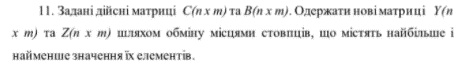
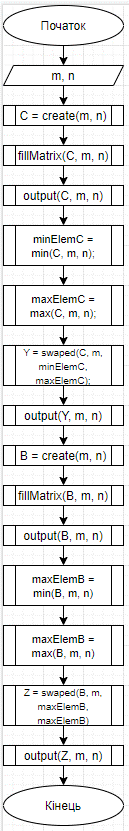
Лабораторна робота з ОП №8

Варіант №11

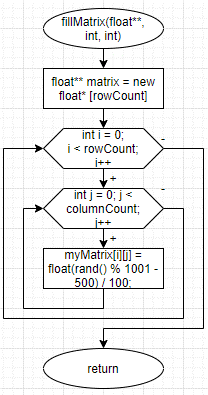
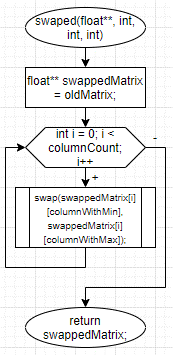
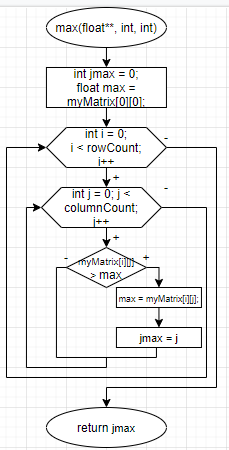
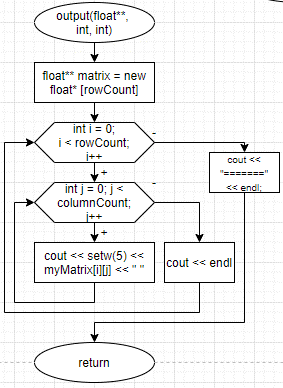
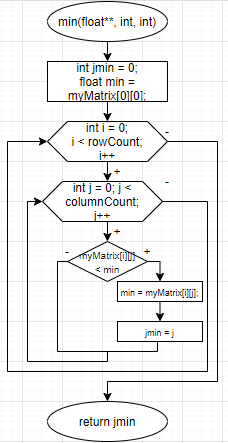
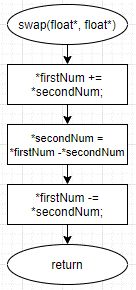
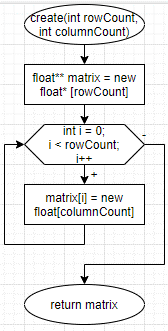
Студента групи ІП-02

Гончара Олексія

Умова:



Блок-схема:

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <iomanip>

using namespace std;

float\*\* create(int, int);

void fillMatrix(float\*\*, int, int);

void output(float\*\*, int, int);

int min(float\*\*, int, int);

int max(float\*\*, int, int);

void swap(float\*, float\*);

float\*\* swaped(float\*\*, int, int, int);

int main()

{

srand(time(0));

float\*\* C;

float\*\* B;

float\*\* Y;

float\*\* Z;

int m, n, minElemC, maxElemC, minElemB, maxElemB;

cout << "Input m...\n";

cin >> m;

cout << "Input n...\n";

cin >> n;

cout << "=============================" << endl;

cout << "First matrix:\n";

C = create(m, n);

fillMatrix(C, m, n);

output(C, m, n);

minElemC = min(C, m, n);

maxElemC = max(C, m, n);

cout << "The minimal element of the first matrix is in column #" << minElemC + 1 << endl;

cout << "The maximal element of the first matrix is in column #" << maxElemC + 1 << endl;

cout << "=============================" << endl;

Y = swaped(C, m, minElemC, maxElemC);

cout << "Swapped first matrix:\n";

output(Y, m, n);

cout << "Second matrix:\n";

B = create(m, n);

fillMatrix(B, m, n);

output(B, m, n);

minElemB = min(B, m, n);

maxElemB = max(B, m, n);

cout << "The minimal element of the first matrix is in column #" << minElemB + 1 << endl;

cout << "The maximal element of the first matrix is in column #" << maxElemB + 1 << endl;

cout << "=============================" << endl;

Z = swaped(B, m, minElemB, maxElemB);

cout << "Swapped second matrix:\n";

output(Z, m, n);

system("pause");

}

float\*\* create(int rowCount, int columnCount) {

float\*\* matrix = new float\* [rowCount];

