

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Катедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 8 з дисципліни

«Основи програмування 1.

Базові конструкції»

«Багатовимірні масиви»

Варіант 18

Виконав студент ІП-11 Лесів Владислав Ігорович

Перевірив Вітковська Ірина Іванівна

Київ 2021

Лабораторна робота 8

Багатовимірні масиви

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

Варіант №18.

18. По заданій квадратній матриці A побудувати нову матрицю B тієї ж розмірності, кожен елемент b_{ij} якої одержується як середнє арифметичне сусідів елемента a_{ij} по рядкам та стовпцям (сусідами для елемента a_{ij} є елементи $a_{i-1,j}$, $a_{i+1,j}$, $a_{i,j-1}$, $a_{i,j+1}$).

Постановка задачі. Результатом розв'язку є квадратна матриця заданої розмірності, елементи якої обраховані так, як це вказано в умові задачі: як середнє арифметичне сусідів елемента такого ж індексу з початкової матриці. Для визначення результату повинне бути задане натуральне число $size$ – розмірність квадратної матриці, а також $size*size$ дійсних чисел, що утворюють матрицю. Інших початкових даних для розв'язку не потрібно.

Математичне формулювання задачі зводиться до знаходження середнього арифметичного між певною кількістю чисел. Якщо елемент має сусіда $[i-1][j]$ – додаємо його до змінної суми чисел, а також інкрементуємо значення змінної кількості сусідів. Аналогічно з наявністю сусідів $[i+1][j]$, $[i][j-1]$, $[i][j+1]$. Обрахунок здійснюватимемо за формулою $s = \frac{sum}{num}$, де sum – сума сусідів, num – їх кількість.

Виконання мовою C++.

Код програми:

```
#include <iostream>
#include <ctime>

using namespace std;

void processMatrixA(double*[], int);           //Прототипи функцій: обробка першої
void processMatrixB(double* [], double* [], int); //Обробка нової матриці
void outputMatrix(double* [], int);           //Виведення матриці
void deleteMatrix(double* [], int);           //Видалення матриці

int main()
{
    setlocale(LC_ALL, "ukr");
    int size;
    cout << "Введіть розмір матриці: ";
    cin >> size;
    double** arrA = new double*[size];
    processMatrixA(arrA, size);
    double** arrB = new double* [size];
    processMatrixB(arrA, arrB, size);
    cout << "\nВведена матриця A:\n";
    outputMatrix(arrA, size);
    cout << "\nПобудована матриця B:\n";
    outputMatrix(arrB, size);
    deleteMatrix(arrA, size);
    deleteMatrix(arrB, size);
}

void processMatrixA(double *arrA[], int size) {
    for (int i = 0; i < size; i++)
        arrA[i] = new double[size];           //Ініціалізація динамічних масивів
    srand(time(NULL));
    for (int i = 0; i < size; i++)
        for (int j = 0; j < size; j++)
            arrA[i][j] = rand() & 30 - 15;     //Заповнюємо матрицю від -15 до 15
}

void processMatrixB(double* arrA[], double* arrB[], int size) {
    for (int i = 0; i < size; i++)
        arrB[i] = new double[size];
    if (size == 1)                             //Матриця одного елементу
        arrB[0][0] = arrA[0][0];
    else {
        int num;
        double sum;
        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size; j++) {
                sum = 0;
                num = 0;
                if (i >= 1) {                     //Існує елемент a[i-1][j]
                    sum += arrA[i - 1][j];
                    num++;
                }
                if (i < size - 1) {                //Існує елемент a[i+1][j]
                    sum += arrA[i + 1][j];
                    num++;
                }
                if (j >= 1) {                     //Існує елемент a[i][j-1]
                    sum += arrA[i][j - 1];
                    num++;
                }
            }
        }
    }
}
```

```

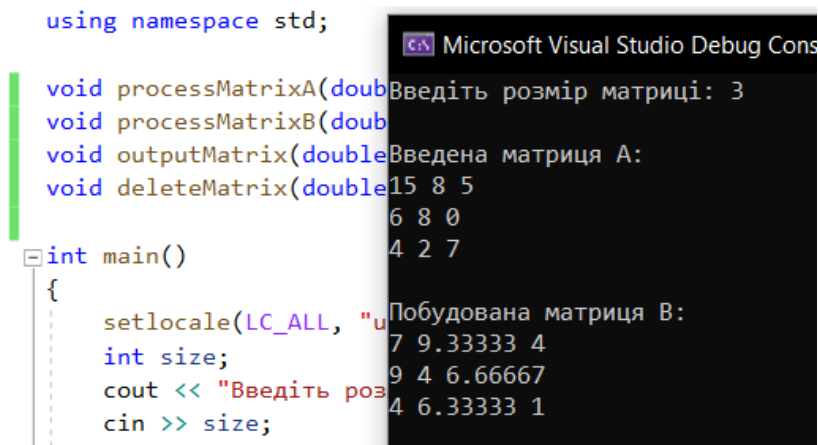
        }
        if (j < size - 1) { //Існує елемент a[i][j+1]
            sum += arrA[i][j + 1];
            num++;
        }
        arrB[i][j] = sum / num;
    }
}

void outputMatrix(double* arr[], int size) {
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }
}

void deleteMatrix(double* arr[], int size) {
    for (int i = 0; i < size; i++)
        delete[] arr[i];
    delete[] arr;
}

