# Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №8 з дисципліни

«Основи програмування-1.

Базові конструкції»

«Багатовимірні масиви»

Варіант 29

Виконав студент ІП-11 Тарасьонок Дмитро Євгенович

(шифр, прізвище, ім'я, по батькові)

Перевірив Вітковська Ірина Іванівна

( прізвище, ім'я, по батькові)

Київ 2021

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

Умова задачі: На основі заданої матриці B (n×n) побудувати вектор X(n), елементи якого xi – скалярний добуток i-го рядка матриці A на стовпець, що містить найменший елемент в цьому рядку

Математична модель:

Для вирішення даної задачі необхідно спочатку ввести значення n, що відповідає розміру матриці, після чого в підпрограмі розмістити масив у динамічній пам’яті, заповнити його випадковими значеннями, потім треба розробити підпрограму, що міститеме в собі арифметичний цикл, що проходитиме по всіх рядках матриці, шукаючи в кожному стовпчик із мінімальним елементом. Коли стовпець знайдено, обраховується скалярний добуток рядка на цей стовпель з використанням арифметичного, в якому кожен елемент рядка множитиметься на відповідний йому елемент стовпця, що міститеме найменший елемент у цьому рядку.

Блок-схема:



Програма мовою C++:

#include <iostream>

#include <locale>

#include <limits>

#include <iomanip>

using namespace std;

double\*\* input\_matrix(int);

void output\_matrix(double\*\*, int);

void output\_array(double\*, int);

double\* process\_matrix(double\*\*, int);

int main()

{

locale::global(locale("rus"));

srand(time(NULL));

int n;

cout << "Введiть розмiр матрицi: ";

cin >> n;

double\*\* B = input\_matrix(n);

cout << "Згенерована матриця:" << endl;

output\_matrix(B, n);

double\* X = process\_matrix(B, n);

cout << "Скалярний добуток i-го рядка матрицi на стовпцi, що мiстять найменшi елементи радкiв:" << endl;

output\_array(X, n);

system("pause");

return 0;

}

double\*\* input\_matrix(int size)

{

double\*\* B = new double\* [size];

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

{

B[i] = new double[size];

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

{

B[i][j] = (RAND\_MAX / 2 - rand()) / 100.f;

}

}

return B;

}

void output\_matrix(double\*\* matrix, int size)

{

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

{

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

{

cout << setw(8) << fixed << setprecision(2) << matrix[i][j];

}

cout << endl;

}

cout << endl;

}

void output\_array(double\* arr, int size)

{

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

{

cout << setw(10) << fixed << setprecision(2) << arr[i];

}

cout << endl << endl;

}

double\* process\_matrix(double\*\* A, int size)

{

double\* X = new double[size];

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

{

double min\_element = A[i][0];

int min\_element\_column = 0;

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

{

if (A[i][j] < min\_element)

{

min\_element = A[i][j];

min\_element\_column = j;

}

}

X[i] = 0;

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

{

X[i] += A[i][o] \* A[o][min\_element\_column];

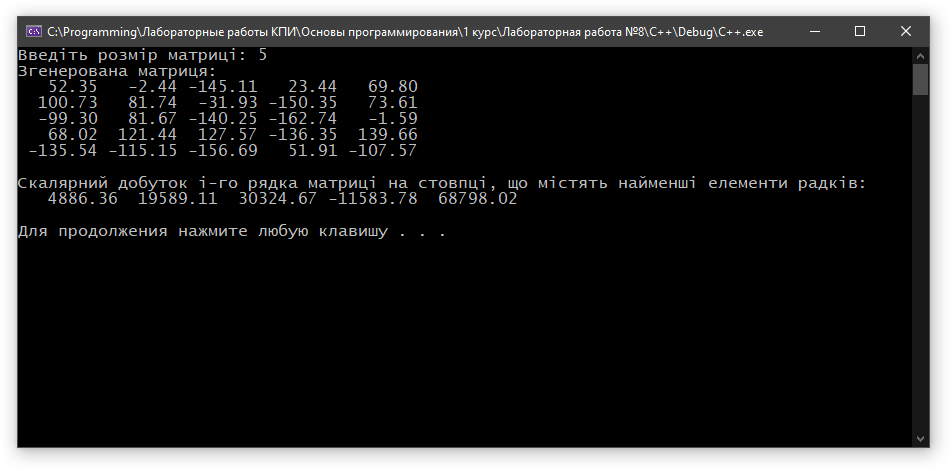
}

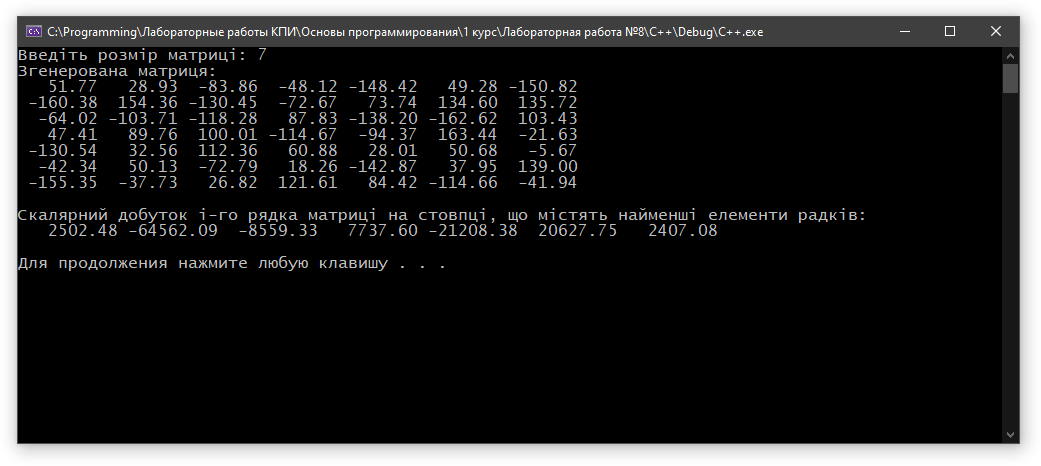
}

return X;

}

Виконання коду мовою C++:





Висновок:

Під час виконання даної лабораторної роботи я набув практичних навичок з обробки багатовимірних масивів. Я освоїв специфіку виділення багатовимірних масивів у динамічній пам’яті, що певним чином відрізняється від виділення в ній же одновимірних масивів, опанував алгоритм знаходження мінімального значення для кожного рядка багатовимірного масиву (матриці), навчився створювати одновимірний масив, проводячи певні операції над багатовимірним масивом на прикладі обчислення скалярного добутку в моїй конкретній задачі