**Кафедра інженерії програмного забезпечення та кібербезпеки**

ЛАБОРАТОРНА РОБОТА №3

**РОБОТА З МАСИВАМИ**

|  |  |
| --- | --- |
| **Виконав:** | *Лупина Ігор Ігорович* |
| **Група:** | *1-6* |
| **Варіант:** | *12* |
|  |  |
| **Викладач:** | *Макоєдова Валентина Олександрівна* |
| **Бали:** |  |

|  |  |
| --- | --- |
| **освітній ступінь** | **бакалавр /** bachelor |
| **галузь знань** | **12 Інформаційні технології /** Information Technology |
| **спеціальність** | **121Інженерія програмного забезпечення /** Software engineering |
|  |  |

**Київ 2023**

**Тема:** Робота з масивами

**Мета:** Закріпити вміння використовувати умовні та циклічні оператори при написанні програм, навчитись працювати з одновимірними і багатовимірними масивами

**Завдання:** Виконати задачі згідно варіанту

**Хід роботи**

**Задача 1:** В одномірному масиві, що складається з дійсних елементів, обчислити суму додатних елементів масиву; добуток елементів масиву, розташованих між максимальним за модулем і мінімальним за модулем елементами. Упорядкувати елементи масиву за спаданням.

Для виконання задачі 1 було написано наступну програму:

namespace Task1;

class Program

{

static void Main()

{

var arr = Enumerable.Range(0, 50)

.Select(\_ => Random.Shared.Next(-1000, 1000) / 100.0)

.ToArray();

Console.WriteLine("Initial array:");

PrintArray(arr);

Console.WriteLine();

Console.WriteLine($"Sum of positive numbers: {SumPos(arr):f2}");

Console.WriteLine("Product of elements between the absolute minimum " +

$"and absolute maximum: {MinMaxProduct(arr):f2}");

Console.WriteLine();

Console.WriteLine("Sorted array:");

PrintArray(SortByDescending(arr));

}

static void PrintArray(double[] arr)

{

for (int i = 0; i < arr.Length;)

{

Console.Write($"{arr[i],5} ");

i++;

if (i % 10 == 0)

{

Console.WriteLine();

}

}

}

static double SumPos(double[] data)

=> data.Where(x => x > 0).Sum();

static double MinMaxProduct(double[] data)

{

var absIndex = data.Select((x, i) => new { Value = Math.Abs(x), Index = i });

var min = absIndex.MinBy(x => x.Value)!;

var max = absIndex.MaxBy(x => x.Value)!;

if (min.Index > max.Index)

{

(min, max) = (max, min);

}

var range = data[(min.Index + 1)..max.Index];

return range.Length == 0 ? 0 : range.Aggregate(1.0, (x, y) => x \*= y);

}

static double[] SortByDescending(double[] data)

{

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

{

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

{

if (data[j] < data[j + 1])

{

(data[j + 1], data[j]) = (data[j], data[j + 1]);

}

}

}

return data;

}

}

В програмі присутні наступні методи:

* PrintArray – виводить масив дійсних чисел в консоль по 10 чисел в ряду
* SumPos – повертає суму додатних елементів масиву дійсних чисел
* MinMaxProduct – повертає добуток чисел, розташованих між елементами з найбільшим та найменшим модулями масиву дійсних чисел
* SortByDescending – сортує масив дійсних чисел за спаданням бульбашковим методом

Результат виконання програми зображено на рисунку 1.

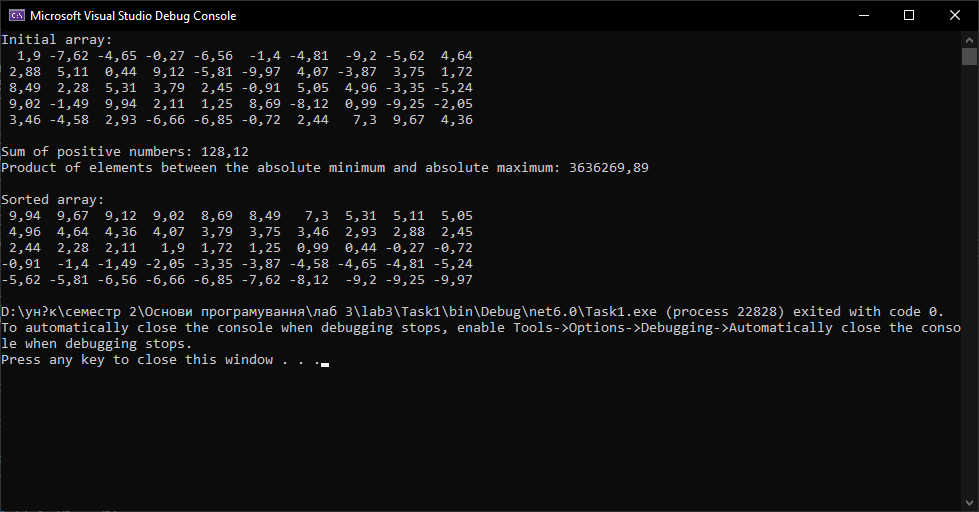


Рисунок 1 – Результат виконання програми задачі 1

**Задача 2:** Створити числовий масив, у якому міститься 30 елементів. Відсортувати елементи масиву двома способами – один з лекції, інший розібрати самостійно та застосувати. Вбудовані методи сортування використовувати заборонено.

