МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпропетровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №3**

**з дисципліни «Основи програмування»**

**на тему: «Розробка програм з циклічною обробкою даних»**

Виконав: студент гр. ПЗ1911

Сіньков Г.О.

Прийняла: ас. каф. КІТ

Нежуміра О. І.

Дніпро, 2019

**Лабораторна робота №4**

**Тема.** Використання вкладених циклів в задачах обробки матриць.

**Мета.** Навчитися використовувати багатомірні масиви та вкладені цикли в розв’язанні задач обробки матриць.

**Постановка задачи**

1.Загальна постановка: розробити програму відповідно до обраного варіанту завдання.

2.Індивідуальна постановка: дана дійсна квадратна матриця порядку N. У рядках з від’ємним елементом на головній діагоналі знайти суму всіх елементів, в рядках з додатним елементом на головній діагоналі знайти найбільший з усіх елементів і його індекс в рядку, для рядків з нулем на головній діагоналі видати про це повідомлення.

3.Вимоги до програми:

-вхідні дані вводяться з клавіатури;

-передбачити перевірку вхідних даних на відповідність діапазону значень і некоректні символи;

-результати роботи програми виводяться на екран;

4.Вимоги до тексту програми:

-коментарі щодо призначення програми, її вхідних і вихідних даних;

-коментарі щодо призначення кожного блоку програми, дій окремих операторів для пояснення алгоритму;

-самодокументованість коду: всі ідентифікатори повинні мати назви, що відповідають суті змінних.

**Зовнішні специфікації**

1. Вхідні дані

Вхідні дані:

**Формат вхідних даних**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Найменування даних | Умовне позначення | Вимоги до даних | Приклад |
| 1 | Розмір матриці | n | Ціле число | 3 |
| 2 | Елементи матриці | arr[i][j] | Ціле число | 1 2 3  4 -5 6  7 8 0 |

2. Вихідні дані:

**Формат вихідних даних**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Найменування даних | Умовне позначення | Вимоги до даних | Приклад |
| 1 | Сума рядка | sum | Ціле число | 5 |
| 2 | Найбільше число | max | Ціле число | 3 |
| 3 | Повідомлення про нульовий елемент | - | Ціле число | 0 |

3. Функціональні вимоги до програми

Програма повинна реалізувати такі дії:

