Звіт

з лабораторної роботи № 8 з дисципліни

«Основи програмування»

Варіант 9

Виконала: Грицюк Христина Ярославівна

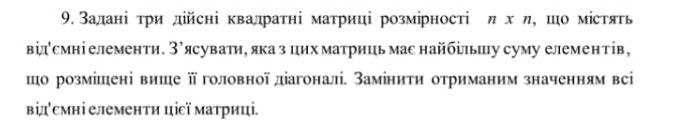
Київ 2020

**Лабораторна робота 8**

**Багатовимірні масиви**

**Мета роботи – опанувати технологію використання двовимірних масивів даних(матриць), навчитися розробляти алгоритми та програми із застосуванням матриць.**

**Варіант 9**

****

**1.Постановка задачі**

Дано три дійсні квадратичні матриці А,В,С, що містять від’ємні елементи. Спочатку знайдемо суму елементів матриць, що знаходяться над головною діагоналлю, використовуючи умову, що індекс стовпчика більший за індекс рядка елемента. Потім дізнаємось найбільшу суму. Результатом буде виведення матриці(з найбільшою сумою) , у якої будуть замінені від’ємні елементи на цю ж значення.

**2.Побудова математичної моделі**.

Складемо таблицю імен змінних.

Головна програма:

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Ім’я матриці | Дійсний | А | Початкове дане |
| Ім’я матриці | Дійсний | В | Початкове дане |
| Ім’я матриці | Дійсний | С | Початкове дане |
| Розмір матриці | Цілий | n | Початкове дане |
| Змінна в циклі | Цілий | і | Проміжне дане |
| Сума | Дійсний | sumA | Проміжне дане |
| Сума | Дійсний | sumВ | Проміжне дане |
| Сума | Дійсний | sumС | Проміжне дане |
| Максимальна сума | Дійсний | max | Результат |

Підпрограма(fillMatrix):

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Матриця | Дійсний | matrix | Аргумент функції |
| Розмір матриці | Цілий | size | Аргумент функції |
| Змінна у циклі | Цілий | і | Проміжне дане |
| Змінна у циклі | Цілий | j | Проміжне дане |

Підпрограма(printMatrix):

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Матриця | Дійсний | matrix | Аргумент функції |
| Розмір матриці | Цілий | size | Аргумент функції |
| Змінна у циклі | Цілий | і | Проміжне дане |
| Змінна у циклі | Цілий | j | Проміжне дане |

Підпрограма(sum):

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Матриця | Дійсний | matrix | Аргумент функції |
| Розмір матриці | Цілий | size | Аргумент функції |
| Сума | Дійсний | sum | Проміжне дане |
| Змінна у циклі | Цілий | і | Проміжне дане |
| Змінна у циклі | Цілий | j | Проміжне дане |

Підпрограма(findMax):

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Сума | Дійсний | a | Аргумент функції |
| Сума | Дійсний | b | Аргумент функції |

Підпрограма(replaceAllNegatives):

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Матриця | Дійсний | matrix | Аргумент функції |
| Розмір матриці | Цілий | size | Аргумент функції |
| Значення ,на яке замінятимуться від’ємні числа | Дійсний | value | Аргумент функції |
| Змінна у циклі | Цілий | і | Проміжне дане |
| Змінна у циклі | Цілий | j | Проміжне дане |

3.Розв’язок

Крок 1. Визначимо основні дії головної програми.

Крок 2. Деталізуємо дію заповнення матриць.

Крок 3. Деталізуємо дію виводу матриць.

Крок 4. Деталізуємо дію знаходження сум елементів матриць, що знаходяться над головною діагоналлю.

Крок 5. Деталізуємо дію знаходження максимальної суми.

Крок 6. Деталізуємо дію заміни від’ємних значень.

