Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут”

Кафедра АСОІУ

ЗВІТ

про виконання лабораторної роботи №8

з дисципліни

“Основи програмування”

Тема: МАСИВИ.

БАГАТОВИМІРНІ МАСИВИ

Прийняв: Виконав:

студент 1-го курсу

гр. ІП-92 ФІОТ

Залізчук Данило

Дмитрович

Київ 2019

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

**Завдання 10:**

Задані три цілочисельні матриці розмірності *m x m*, що містять від’ємні елементи. З’ясувати, яка з цих матриць має найменшу суму модулів діагональних елементів тих рядків, що починаються з від’ємного елемента. Упорядкувати рядки визначеної матриці за збільшенням значень її елементів.

**Блок-схема:**





**Код:**

#include <iostream>

#include <iomanip>

#include <ctime>

using namespace std;

void arrayInit(int\*\*, int); // Создание матрицы

void arrayOutput(int\*\*, int); // Вывод матрицы

int arraysMinSum(int\*\*, int\*\*, int\*\*, int); // Нахождение минимальной суммы модулей диагонали матрицы

void arraySort(int\*\*, int); // Сортировка строк матрицы

void addSecondDimension(int\*\*, int); //Добавление второго измерения массиву

int main() {

setlocale(LC\_CTYPE, "rus");

srand(time(NULL));

int m; // Размер матрицы

cout << "Введите порядок матриц: ";

cin >> m;

int\*\* arr1 = new int\* [m];

int\*\* arr2 = new int\* [m]; //Обьявление динамических двумерных массивов

int\*\* arr3 = new int\* [m];

addSecondDimension(arr1, m);

addSecondDimension(arr2, m); //Добавляем ещё одно измерение

addSecondDimension(arr3, m);

arrayInit(arr1, m);

arrayInit(arr2, m);

arrayInit(arr3, m);

arrayOutput(arr1, m);

arrayOutput(arr2, m);

arrayOutput(arr3, m);

int sum = arraysMinSum(arr1, arr2, arr3, m); // Номер матрицы с минимальной суммой модулей диагонали

if (sum == 1) {

cout << "Матрица 1 имеет наименьшую сумму модулей по диагонали." << endl;

cout << "Отсортированные строки матрицы 1:" << endl << endl;

arraySort(arr1, m);

arrayOutput(arr1, m);

}

else

if (sum == 2) {

cout << "Матрица 2 имеет наименьшую сумму модулей по диагонали." << endl;

cout << "Отсортированные строки матрицы 2:" << endl << endl;

arraySort(arr2, m);

arrayOutput(arr2, m);

}

else

if (sum == 3) {

cout << "Матрица 3 имеет наименьшую сумму модулей по диагонали." << endl;

cout << "Отсортированные строки матрицы 3:" << endl << endl;

arraySort(arr3, m);

arrayOutput(arr3, m);

}

return 0;

}

//================== Создание матрицы ==================

void arrayInit(int\*\* p, int size) {

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

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

p[i][j] = rand() % 60 - 60;

}

//======================================================

//================== Вывод матрицы ==================

void arrayOutput(int\*\* p, int size) {

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

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

cout << setw(4) << p[i][j] << ' ';

cout << endl;

}

cout << endl;

}

//===================================================

//================== Нахождение минимальной суммы модулей диагонали матрицы ==================

int arraysMinSum(int\*\* p1, int\*\* p2, int\*\* p3, int size) {

int sum11 = 0, sum12 = 0, sum21 = 0, sum22 = 0, sum31 = 0, sum32 = 0;

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

sum11 += abs(p1[i][i]);

sum21 += abs(p2[i][i]);

sum31 += abs(p3[i][i]);

}

int j = 0;

for (int i = size - 1; i >= 0; i--) {

sum12 += abs(p1[i][j]);

sum22 += abs(p2[i][j]);

sum32 += abs(p3[i][j]);

j++;

}

if (sum11 > sum12)

sum11 = sum12;

if (sum21 > sum22)

sum21 = sum22;

if (sum31 > sum32)

sum31 = sum32;

if (sum11 < sum21 && sum11 < sum31)

return 1;

else

if (sum21 < sum11 && sum21 < sum31)

return 2;

else

if (sum31 < sum21 && sum31 < sum11)

return 3;

}

//==========================================================================================

//================== Сортировка строк матрицы ==================

void arraySort(int\*\* p, int size) {

int temp;

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

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

for (int k = 0; k < size - j - 1; k++) {

if (p[i][k] > p[i][k + 1]) {

temp = p[i][k];

p[i][k] = p[i][k + 1];

p[i][k + 1] = temp;

}

}

}

}

//======================= Добавление измерения массива =======================================

void addSecondDimension(int\*\* arr, int arrLength) {

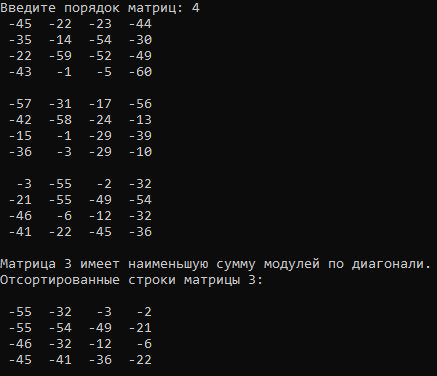
for (int i = 0; i < arrLength; i++)

arr[i] = new int[arrLength];

}

//==============================================================

**Результат:**



**Висновок:**

Як підсумок, хочу сказати, що після виконання даної лабораторної роботи я отримав навички написання програм для обробки багатовимірних масивів.