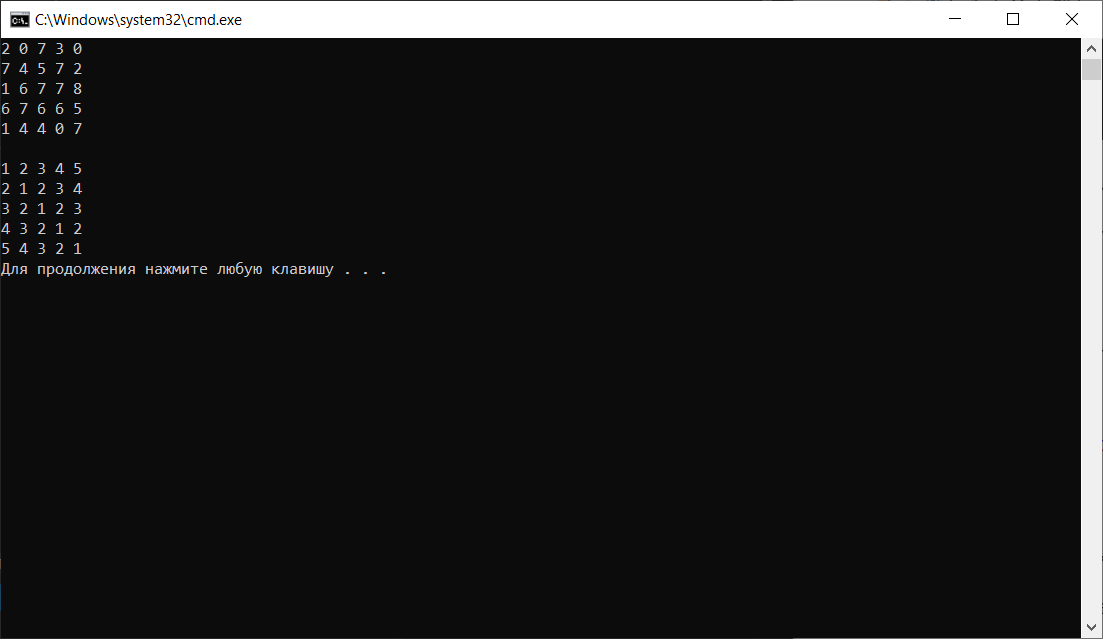
matrix[i, j] = ++k;

}

}

return matrix;

}



1. Дана действительная матрица размера nxm. Упорядочить столбцы по неубыванию первых элементов.

static int[,] MatrixSort(int [,] arr, int n)

{

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

{

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

{

if (arr[0, i] < arr[0, j])

{

arr = swapCols(arr, n, i, j);

}

}

}

return arr;

}

static int[,] swapCols (int[,] matrix, int n, int first, int second)

{

int[,] newMatrix = new int[n, n];

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

{

newMatrix[i, first] = matrix[i, second];

}

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

{

newMatrix[i, second] = matrix[i, first];

}

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

{

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

{

if (i != first && i != second)

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

}

}

return newMatrix;

}