**Псевдокод**

**Крок 1**

Головна програма:

початок

float\*\* A

float\*\* B

float\*\* C

int n

ввід n

A = new float\*[n]

B = new float\*[n]

C = new float\*[n]

для і від 0 до n з кроком 1

виконати:

A[i] = new float[n]

B[i] = new float[n]

C[i] = new float[n]

все виконати

виклик процедури fillMatrix(A,n)

виклик процедури fillMatrix(В,n)

виклик процедури fillMatrix(С,n)

виклик процедури printMatrix(A,n)

виклик процедури printMatrix(В,n)

виклик процедури printMatrix(С,n)

float sumA = виклик функції sum(A,n)

float sumB = виклик функції sum(B,n)

float sumC = виклик функції sum(C,n)

float max = виклик функції findMax(sumA, findMax(sumB,sumC))

якщо sumA == max

то

виклик процедури replaceAllNegatives(A,n,max)

виклик процедури printMatrix(A,n)

все якщо

якщо sumВ == max

то

виклик процедури replaceAllNegatives(В,n,max)

виклик процедури printMatrix(В,n)

все якщо

якщо sumС == max

то

виклик процедури replaceAllNegatives(С,n,max)

виклик процедури printMatrix(С,n)

все якщо

delete[] A, B, C

кінець

**Крок 2**

Підпрограма:

процедура fillMatrix(float\*\* matrix,int size)

для і від 0 до size з кроком 1

виконати:

для j від 0 до size з кроком 1

виконати:

ввід matrix[i][j]

все виконати

все виконати

повернути

все процедура

**Крок 3**

підпрограма:

процедура printMatrix(float\*\* matrix, int size)

для і від 0 до size з кроком 1

виконати:

для j від 0 до size з кроком 1

виконати:

вивід matrix[i][j]

все виконати

все виконати

повернути

все процедура

**Крок 4**

підпрограма:

функція sum(float\*\* matrix,int size)

float sum = 0

для і від 0 до size з кроком 1

виконати:

для j від і+1 до size з кроком 1

виконати:

sum += matrix[i][j]

все виконати

все виконати

повернути

все функція

**Крок 5**

підпрограма:

функція findMax(float a,float b)

якщо a > b

то

повернути а

все якщо

повернути b

все функція

**Крок 6**

підпрограма:

процедура replaceAllNegatives(float\*\* matrix,int size,float value)

для і від 0 до size з кроком 1

виконати:

для j від 0 до size з кроком 1

виконати:

якщо matrix[i][j]<0

то

matrix[i][j] = value

все якщо

все виконати

все виконати

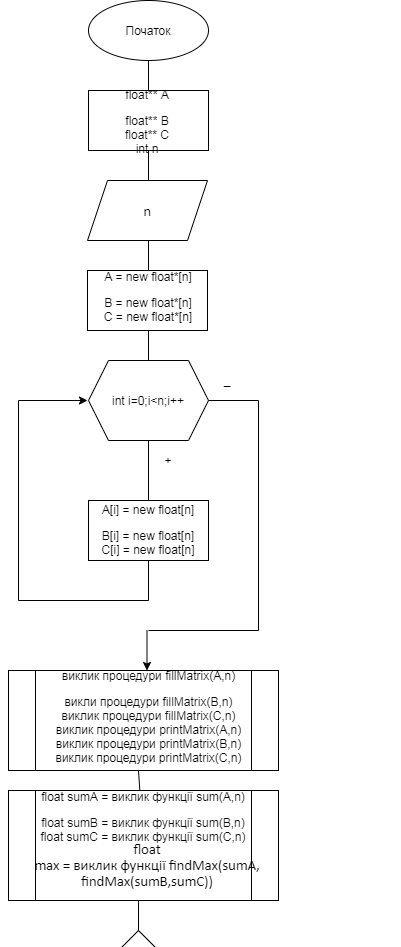
повернути

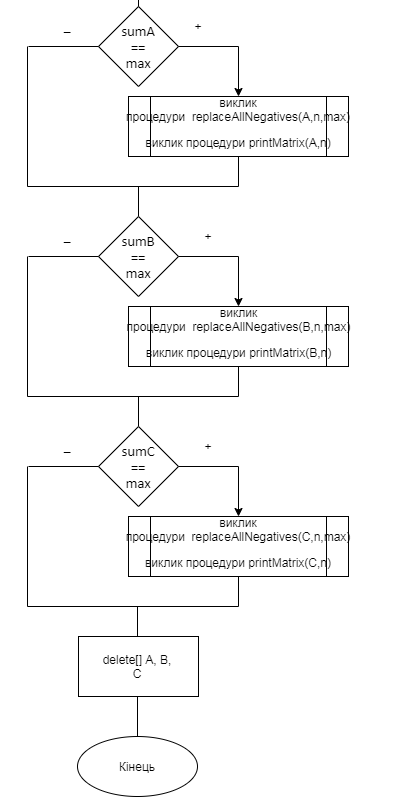
все процедура

**Блок-схема**

**Крок 1**

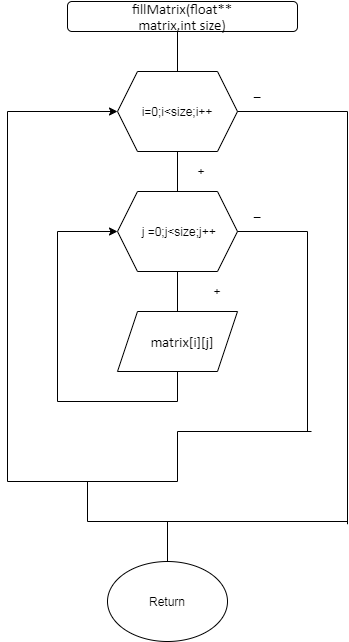
**Головна програма:**

****



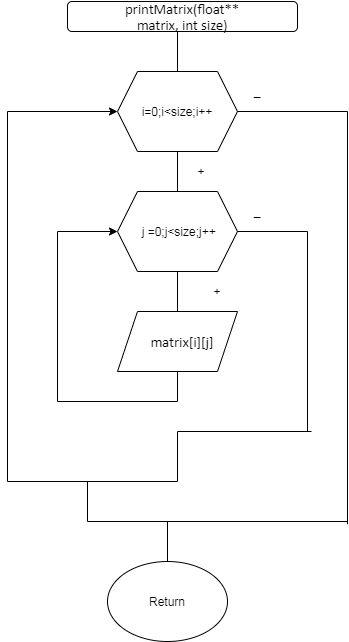
Крок 2

Підпрограма:



