МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**ЗВІТ**

**з лабораторної роботи № 2**

**«Алгоритми обходу матриці**

**121 – Інженерія програмного забезпечення. 2021–2022 н.р.»**

# Дисципліна «Алгоритми та структури даних 1. Основи алгоритмізації»

**Виконала :**

студентка 1-го курсу,

групи КП-11,

спеціальності 121 – Інженерія

програмного забезпечення

*Кирильчук Олександра Артурівна*

Київ – 2021

**Варіант**

|  |  |  |
| --- | --- | --- |
| 12. | Кирильчук Олександра Артурівна | 9 |

**Задача**

Заданий двовимірний масив (далі – матриця) розміру **N×M**. Потрібно здійснити обхід всіх елементів цієї матриці згідно з варіантом.

**Вхідні дані:** 𝑁, 𝑀 ∈ ℕ

Результатом роботи програми має бути:

1) Виведена матриця, для якої здійснюється обхід

2) Рядок, що складається з послідовності елементів відповідно до шляху обходу

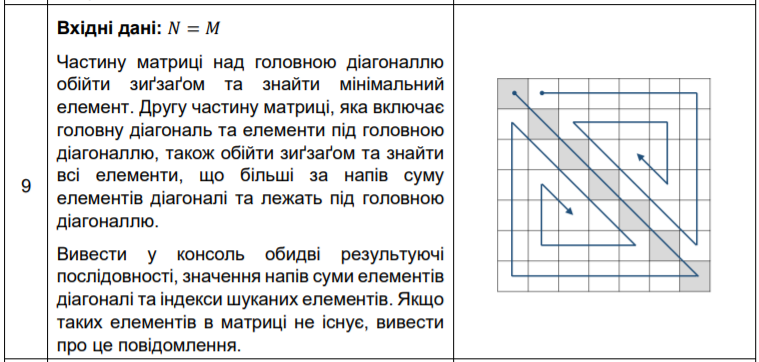
Розмір матриці повинен задаватися користувачем. При цьому, має здійснюватися

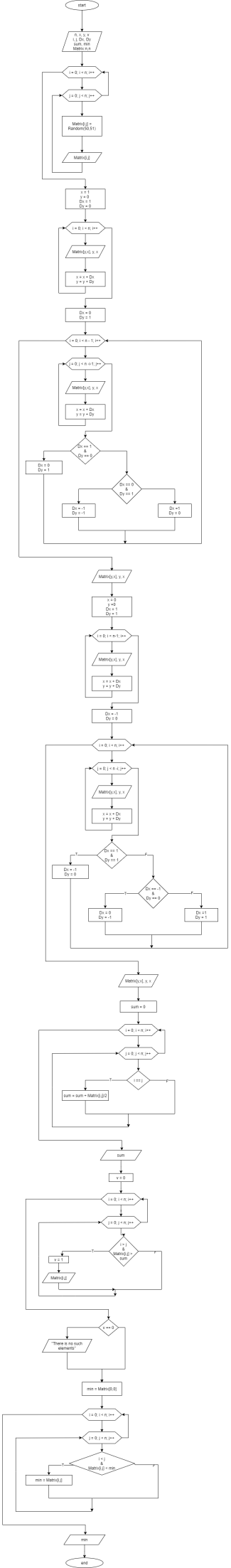
перевірка коректності вхідних даних відповідно до варіанту. Елементи матриці повинні бути цілими додатними числами та генеруватися псевдовипадково.

Для перевірки правильності роботи алгоритму повинен бути передбачений контрольний приклад – матриця, що заповнюється елементами від 0 до 𝑁 ∙ 𝑀.

Користувацьке меню має включати в себе можливість генерування псевдовипадкової матриці та використання контрольного прикладу.

Тоді, коли це виправдано, можна використовувати користувацькі функції. При цьому, в блок-схемі функція позначається як наперед визначений алгоритм та зображується у вигляді окремої блок-схеми.



