Лабораторна робота №7

Завдання 1. Визначити суму додатних елементів матриці з непарною сумою індексів.

using System;

using System.Collections.Generic;

using System.Linq;

namespace task1

{

class Program

{

static void Main(string[] args)

{

int n, m;

Console.Write("К-сть рядкiв: ");

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

Console.Write("К-сть стоапцiв: ");

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

int[,] A = new int[n, m];

int sum = 0;

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

{

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

{

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

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

}

}

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

{

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

{

if ((i + j) % 2 == 1 && A[i, j] > 0)

sum += A[i, j];

}

}

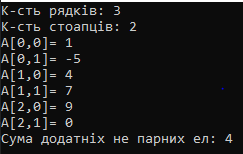
Console.WriteLine($"Сума додатнiх не парних ел: {sum}");

}

}

}

Контрольний приклад.



Завдання 2 Побудувати прямокутну матрицю *А*, елементи якої задаються формулою:



using System;

namespace task2

{

class Program

{

static double Matrics(int i, int j, int x)

{

return i \* (Math.Sin(i \* x) + Math.Cos(j \* x));

}

static void Main(string[] args)

{

int n, m, x;

Console.Write("К-ть рядкiв =");

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

Console.Write("К-ть стовпцiв =");

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

Console.Write("x =");

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

double[,] A = new double[n, m];

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

{

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

A[i, j] = Matrics(i, j, x);

}

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

{

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

if (A[i, j] < x)

Console.WriteLine($"{A[i, j]}");

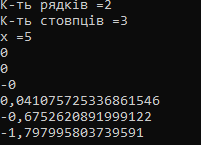
}

}

}

}

Контрольний приклад.



Завдання 3. Дано матрицю  і вектор . Знайти вектор .

using System;

namespace task3

{

class Program

{

static void Main(string[] args)

{

int n, m,k=-1;

Console.Write("К-ть рядкiв: ");

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

Console.Write("К-ть стовцiв: ");

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

int[,] A = new int[n, m];

int[] Vect = new int[m];

int[,] B = new int[m, n];

int[] Answer = new int[m];

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

{

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

{

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

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

}

}

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

{

Console.Write($"Введiть вект:{i}= ");

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

}

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

{

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

B[j, i] = A[i, j];

}

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

{

k++;

if (k == m) break;

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

Answer[i] = Vect[k] \* B[j, i];

}

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

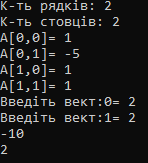
Console.WriteLine(Answer[i]);

}

}

}

Контрольний приклад.



Завдання 4. Розмістити елементи непарних рядків у порядку спадання.

Контрольний приклад.

int[] a = { 0, 5, 1, 2, 85 };

int temp;

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

{

for (int j = i + 1; j <a.Length; j++)

{

if (a[i] > a[j])

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

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

{

Console.WriteLine(a[i]);

}

# Створюємо відсортований список списків  
lis = []  
for el in A[::2]:  
 B = [sorted(el, reverse=True)]  
 lis += B  
 # Замінюємо списки на відсортовані  
A[::2] = lis  
for i in A:  
 print(i)

Завдання 5. Дана цілочислова прямокутна матриця. Визначити суму елементів в тих стовпцях, які містять хоча б один від’ємний елемент.

n = int(input("Кількість рядків та стовпців: "))  
A = [[int(input("El[{0}][{1}]= ".format(i, j))) for j in range(1, n + 1)] for i in range(1, n + 1)]  
matrix = [[0] \* n for i in range(n)]  
for j in range(n):  
 for i in range(n):  
 matrix[j][i] = A[i][j]  
for i in matrix:  
 print(" " + str(i))  
collumn = 0  
for el in matrix:  
 count = 0  
 m = list(filter(lambda x: x < 0, el))  
 if len(m) < 1:  
 collumn += 1  
 continue  
 else:  
 collumn += 1  
 count = sum(el)  
 print("Сума чисел стовпця № {0}: {1}".format(collumn, count))

Контрольний приклад.

Завдання 6. Дана цілочислова прямокутна матриця. Визначити номера рядків і стовпців всіх сідлових точок матриці. Матриця А має сідлову точку А0, якщо Aij є мінімальним елементом в і-у рядку і максимальним в j-у стовпці.

def saddle\_point(matrix):  
 if len(matrix) == 1:  
 return 0  
 y = 0  
 while len(matrix) > y:  
 for i in matrix:  
 l\_min = min(i)  
 l\_index = i.index(l\_min)  
 for j in matrix:  
 if l\_min > j[l\_index]:  
 return matrix.index(i), l\_index  
 return None  
 y += 1  
  
  
b = saddle\_point([[0, 2, -1], [1, 2, -2]])  
print(b)

Контрольний приклад.