Національний технічний університет України

«Київський політехнічний інститут»

Кафедра автоматизованих систем обробки інформації та управління

Звіт про виконання лабораторної роботи №8 з дисципліни

«Алгоритмизація та програмування»

Перевірено: Ковалюк Т.В. Виконав ст. 1 курсу ФОІТ

Проскура С.Л. гр. ІС-52

Дорошенко А.В.

Київ 2015

## Лабораторна робота 8

# **Багатовимірні масиви**

Варіант № 9

*Мета:*

1. Ознайомитися з особливостями масивів;
2. Опанувати технологію застосування масивів даних;
3. Навчитися розробляти алгоритми та програми із застосуванням багатовимірних масивів.

### Завдання

|  |  |
| --- | --- |
| Задана дійсна матриця вимірності m\*n. Упорядкувати її рядки за неспадністю суми елементів рядків. Вивести масив сум елементів рядків та матрицю. |  |

### Блок-схема алгоритму



Рис.1 (Алгоритм генерації матриці)



Рис.2 (Алгоритм перестановки рядків)



Рис.3 (Алгоритм виведення масиву)



Рис.4 (Головний алгоритм)



Рис.5(Алгоритм виведення матриці)



Рис.6(Алгоритм знаходження сум елементів рядків)

### Код програми

//Laboratory work

//written by student of the first curse

//of the group IC-52

//Anton Doroshenko

//2015.11.29

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

#include <iostream>

#include <time.h>

using namespace std;

int m = 0;//кількість стовчиків

int n = 0;//кількість рядків

int array[10][10];//матриця

int ArraySum[10];//масив сум елементів рядків

//====== Функція генерації матриці ======

void input()

{

srand((unsigned)time(NULL));

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

{

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

{

array[i][j] = rand() % 50;

}

}

}

//====== Функція виведення матриці ======

void output()

{

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

{

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

{

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

}

cout << endl;

}

}

//====== Функція знаходження сум елементів рядків ======

void SearchSum()

{

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

{

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

{

ArraySum[i] += array[i][j];

}

}

}

//====== Функція виведення масиву сум елементів рядків ======

void OutputSum()

{

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

{

cout << ArraySum[i] << " ";

}

cout << endl;

}

//====== Функція перестановки ======

void replacement()

{

int temp = 0;

for (int i = 0; i < m - 1; i++)//перебір усіх елементів рядка

{

int min = ArraySum[i];//мінімальний елемент в масиві сум

for (int j = i + 1; j < m; j++)

{//знаходження мінімального елемента

if (ArraySum[j] < min)

{

min = ArraySum[j];

ArraySum[j] = ArraySum[i];

ArraySum[i] = min;

for (int k = 0; k < n; k++)//перестановка рядків

{

temp = array[j][k];

array[j][k] = array[i][k];

array[i][k] = temp;

}

}

}

}

}

//====== Головна функція ======

int main()

{

cout << "laboratory work number 8 made by Anton Doroshenko, IS-52" << endl;

cout << "enter n, m" << endl;

cin >> m >> n;

cout << "enter your elements" << endl;

input();//генерація масиву

cout << "genereted array" << endl;

output();//виведення згенерованого масиву

SearchSum();

cout << "sum of rows" << endl;

OutputSum();

replacement();

cout << "new array" << endl;

output();

cout << "new sum of rows" << endl;

OutputSum();

system("pause");