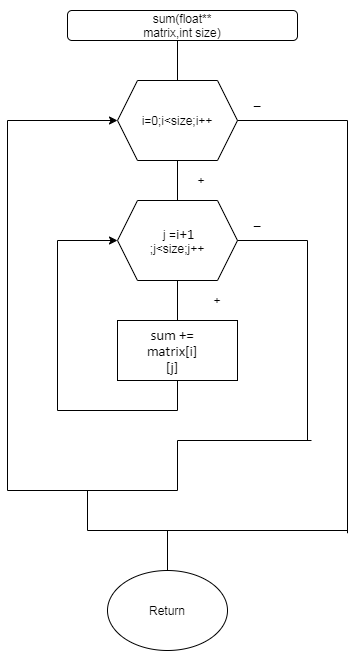
**Крок 3**

**Підпрограма:**

****

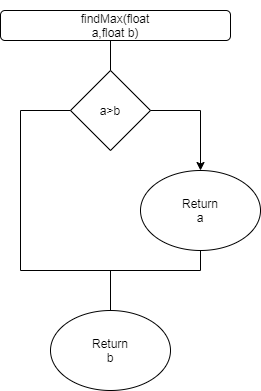
**Крок 4**

**Підпрограма:**

****

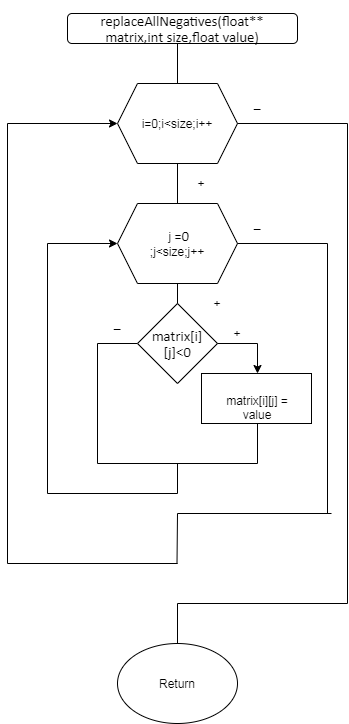
**Крок 5**

**Підпрограма:**

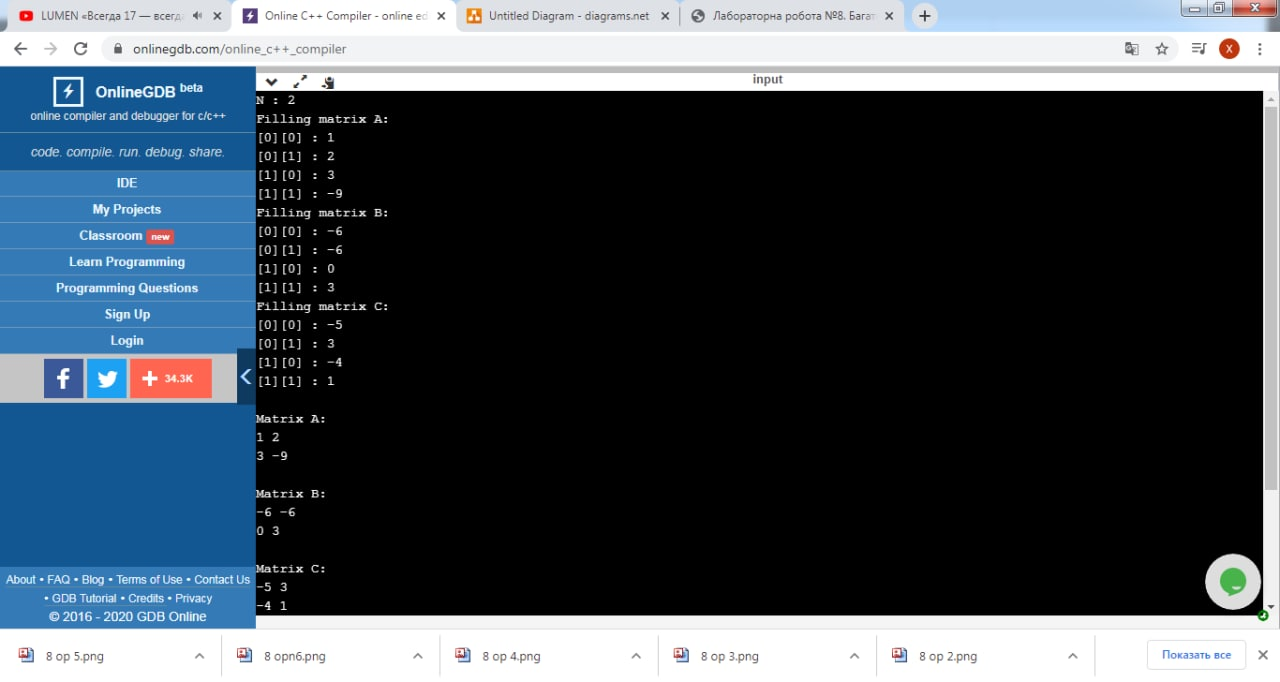