Для виконання задачі 2 було написано наступну програму:

namespace Task2;

class Program

{

static void Main()

{

var initialArray = Enumerable.Range(0, 30)

.Select(\_ => Random.Shared.Next(0, 100))

.ToArray();

var copy = new int[initialArray.Length];

Console.WriteLine("Initial array:");

PrintArray(initialArray);

Console.WriteLine();

Console.WriteLine("Sorted array (Bubble Sort): ");

initialArray.CopyTo(copy, 0);

PrintArray(BubbleSort(copy));

Console.WriteLine();

Console.WriteLine("Sorted array (Insertion Sort): ");

initialArray.CopyTo(copy, 0);

PrintArray(InsertionSort(initialArray));

}

static void PrintArray(int[] arr)

{

for (int i = 0; i < arr.Length;)

{

Console.Write($"{arr[i],2} ");

i++;

if (i % 10 == 0)

{

Console.WriteLine();

}

}

}

static int[] BubbleSort(int[] data)

{

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

{

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

{

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

{

(data[j + 1], data[j]) = (data[j], data[j + 1]);

}

}

}

return data;

}

static int[] InsertionSort(int[] data)

{

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

{

int key = data[i];

int j = i - 1;

while (j >= 0 && key < data[j])

{

data[j + 1] = data[j];

--j;

}

data[j + 1] = key;

}

return data;

}

}

В програмі присутні наступні методи:

* PrintArray – виводить масив цілих чисел в консоль по 10 чисел в ряду
* BubbleSort – сортує масив цілих чисел бульбашковим методом
* InsertionSort – сортує масив цілих чисел методом вставки

Результат виконання програми зображено на рисунку 2.

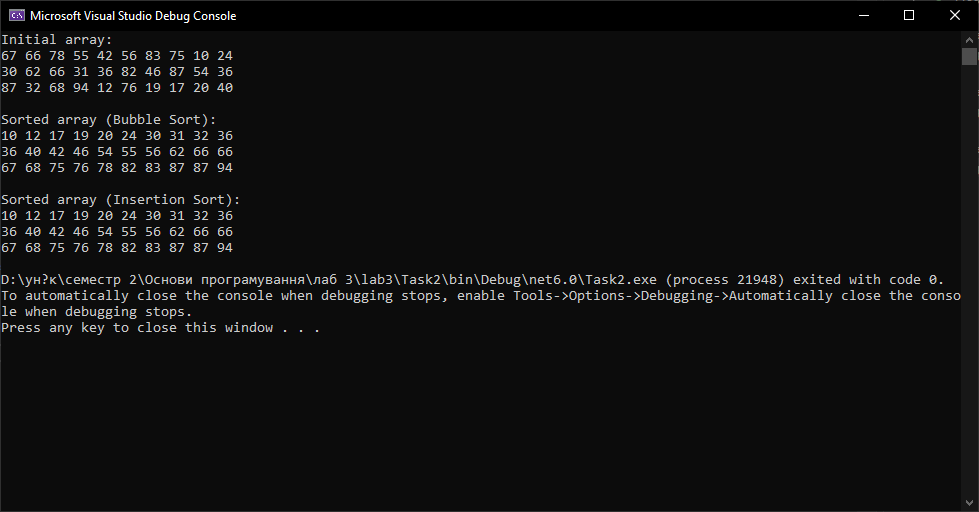


Рисунок 2 – Результат виконання програми задачі 2

**Задача 3:** Для масиву цілих чисел довжиною 120 елементів з випадково згенерованими елементами в діапазоні від -70 до 70 визначити середню довжину неперервних додатних послідовностей масиву (проміжні значення теж виводити).