```

Випробування алгоритму.



```

using namespace std;

void processMatrixA(double* arrA[], int size) {
    for (int i = 0; i < size; i++)
        for (int j = 0; j < size; j++)
            arrA[i][j] = (rand() % 10) + 1;
}

void processMatrixB(double* arrB[], int size) {
    for (int i = 0; i < size; i++)
        for (int j = 0; j < size; j++)
            arrB[i][j] = (rand() % 10) + 1;
}

void outputMatrix(double* arr[], int size) {
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++)
            cout << arr[i][j] << " ";
        cout << endl;
    }
}

void deleteMatrix(double* arr[], int size) {
    for (int i = 0; i < size; i++)
        delete[] arr[i];
    delete[] arr;
}

int main() {
    setlocale(LC_ALL, "uk");
    int size;
    cout << "Введіть розмір матриці: ";
    cin >> size;

    double** arrA = new double*[size];
    processMatrixA(arrA, size);
    double** arrB = new double*[size];
    processMatrixB(arrB, size);
    outputMatrix(arrA, size);
    outputMatrix(arrB, size);
    deleteMatrix(arrA, size);
    deleteMatrix(arrB, size);
}

```

Microsoft Visual Studio Debug Console

Введіть розмір матриці: 3

Введена матриця A:

```

15 8 5
6 8 0
4 2 7

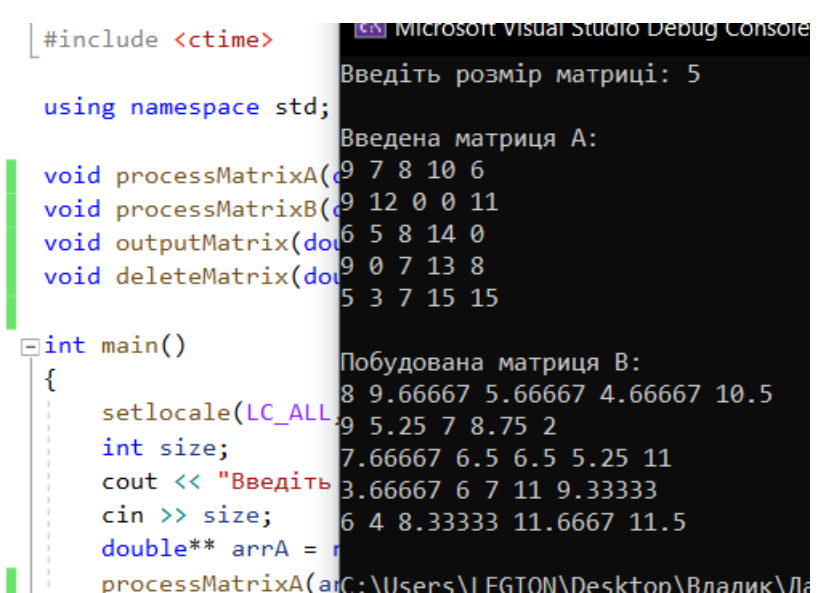
```

Побудована матриця B:

```

7 9.33333 4
9 4 6.66667
4 6.33333 1

```



```

#include <ctime>
using namespace std;

void processMatrixA(double* arrA[], int size) {
    for (int i = 0; i < size; i++)
        for (int j = 0; j < size; j++)
            arrA[i][j] = (rand() % 10) + 1;
}

void processMatrixB(double* arrB[], int size) {
    for (int i = 0; i < size; i++)
        for (int j = 0; j < size; j++)
            arrB[i][j] = (rand() % 10) + 1;
}

void outputMatrix(double* arr[], int size) {
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++)
            cout << arr[i][j] << " ";
        cout << endl;
    }
}

void deleteMatrix(double* arr[], int size) {
    for (int i = 0; i < size; i++)
        delete[] arr[i];
    delete[] arr;
}

int main() {
    setlocale(LC_ALL, "uk");
    int size;
    cout << "Введіть розмір матриці: ";
    cin >> size;

    double** arrA = new double*[size];
    processMatrixA(arrA, size);
    double** arrB = new double*[size];
    processMatrixB(arrB, size);
    outputMatrix(arrA, size);
    outputMatrix(arrB, size);
    deleteMatrix(arrA, size);
    deleteMatrix(arrB, size);
}

```

Microsoft Visual Studio Debug Console

Введіть розмір матриці: 5

Введена матриця A:

```

9 7 8 10 6
9 12 0 0 11
6 5 8 14 0
9 0 7 13 8
5 3 7 15 15

```

Побудована матриця B:

```

8 9.66667 5.66667 4.66667 10.5
9 5.25 7 8.75 2
7.66667 6.5 6.5 5.25 11
3.66667 6 7 11 9.33333
6 4 8.33333 11.6667 11.5

```

Висновок. Отже, у цій роботі я опанував технологію використання двовимірних масивів даних (матриць), навчився розробляти алгоритми та програми із застосуванням матриць. У результаті лабораторної роботи було розроблено програму, яка виконує задачу відповідно до постановки. Використовуючи функції для введення матриці випадковим чином, обчислення елементів нової матриці, виведення матриць та їхнього видалення, отримуємо коректний результат.