-введення даних у програму

-перевірку заданих даних

-вивід результату перевірки

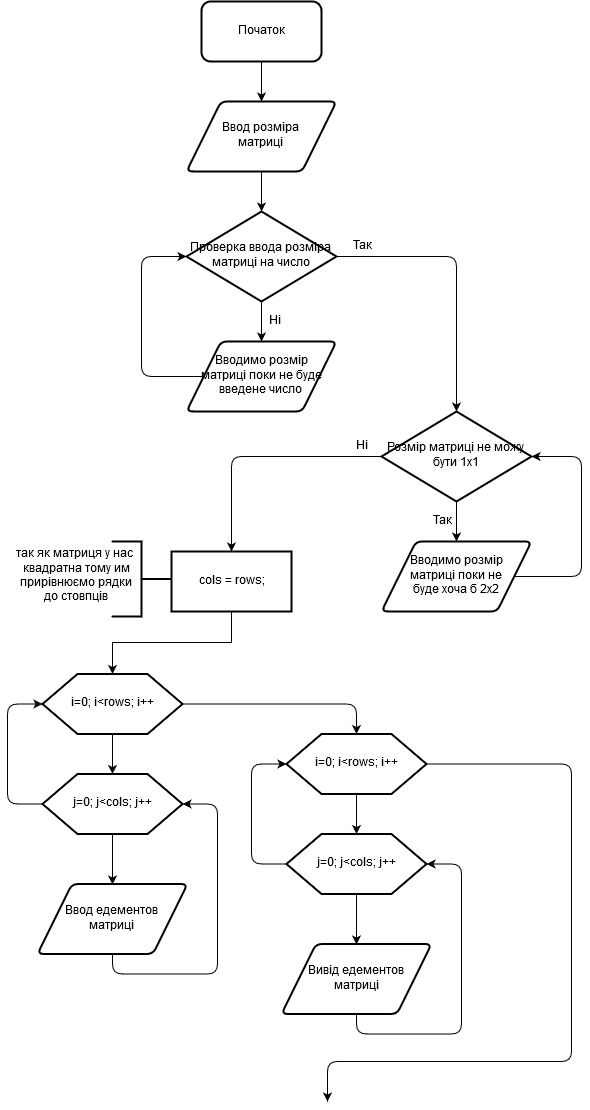
**Вибір методу рішення задачі**

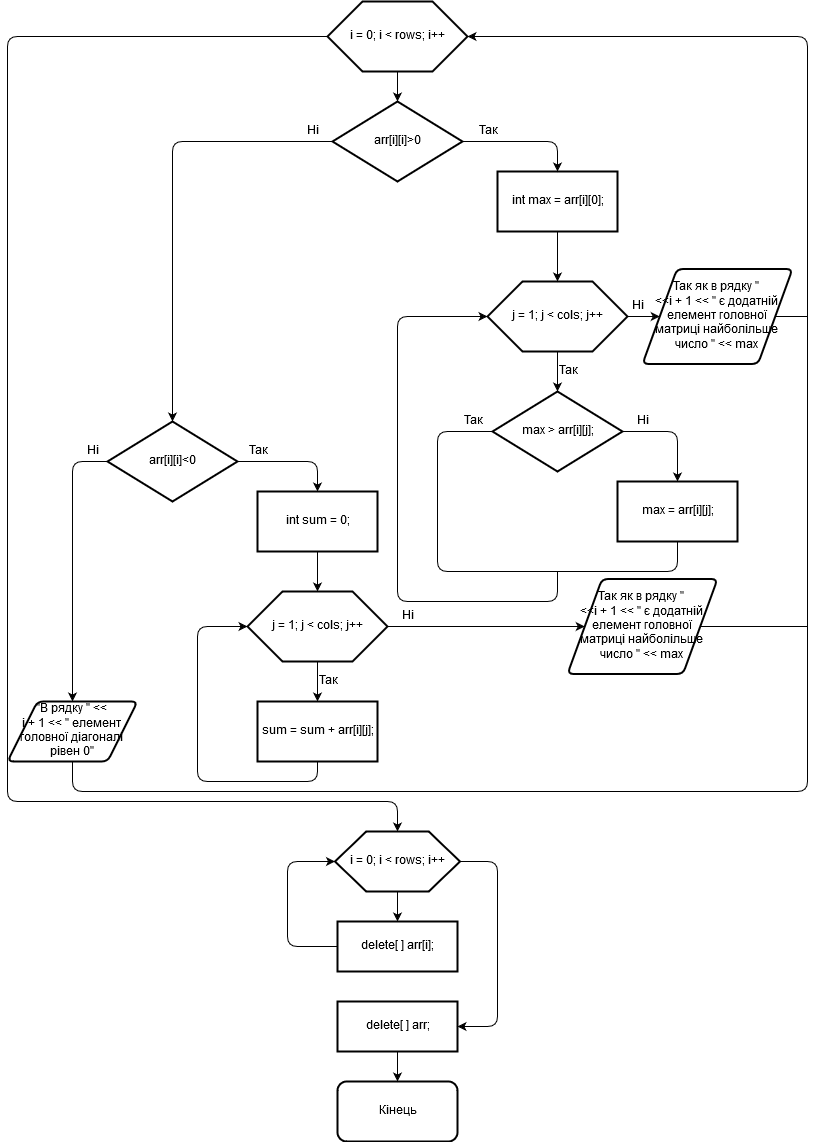
**Розробка тестів**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва | Вхідні дані | Очікувані результати |
| 1 | Некоректне значення числа | n = 1 | “ Розмір матриці повинен бути більше одного елемента! ” |
| 2 | Введене замість числа символ | n = f | “Помилка вводу! Повторить ввод. n має бути цілим” |
| 3 | Матриця | 1 2 3  4 -5 6  7 8 0 | Так як в рядку 1 є додатній елемент головної матриці найбільше число 3  Так як в рядку 2 є від’ємним елемент головної матриці сума рядка 5  В рядку 3 елемент головної діагоналі рівен 0 |