}

1. **Screen Shot результатів**

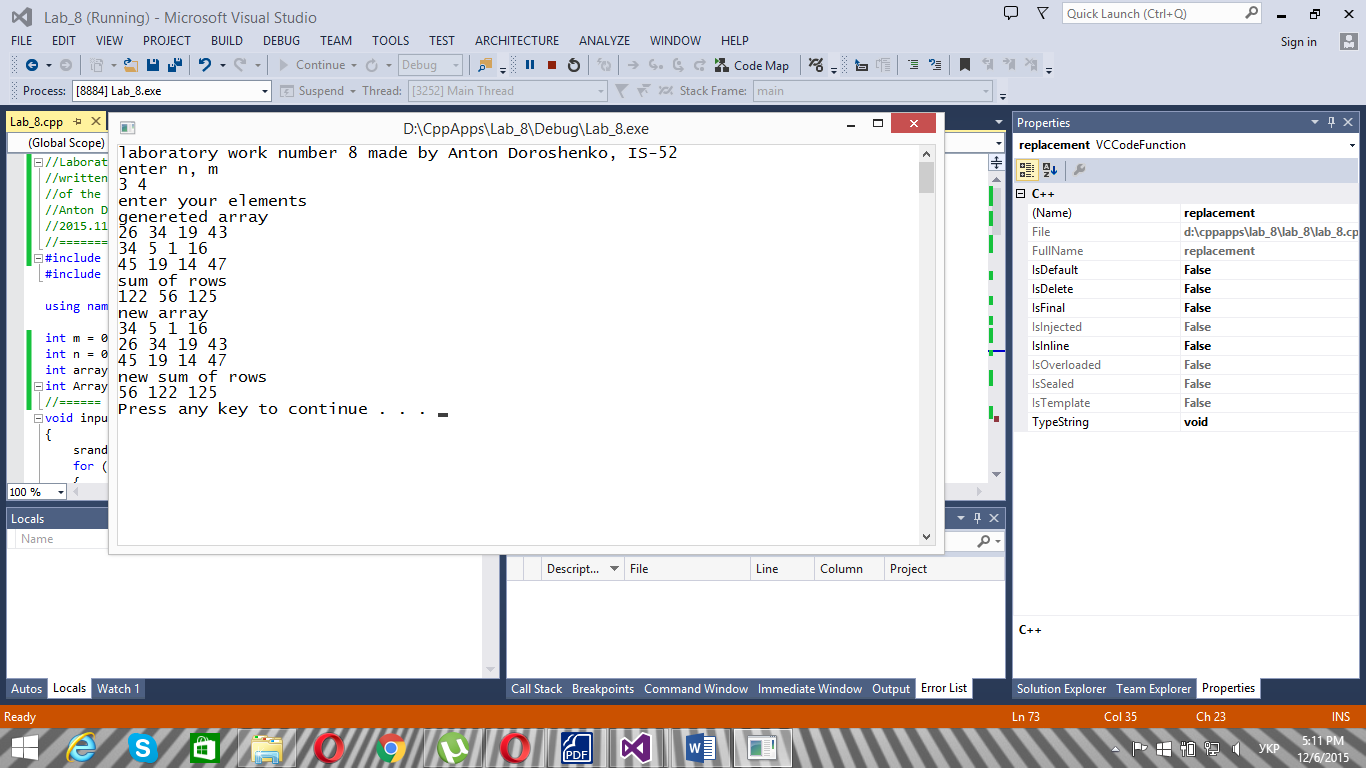


Рис.7 (Screen Shot результатів 1)