Для виконання задачі 3 було написано наступну програму:

namespace Task3;

class Program

{

static void Main()

{

const int size = 120;

var arr = Enumerable.Range(0, size).Select(\_ => Random.Shared.Next(-70, 70))

.ToArray();

Console.WriteLine("Initial array");

for (int i = 0; i < arr.Length;)

{

Console.Write($"{arr[i],3} ");

i++;

if (i % 15 == 0) { Console.WriteLine(); }

}

Console.WriteLine();

var seq = GetPositiveSequences(arr);

Console.WriteLine("Sequences: ");

Console.WriteLine("First Last Length Sequence");

foreach (var sequence in seq)

{

Console.WriteLine($"{sequence.Offset,-5} " +

$"{sequence.Offset + sequence.Count - 1,-4} " +

$"{sequence.Count,-6} " +

$"{string.Join(' ', sequence)}");

}

Console.WriteLine();

Console.WriteLine($"Average sequence length: {seq.Average(x => x.Count):f2}");

}

static List<ArraySegment<int>> GetPositiveSequences(int[] arr)

{

var sequences = new List<ArraySegment<int>>();

int sequenceStart = -1;

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

if (sequenceStart == -1 && arr[i] > 0) { sequenceStart = i; continue; }

if (sequenceStart != -1 && arr[i] <= 0) {

sequences.Add(new(arr, sequenceStart, i - sequenceStart));

sequenceStart = -1;

}

}

return sequences;

}

}

Метод GetPositiveSequences повертає список сегментів масиву цілих чисел, де наявні лише додатні значення. Результат виконання програми зображено на рисунку 3.

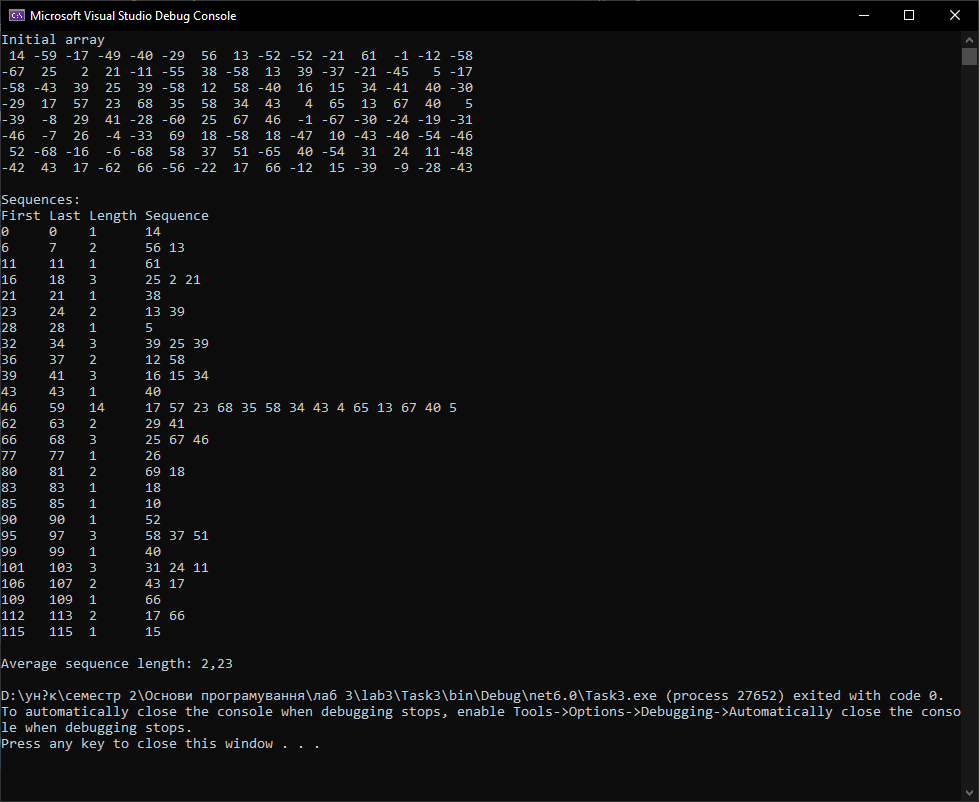


Рисунок 3 – Результат виконання програми задачі 3

**Задача 4:** Для двовимірного масиву nхn:

* заповнити сектори матрицi, якi лежать вище i нижче вiд головної та бiчної дiагоналей, значеннями вектора b1, b2, ... , b32, вiд лiвого верхньго кута вправо - вниз. Решта матрицi заповнити нулями.
* визначити середню довжину неперервних додатних послідовностей масиву (проміжні значення теж виводити).

