**C# Лабораторна робота 7**

**ВАРІАНТ 4**

**Завдання 1**

Визначити добуток додатних елементів матриці вище головної діагоналі

class Program

{

static void Main(string[] args)

{

// Створення матриці

Console.Write("Розмiр матрицi = ");

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

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

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

{

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

{

Console.Write("Елемент [{0}][{1}] = ", i, j);

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

}

}

// вихід

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

{

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

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

Console.WriteLine();

}

// пошук

int element = 1;

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

{

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

{

if (i < j && matrix[i, j] > 0)

element \*= matrix[i, j];

}

}

Console.WriteLine((element > 1) ? $"Добуток позитивних елементiв = {element}" : "Немає позитивних елементів вище основної діагоналі.");

}

}

}

**Завдання 2**

Дано дійсну матрицю розмірності , всі елементи якої різні. Знайти ска­ляр­ний добуток *i*-го рядка і *j*-го стовпчика (*i*, *j* задаються користувачем).

class Program

{ static void Main(string[] args)

{ // Matrix creation

Console.Write("Розмiр матрицi = ");

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

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

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

{

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

{

Console.Write("Елеметн [{0}][{1}] = ", i, j);

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

} }

// Matrix вихід

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

{

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

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

Console.WriteLine();

}

// Matrix пошук

int[] vectorRow = new int[size];

int[] vectorColumn = new int[size];

Console.Write("Номер рядка = ");

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

if (rowNumber >= size || rowNumber < 0)

Console.WriteLine("Номер стовпця повинен бути в дiапазонi [0;{0}]", size);

else

{ // вибір ряду

for (int i = rowNumber; i <= rowNumber; i++)

{

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

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

} }

Console.Write("Номер рядка = ");

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

if (columnNumber >= size || columnNumber < 0)

Console.WriteLine("Номер стовпця повинен бути в дiапазонi [0;{0}]", size);

// вибір стовпця

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

{

for (int j = columnNumber; j <= columnNumber; j++)

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

}

// скалярний добуток

int result = 0;

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

{

result += vectorRow[i] \* vectorColumn[i];

}

Console.WriteLine(result);

}

}

}

**Завдання 3**

Дано матриці  і . Знайти матрицю .

class Program

{

//створюю матрицю

static int[,] MatrixCreation(int row, int column)

{

int[,] matrix = new int[row, column];

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

{

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

{

Console.Write("Елемент [{0}][{1}] = ", i, j);

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

} } return matrix; }

// виводить матрицю

static void MatrixOutput(int[,] matrix, int rows, int columns)

{

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

{

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

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

Console.WriteLine();

} }

static void Main(string[] args)

{ // matrix A

Console.Write("Рядки = ");

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

Console.Write("Стовпцi = ");

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

int[,] A = (MatrixCreation(rA, columnsA));

MatrixOutput(A, rA, columnsA);

// matrix B

Console.Write("Рядки = ");

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

Console.Write("Стовпцi = ");

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

int[,] B = MatrixCreation(rB, columnsB);

MatrixOutput(B, rB, columnsB);

if (columnsA != rA)

Console.Write("Кількість стовпців у першій матриці повинна дорівнювати кількості рядків у другій.");

else

{ // matrix C

int[,] C = new int[columnsA, rB];

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

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

{ C[i, j] = 0;

for (int k = 0; k < A.GetLength(1); k++)

C[i, j] = C[i, j] + A[i, k] \* B[k, j];

} }

MatrixOutput(C, columnsA, rB);

} } }}

**Завдання 4**

Розмістити елементи непарних стовпців у порядку зростання.

class Program

{ // транспортування матриці

static int[][] TransposedMatrix(int[][] matr)

{ int rowCount = matr.Length;

int columnCount = matr[0].Length;

int[][] transposed = new int[columnCount][];

transposed = (int[][])matr.Clone();

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

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

{

int temp = transposed[i][j];

transposed[i][j] = transposed[j][i];

transposed[j][i] = temp;

} } return transposed; }

// матричний вихід

static void MatrixOutput(int[][] matr)

{ int rowCount = matr.Length;

int columnCount = matr[0].Length;

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

{ for (int j = 0; j < columnCount; j++)

{

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

}

Console.WriteLine();

} }

static void Main(string[] args)

{ // створення матриці

Console.Write("Рядки = ");

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

Console.Write("Стовпцi = ");

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

int[][] matrix = new int[rows][];

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

{ matrix[i] = new int[columns];

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

{

Console.Write("Елеметн = ");

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

} }

Console.WriteLine("Матриця:");

MatrixOutput(matrix);

Console.WriteLine();

int[][] transposed = TransposedMatrix(matrix);

// сортування рядків транспонованої матриці

for (int i = 0; i < transposed.Length; i += 2)

{

Array.Sort(transposed[i]);

}

int[][] res = TransposedMatrix(transposed);

Console.WriteLine("Результат матрицi: ");

MatrixOutput(res);

}

}

}

**Завдання 5**

Дана цілочислова квадратна матриця. Визначити добуток елементів в тих рядках, які не містять від’ємних елементів.

class Program

{

static void Main(string[] args)

{

Console.Write("Кiлькiсть рядкiв = ");

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

Console.Write("Кiлькiсть стовпцiв = ");

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

int[][] matrix = new int[rows][];

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

{

matrix[i] = new int[columns];

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

{

Console.Write("Елемент = ");

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

}

}

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

{

if (matrix[i].Min() > 0)

{

int product = 1;

foreach (int element in matrix[i])

product \*= element;

Console.WriteLine($"Виріб елементів в { i + 1} рядок є {product}.");

}

else

Console.WriteLine($"{i + 1} row contains negative element.");

}

}

}

}

**Завдання 6**

Дана цілочислова квадратна матриця. Визначити максимум серед сум елементів діагоналей, паралельних головній діагоналі матриці.

class Program

{

//створю матрицю

static int[,] MatrixCreation(int size)

{ int[,] matrix = new int[size, size];

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

{

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

{

Console.Write("Елемент [{0}][{1}] = ", i, j);

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

} }

return matrix;

}

// виводить матрицю

static void MatrixOutput(int[,] matrix, int size)

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

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

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

Console.WriteLine();

} }

static void Main(string[] args)

{

Console.Write("Розмiр матрицi = ");

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

int[,] matrix = MatrixCreation(size);

MatrixOutput(matrix, size);

int maxSum = -10;

// прокручування елементів над головною діагоналлю

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

{

int i = 0;

int sum = 0;

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

{

sum += matrix[i, j];

i++;

}

if (sum > maxSum)

maxSum = sum;

}

// через елементи нижче основної діагоналі

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

{ int j = 0;

int sum = 0;

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

{

sum += matrix[i, j];

j++;

}

if (sum > maxSum)

maxSum = sum;

}

Console.WriteLine(maxSum);

}

}

}