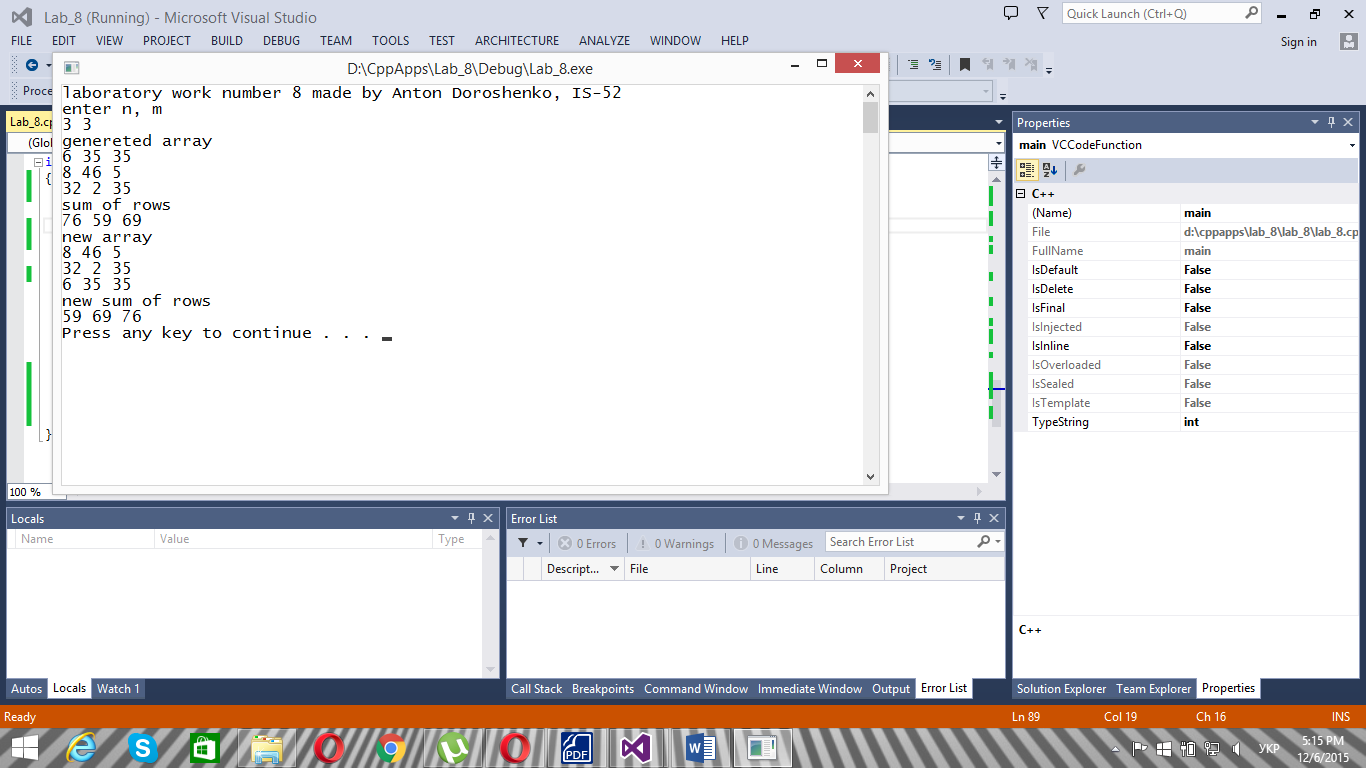


Рис.8 (Screen Shot результатів 2)

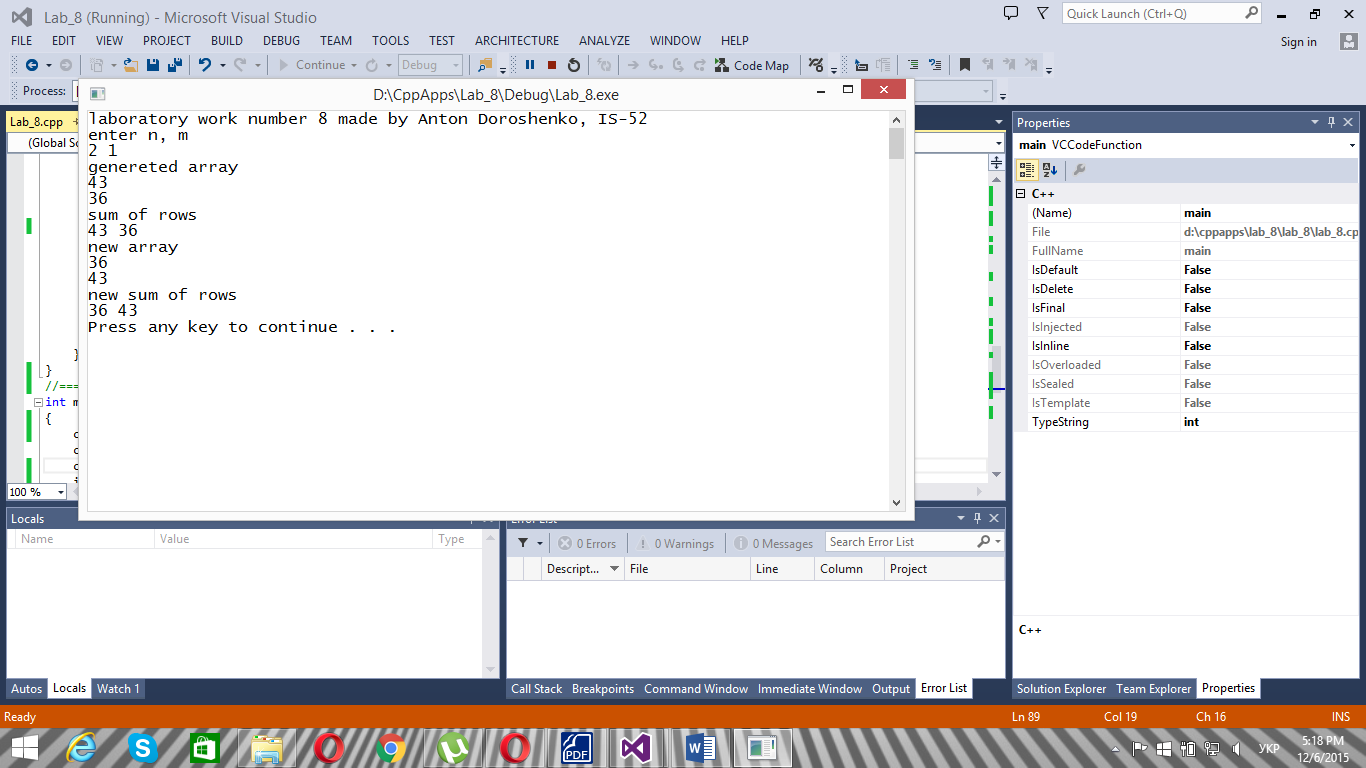


Рис.9 (Screen Shot результатів 3)

1. **Аналіз результатів**

Як ми бачимо зі ScreenShot’ів, ми ввели 3 варіанти даних.

Для різних розмірів матриці програма упорядковує її за не спадністю сум елементів її рядків.

1. **Висновок**

Програма працює правильно, про що свідчить аналіз результатів та ScreenShot’и. Програма коректно виводить результати.