Для виконання задачі 4 було написано наступну програму:

namespace Task4;

class Program {

static void Main() {

var initialMatrix = new int[12, 12];

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

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

initialMatrix[j, k] = Random.Shared.Next(-70, 70);

Console.WriteLine("Initial matrix:");

PrintMatrix(initialMatrix);

Console.WriteLine();

Console.WriteLine("Altered matrix:");

PrintMatrix(TransformMatrix(initialMatrix));

Console.WriteLine();

int[][] jagged = new int[4][];

jagged[0] = new int[6];

jagged[1] = new int[2];

jagged[2] = new int[4];

jagged[3] = new int[11];

Console.WriteLine("Initial jagged array:");

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

for (int j = 0; j < jagged[i].Length; j++) {

jagged[i][j] = Random.Shared.Next(-70, 70);

Console.Write($"{jagged[i][j],3} ");

}

Console.WriteLine();

}

Console.WriteLine();

var seq = GetPositiveSequences(jagged);

Console.WriteLine("Sequences: ");

foreach (var sequence in seq)

Console.WriteLine(string.Join(' ', sequence));

Console.WriteLine();

Console.WriteLine($"Average sequence length: {seq.Average(x => x.Length):f2}");

}

static void PrintMatrix(int[,] arr) {

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

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

Console.Write($"{arr[i, j],3} ");

Console.WriteLine();

}

}

static int[,] TransformMatrix(int[,] array) {

int val = 1;

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

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

bool isOk = (i > j && array.GetLength(1) - j - 1 < i)

|| (i < j && array.GetLength(1) - j - 1 > i);

array[i, j] = isOk ? val++ : 0;

}

return array;

}

static int[][] GetPositiveSequences(int[][] arr) {

var res = new List<int[]>();

var sequence = new List<int>();

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

for (int j = 0; j < arr[i].Length; j++) {

var value = arr[i][j];

if (!sequence.Any() && value > 0) {

sequence.Add(value);

continue;

}

if (sequence.Any()) {

if (value <= 0) {

res.Add(sequence.ToArray());

sequence.Clear();

continue;

}

sequence.Add(value);

}

}

}

return res.ToArray();

}

}

В програмі присутні наступні методи:

* PrintMatrix – виводить матрицю (двовимірний масив) цілих чисел в консоль
* TransformMatrix – замінює всі значення головної і бічної діагоналей, а також елементи справа і зліва від них нулями, елементи що лишилися замінюються натуральними числами починаючи з одиниці справа наліво згори донизу
* GetPositiveSequences – повертає масив послідовностей цілих чисел збережених в масиві

Результат виконання програми зображено на рисунку 4.

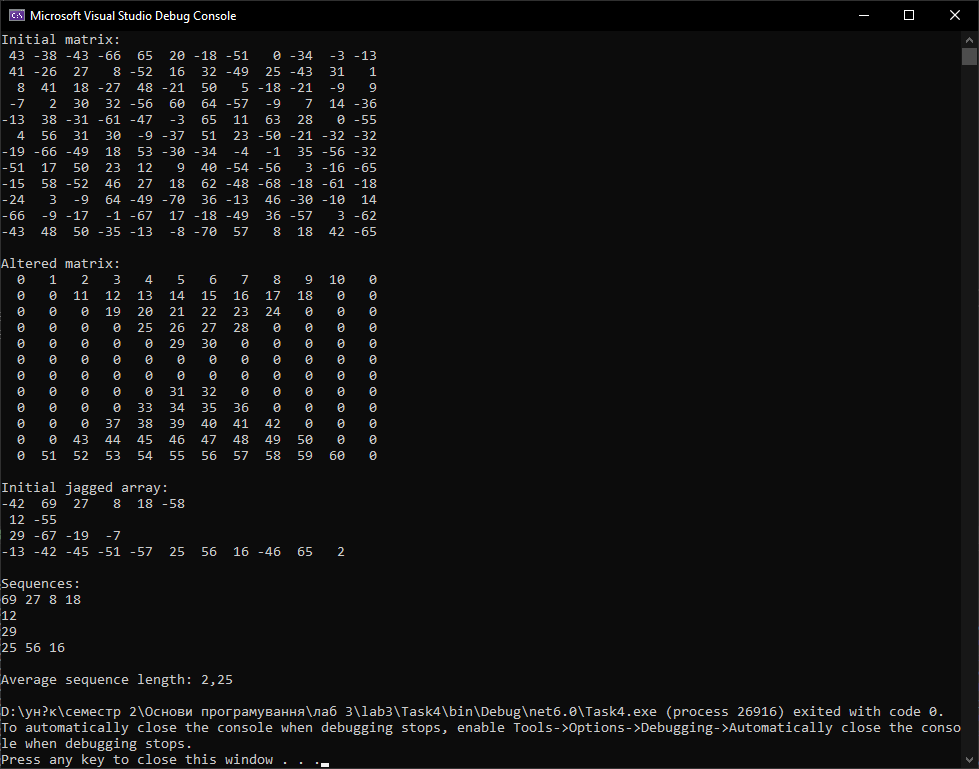


Рисунок 4 – Результат виконання програми задачі 4

**Висновок:** Під час виконання практичної роботи я закріпив вміння використовувати умовні та циклічні оператори при написанні програм, навчився працювати з одновимірними і багатовимірними масивами.