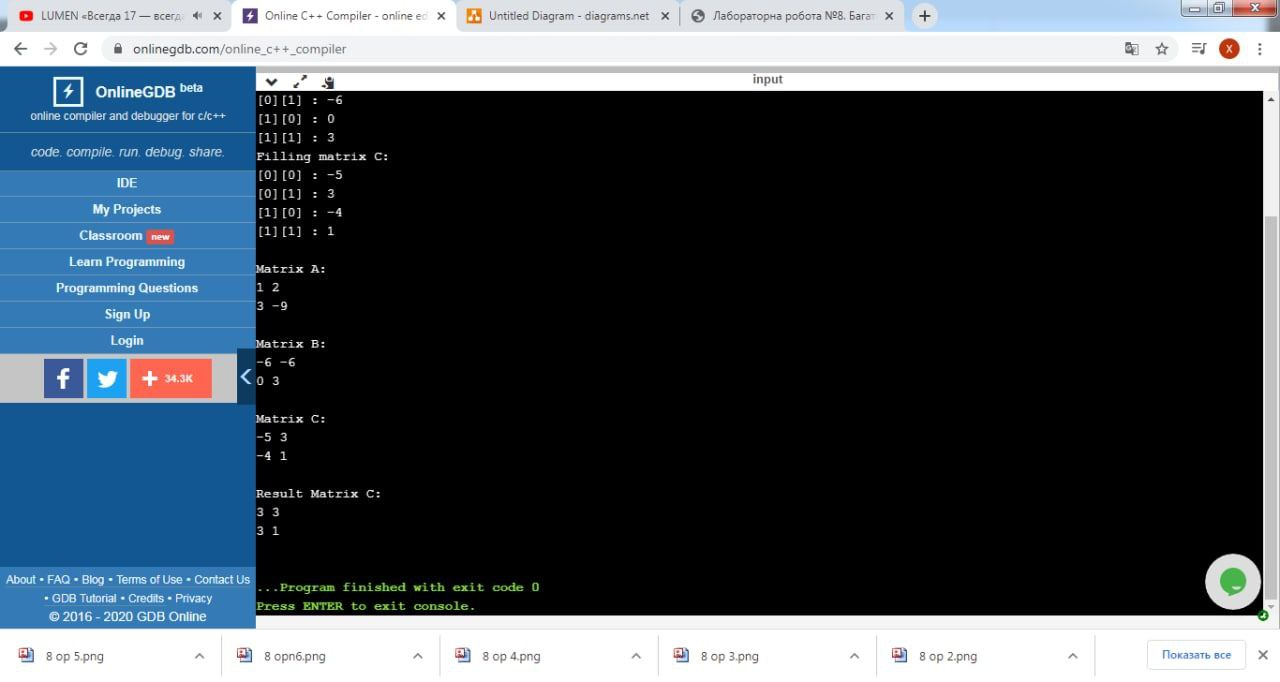
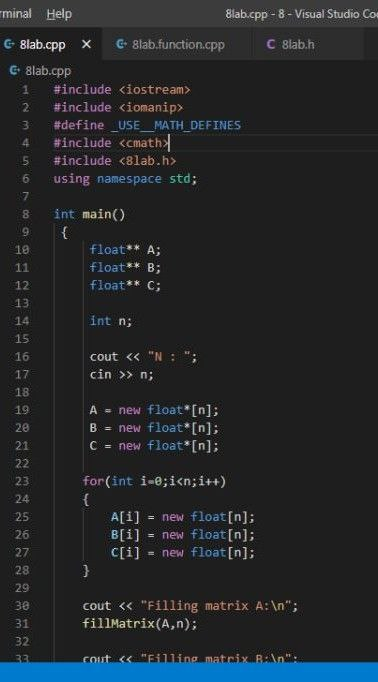
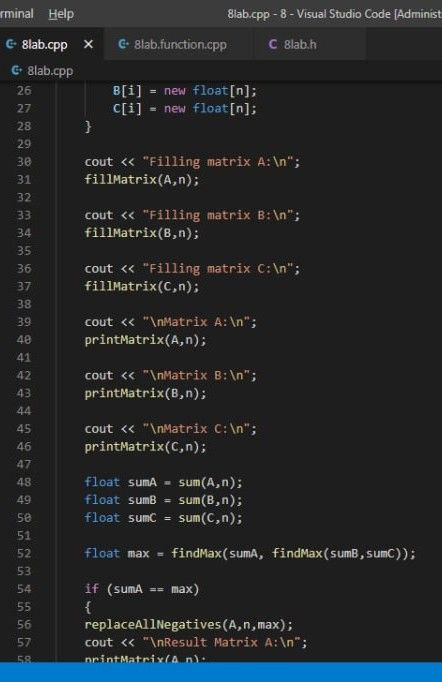
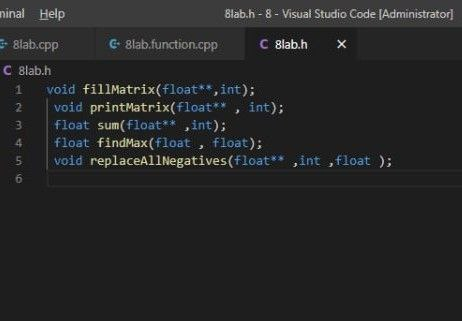
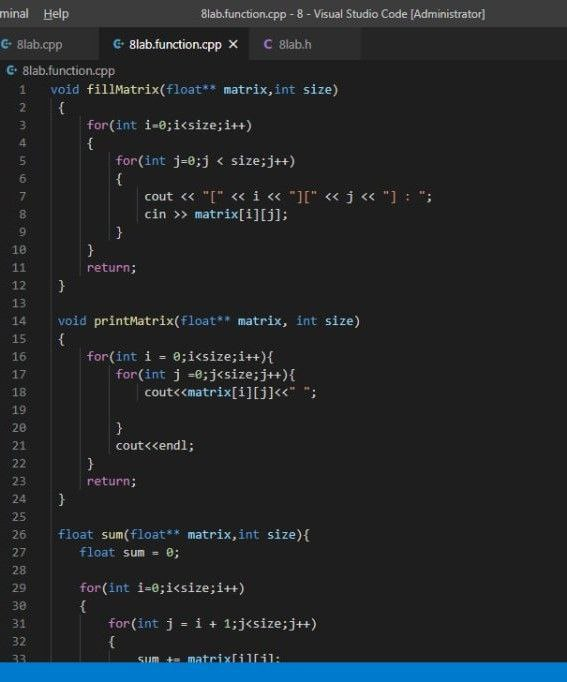
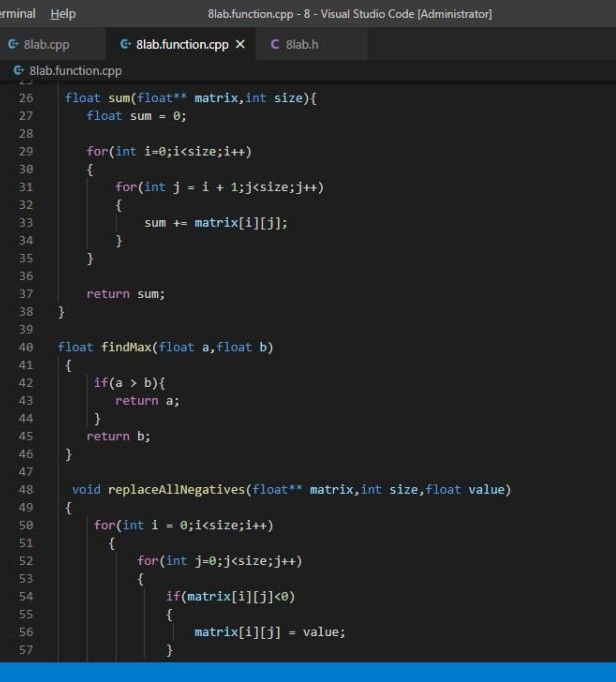
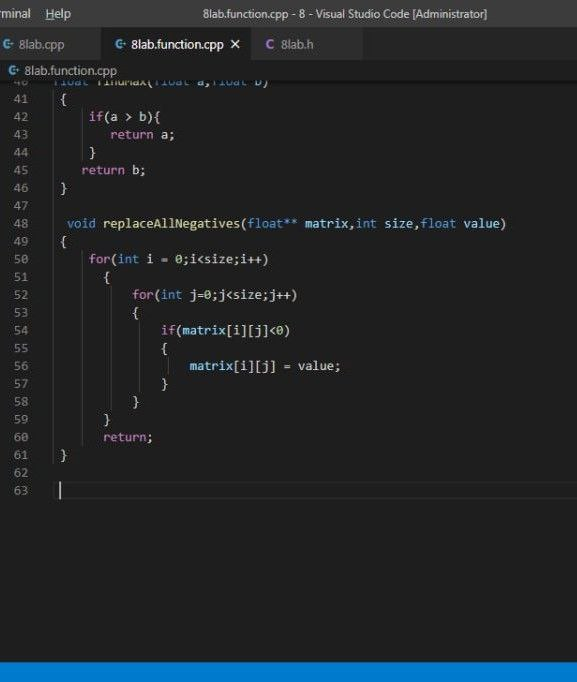
****

**Крок 6**

**Підпрограма:**

****

**С++**

**        **

Висновок:Виконавши цю лабораторну роботу, я – опанувала технологію використання двовимірних масивів даних(матриць), навчилася розробляти алгоритми та програми із застосуванням матриць.