****

**Код:**

using System;

namespace asd\_1

{

class Program

{

static void Main(string[] args)

{

int n = 7;

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

Console.WriteLine("Generated Matrix :");

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

{

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

{

Random rnd = new Random();

Matrix[i, j] = rnd.Next(-50, 51);

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

}

Console.WriteLine();

}

Console.WriteLine(" ");

Console.WriteLine("Above the main diagonal :");

{

int x = 1, y = 0, Dx = 1, Dy = 0;

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

{

Console.WriteLine("Element : " + $"{ Matrix[y, x],4}" + " [" + $"{ y,2}" + "] [" + $"{ x,2}" + "] ");

x += Dx;

y += Dy;

}

Dx = 0; Dy = 1;

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

{

for (int j = 0; j < n - i - 1; j++)

{

Console.WriteLine("Element : " + $"{ Matrix[y, x],4}" + " [" + $"{ y,2}" + "] [" + $"{ x,2}" + "] ");

x += Dx;

y += Dy;

}

if (Dx == 1 && Dy == 0)

{

Dx = 0;

Dy = 1;

}

else if (Dx == 0 && Dy == 1)

{

Dx = -1;

Dy = -1;

}

else

{

Dx = 1;

Dy = 0;

}

}

Console.WriteLine("Element : " + $"{ Matrix[y, x],4}" + " [" + $"{ y,2}" + "] [" + $"{ x,2}" + "] ");

}

Console.WriteLine();

Console.WriteLine("The main diagonal and under it :");

{

int x = 0, y = 0, Dx = 1, Dy = 1;

for (int i = 0; i < n - 1; i++)

{

Console.WriteLine("Element : " + $"{ Matrix[y, x],4}" + " [" + $"{ y,2}" + "] [" + $"{ x,2}" + "] ");

x += Dx;

y += Dy;

}

Dx = -1; Dy = 0;

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

{

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

{

Console.WriteLine("Element : " + $"{ Matrix[y, x],4}" + " [" + $"{ y,2}" + "] [" + $"{ x,2}" + "] ");

x += Dx;

y += Dy;

}

if (Dx == 1 && Dy == 1)

{

Dx = -1;

Dy = 0;

}

else if (Dx == -1 && Dy == 0)

{

Dx = 0;

Dy = -1;

}

else

{

Dx = 1;

Dy = 1;

}

}

Console.WriteLine("Element : " + $"{ Matrix[y, x],4}" + " [" + $"{ y,2}" + "] [" + $"{ x,2}" + "] ");

Console.WriteLine(" ");

}

{

double sum = 0;

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

{

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

{

if (i == j)

{

sum += Matrix[i, j] / 2.0;

}

}

}

Console.WriteLine("Half A Sum = " + sum);

int v = 0;

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

{

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

{

if (Matrix[i, j] > sum && i > j)

{

v = 1;

Console.WriteLine("Element < than half a sum : " + $"{ Matrix[i, j],4}");

}

}

}

if (v == 0)

{

Console.WriteLine("There is no such elements");

}

Console.WriteLine(" ");

int min = Matrix[0, 0];

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

{

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

{

if ( i < j && Matrix[i,j] < min)

{

min = Matrix[i, j];

}

}

}

Console.WriteLine("Min element is " + $"{ min, 4}");

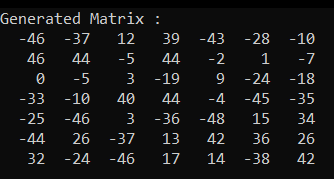
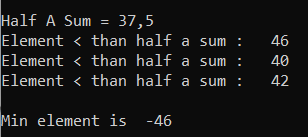
}

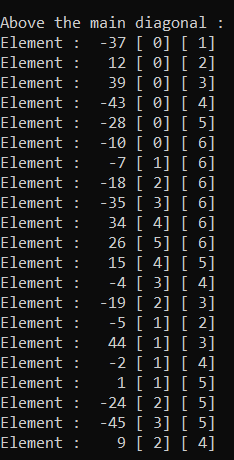
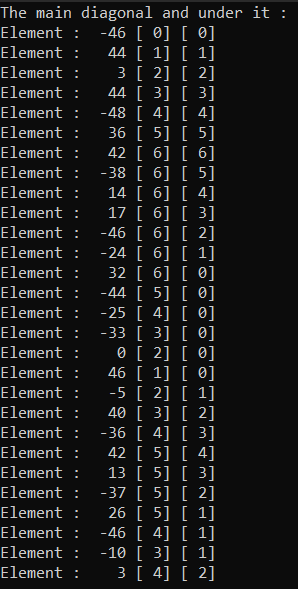
}

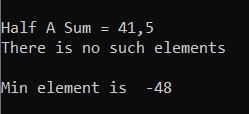
}

}

**Результат:**

****

**** 

****