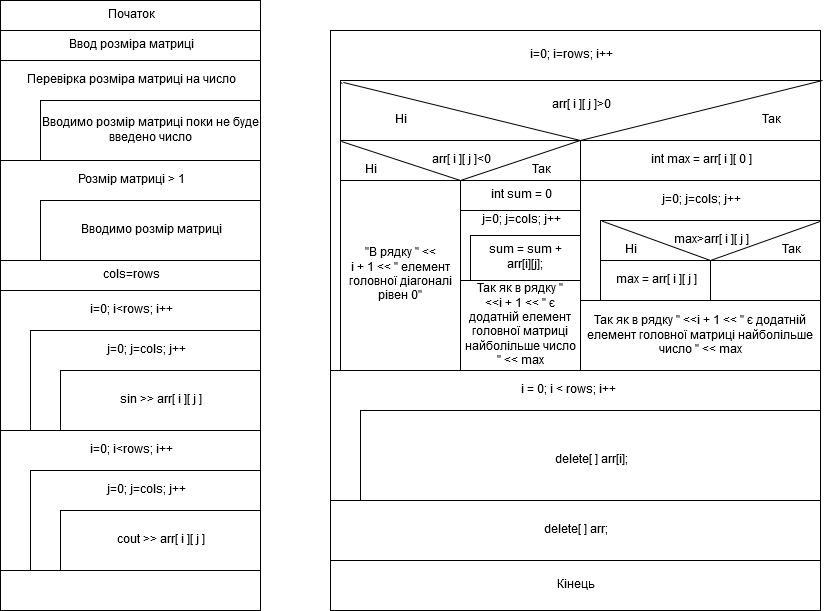
**Алгоритм програм**

1.Блок-схема





2.Діаграма Н-Ш



3.Текст програми

#include <iostream>

#include<Windows.h>

#include <cstdlib>

using namespace std;

void main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int rows, cols;

cout << "Введіть розмір матриці = ";

while (!(cin >> rows) || (cin.peek() != '\n')) // цикл для перевірки введеної зміної на символи

{

cin.clear();

while (cin.get() != '\n');

cerr << "Помилка вводу! Повторить ввод. n повино бути цілим" << endl;

cout << endl;

cout << "Введіть розмір матриці = ";

}

while (!(rows > 1) || (cin.peek() != '\n')) // цикл для перевірки введеної зміної розмір

{

cin.clear();

while (cin.get() != '\n');

cerr << "Розмір матриці повинен бути більше одного елемента!" << endl;

cout << endl;

cout << "Введіть розмір матриці = ";

cin >> rows;

}

cols = rows;

int\*\* arr = new int\*[rows];

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

{

arr[i] = new int[cols];

}

// Заповнення значень масива

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

{

cout << "Введіть " << i + 1 << " рядок матриці = " << endl;

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

{

cin >> arr[i][j];

}

}

// Вивід значень масива

cout << "\t Загальній вид матриці" << endl;

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

{

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

{

cout << "\t" << arr[i][j];

}

cout << endl;

}

cout << endl;

// Дії над елементами головної діагоналі матриці

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

{

if (arr[i][i] > 0)

{

int max;

max = arr[i][0];

for (int j = 1; j < rows; j++)

{

if (max > arr[i][j])

{

//cout << "Найбольшое число " << max << endl;

}

else

{

max = arr[i][j];

}

}

cout << "Так як в рядку " << i + 1 << " є додатній елемент головної матриці найболільше число " << max << endl;

cout << endl;

}

else if (arr[i][i] < 0)

{

int sum = 0;

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

{

sum = sum + arr[i][j];

}

cout << "Так як в рядку " << i + 1 << " є відємним елемент головної матриці сума рядка " << sum << endl;

cout << endl;

}

else

{

cout << "В рядку " << i + 1 << " елемент головної діагоналі рівен 0" << endl;

cout << endl;

}

}

///////////////////////////////

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

{

delete[] arr[i];

}

delete[] arr;

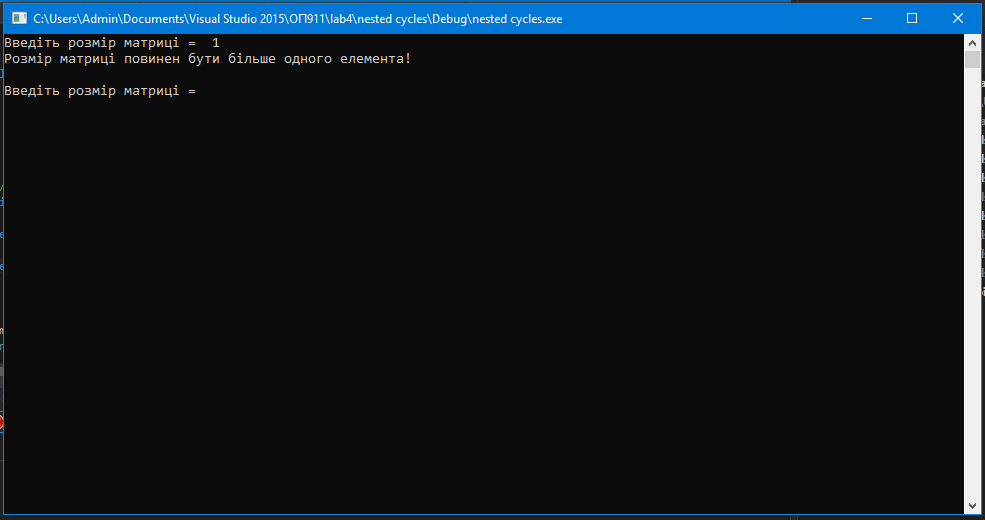
system("pause");

return;