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

matrix[i] = new float[columnCount];

}

return matrix;

}

void fillMatrix(float\*\* myMatrix, int rowCount, int columnCount) {

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

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

myMatrix[i][j] = float(rand() % 1001 - 500) / 100;

}

}

}

void output(float\*\* myMatrix, int rowCount, int columnCount) {

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

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

cout << setw(5) << myMatrix[i][j] << " ";

}

cout << endl;

}

cout << "=============================" << endl;

}

int min(float\*\* myMatrix, int rowCount, int columnCount) {

int jmin = 0;

float min = myMatrix[0][0];

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

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

if (myMatrix[i][j] < min) {

min = myMatrix[i][j];

jmin = j;

}

}

}

return jmin;

}

int max(float\*\* myMatrix, int rowCount, int columnCount) {

int jmax = 0;

float max = myMatrix[0][0];

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

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

if (myMatrix[i][j] > max) {

max = myMatrix[i][j];

jmax = j;

}

}

}

return jmax;

}

void swap(float\* firstNum, float\* secondNum) {

\*firstNum += \*secondNum;

\*secondNum = \*firstNum - \*secondNum;

\*firstNum -= \*secondNum;

}

float\*\* swaped(float\*\* oldMatrix, int columnCount, int columnWithMin, int columnWithMax) {

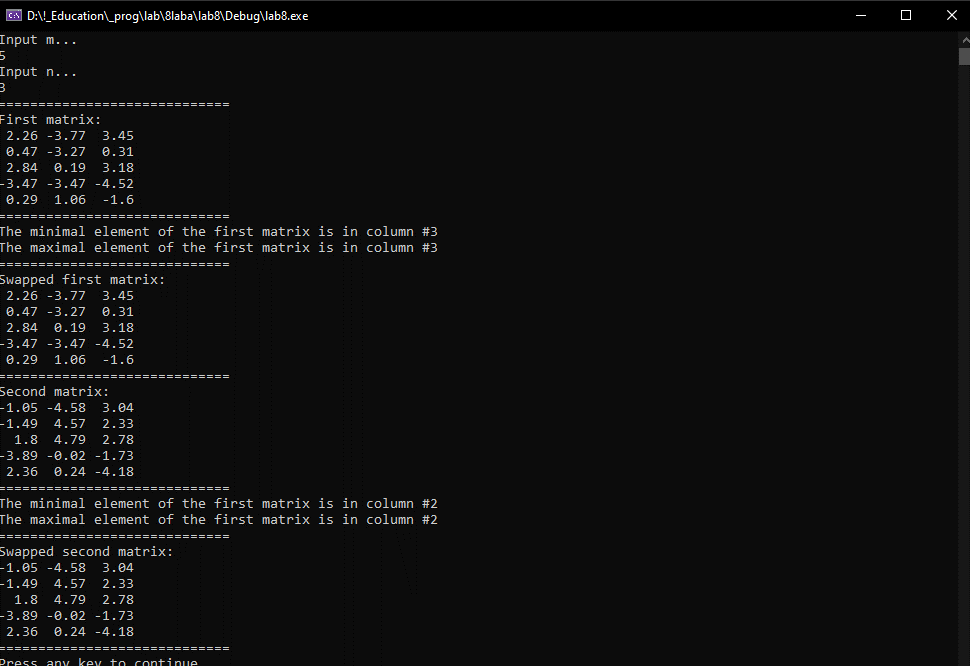
float\*\* swappedMatrix = oldMatrix;

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

swap(swappedMatrix[i][columnWithMin], swappedMatrix[i][columnWithMax]);

}

return swappedMatrix;

} 

Отже, я опанував технологію використання двомірних масивів та навчився розробляти програми з їх використанням. Я розробив програму, яка генерує 2 матриці із заданим розміром n на m. Наступна задача – створити 2 нові матриці, які відрізняються від 2 згенерованих тим, що стовпці із мінімальним та максимальним елементом змінені місцями. Для цього було створено 7 функцій: create(int, int) – створює матрицю із заданими розмірами; fillMatrix(float\*\*, int, int) – заповнює матрицю випадковими елементами; output – виводить матрицю у консоль; min(float\*\*, int, int) – шукає мінімальний елемент матриці; max(float\*\*, int, int) шукає максимальний елемент матриці; swap(float\*, float\*) – змінює місцями 2 дійсних числа; swaped(float\*\*, int, int, int) – змінює стовпці матриці згідно з умовою.