**Міністерство Освіти і Науки України**

**Київський Національний Університет імені Тараса Шевченка**

**Факультет Інформаційних Технологій**

Звіт з лабораторної роботи № 8

з дисципліни **Основи програмування**

Виконав студент групи ІР-12

Федоренко Ярослав Юрійович

Викладач: ас. Пороховніченко І.А.

**Київ – 2023**

**Лабораторна робота №8**

**Варіант № 18**

**Тема:** Двовимірні масиви.

**Завдання № 1**



**Блок – схема**









**­Код**

**main.cpp**

#include <iostream>

#include "functions.h"

using namespace std;

int main()

{

char q = 'y';

do {

int m = 0;

int \*\* A = nullptr;

int \*\* B = nullptr;

cout << "Введіть розмірність матриць: ";

cin >> m;

A = initMatrix(m, m);

B = initMatrix(m, m);

int method = 0;

cout << "Введення елементів матриці з клавіатури чи генерація випадкових чисел? (1/2): ";

cin >> method;

switch (method) {

case 1:

cout << endl;

inputMatrix(A, m, m);

cout << endl;

inputMatrix(B, m, m);

break;

case 2:

randomMatrix(A, m, m);

randomMatrix(B, m, m);

break;

default:

cout << endl << "Невірний вибір, до побачення!" << endl;

return 1;

}

int \*\* C = initMatrix(m, m);

calculateC(A, B, C, m, m);

cout << endl << "Матриця A:" << endl;

printMatrix(A, m, m);

cout << endl << "Матриця B:" << endl;

printMatrix(B, m, m);

cout << endl << "Матриця C:" << endl;

printMatrix(C, m, m);

cout << endl << "Індекси впорядкованих рядків третьої матриці: ";

bool anyOrderedRows = printArrangedRowsIndexes(C, m, m);

if (!anyOrderedRows) {

cout << "відсутні.";

}

deleteMatrix(A, m, m);

deleteMatrix(B, m, m);

deleteMatrix(C, m, m);

cout << endl << endl << "Бажаєте повторити? (y/n): ";

cin >> q;

if (q != 'y')

cout << endl << "До побачення!";

else

cout << endl;

} while (q == 'y');

return 0;

}

**functions.cpp**

#include "functions.h"

void inputMatrix(int \*\* matrix, int m, int n)

{

for (int i = 0; i < m; i++) {

cout << "Введіть " << n << " елементів " << i + 1 << " рядка: ";

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

cin >> matrix[i][j];

}

}

}

void randomMatrix(int \*\* matrix, int m, int n)

{

for (int i = 0; i < m; i++) {

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

matrix[i][j] = rand() % 100 + pow(-1, rand() % 2);

}

}

}

int \*\* initMatrix(int m, int n)

{

int \*\* matrix = new int\*[m];

for (int i = 0; i < m; i++) {

matrix[i] = new int[n] {};

}

return matrix;

}

void printMatrix(int \*\* matrix, int m, int n)

{

for (int i = 0; i < m; i++) {

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

cout << matrix[i][j] << " ";

}

cout << endl;

}

}

void deleteMatrix(int \*\*& matrix, int m)

{

for (int i = 0; i < m; i++) {

delete[] matrix[i];

}

delete[] matrix;

matrix = nullptr;

}

int findMaxNumber(int \* row, int length)

{

int max = row[0];

for (int i = 1; i < length; i++) {

if (row[i] > max) {

max = row[i];

}

}

return max;

}

void calculateC(int \*\* A, int \*\* B, int \*\* C, int m, int n)

{

for (int i = 0; i < m; i++) {

int max = findMaxNumber(B[i], n);

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

C[i][j] = A[i][j] \* max;

}

}

}

bool isAscending(int \* row, int length)

{

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

if (row[i] > row[i + 1]) {

return false;

}

}

return true;

}

bool isDescending(int \* row, int length)

{

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

if (row[i] < row[i + 1]) {

return false;

}

}

return true;

}

bool printArrangedRowsIndexes(int \*\* matrix, int m, int n)

{

bool hasOrderedRow = false;

for (int i = 0; i < m; ++i) {

if (

isAscending(matrix[i], n) ||

isDescending(matrix[i], n)

) {

cout << i << " ";

hasOrderedRow = true;

}

}

return hasOrderedRow;

}

**Висновок**

В цій лабораторній роботі я ознайомився з двовимірними масивами і навчився будувати алгоритми з їхнім застосуванням, наприклад, завдання з матрицями.