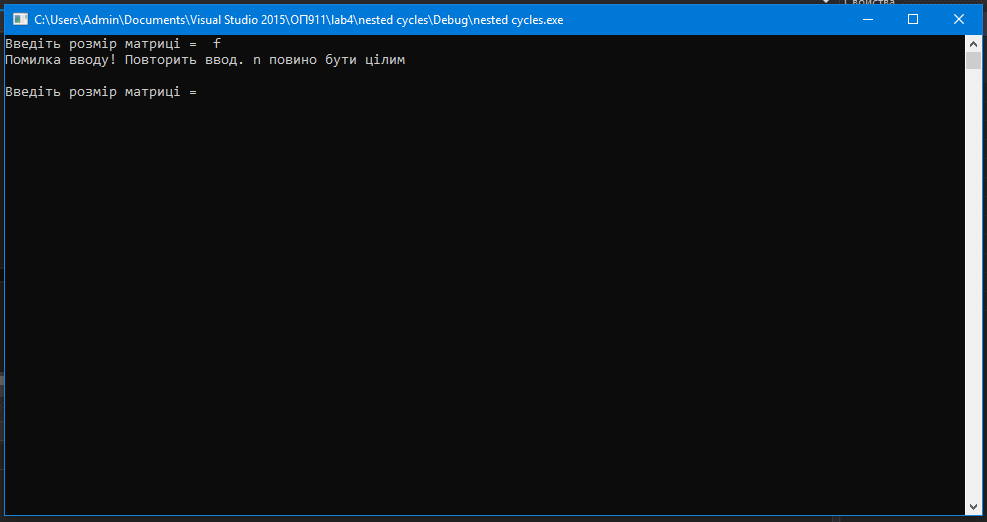
}

4.Результати виповнення програми

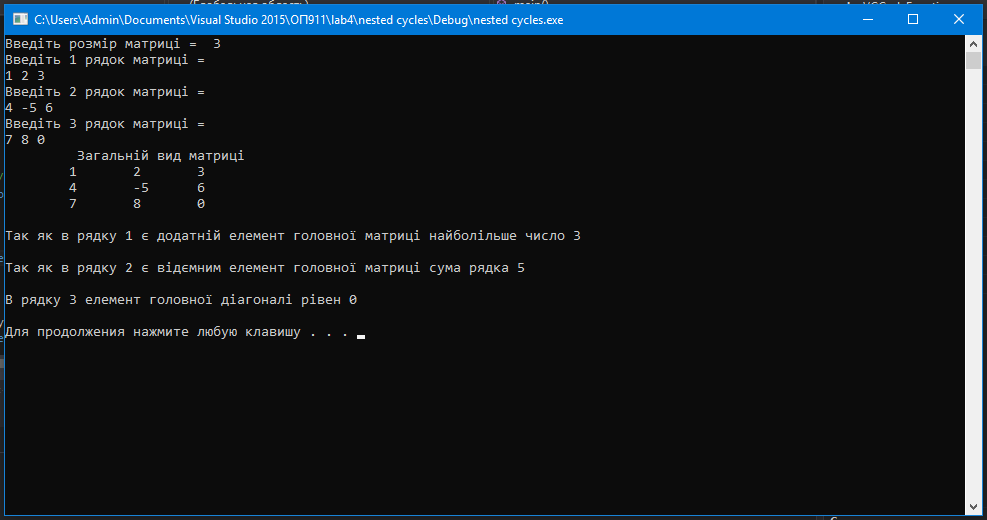
Перший тест:



Другий тест:



Третій тест:



Висновок: на практиці часто виникає необхідність в обробці даних у вигляді довільного набору значень, тобто масивів. Масив являє собою кінцеву іменовану послідовність величин одного типу, які розрізняються за порядковим номером. Для того щоб реалізувати ввод розміра матриці я використав динамічний масив, тому користувач може обрати любий розмір квадратної матриці.