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

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження алгоритмів пошуку та сортування»

Варіант 21

Виконав студент ІП-11, Ляля Іван Олександрович

Перевірила Мартинова Оксана Петрівна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Київ 2021

**Лабораторна робота 8**

**Дослідження алгоритмів пошуку та сортування**

Мета – дослідити алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Задача:

Розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом .

2. Ініціювання змінної, що описана в п.1 даного завдання.

3. Створення нової змінної індексованого типу (одновимірний масив) та її ініціювання значеннями, що обчислюються згідно з варіантом.

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  **варіанта** | **Розмірність** | **Тип даних** | **Обчислення значень елементів одновимірного масиву** |
| 21 | 8х5 | Цілий | Із добутку додатних значень елементів рядків двовимірного масиву. Відсортувати обміном за зростанням. |

Постановка задачі:

Оголошуємо цілочисельну матрицю array1 потрібного нам розміру 8х5. Далі перебираємо всі її елементи арифметичним циклом і присвоюємо кожному з них ціле значення від -99 до 99. Оголошуємо одновимірний масив, в який записуватимемо добутки додатних елементів по рядках. Оскільки всі елементи – цілі числа, то і їх добутки будуть цілими числами, а оскільки рядків 8, то в другому масиві буде 8 елементів. Оголошуємо цілочисельний масив array2 розміру 8. Далі, перебираємо кожен рядок арифметичним циклом, для кожного з рядків виконуємо наступні дії: ініціалізуємо добуток додатних елементів цього рядка product як одиницю і перебираємо знову ж арифметичним циклом всі елементи цього рядка, перевіряючи виконання умови: якщо поточний елемент більший нуля, то домножити product на цей елемент. В кінці перебору елементів i-го рядка присвоюємо i-му елементу одновимірного масиву значення product. При переході до наступного рядка відповідно добуток знову набуває значення 1, а i збільшується на одиницю. Далі переходимо до виконання сортування обміном за зростанням. Кількість проходів дорівнює 8-1=7. В кожному проході від 1-го до 7-го елемента виконуємо: якщо поточний елемент більший за наступний, то міняємо їх місцями, використовуючи додаткову змінну-пам’ять memory. Залишається вивести результат – для всіх і від 1 до 8 вивести i-тий елемент масиву array2.

Математична модель:

|  |  |  |
| --- | --- | --- |
| **Змінна** | **Значення** | **Тип** |
| **array1[][]** | **Двовимірний вхідний масив** | **Масив** |
| **array2[]** | **Одновимірний масив, вихідні дані** | **Масив** |
| **i, j, prohid** | **Лічильники в арифметичних циклах** | **Натуральні числа** |
| **product** | **Добуток додатних значень елементів в поточному рядку** | **Натуральне число** |
| **memory** | **Змінна пам’яті при заміні місцями двох елементів** | **Натуральне число** |

**Крок 1**. Визначаємо основні дії

**Крок 2**. Деталізуємо опис та ініціювання двовимірного масиву

**Крок 3**. Деталізуємо опис та ініціювання одновимірного масиву

**Крок 4**. Деталізуємо сортування одновимірного масиву обміном за зростанням

**Крок 5**. Деталізуємо виведення результату

***Псевдокоди з блок-схемами:***

*Крок 1*

**початок**

опис двовимірного масиву

ініціювання двовимірного масиву

опис одновимірного масиву

ініціювання одновимірного масиву

викон. сорт. за зрост. одновим. масиву

виводимо одновимірний масив

**кінець**

*Крок 2*

**початок**

**int** array1[8][5];

**для всіх** і **від** 1 **до** 8 **з кроком** 1:

**для всіх** j **від** 1 **до** 5 **з кроком** 1:

array1[i][j]=-99+rand()%199;

опис одновимірного масиву

ініціювання одновимірного масиву

викон. сорт. за зрост. одновим. масиву

виводимо одновимірний масив

**кінець**

*Крок 3*

**початок**

**int** array1[8][5];

**для всіх** і **від** 1 **до** 8 **з кроком** 1:

**для всіх** j **від** 1 **до** 5 **з кроком** 1:

array1[i][j]=-99+rand()%199;

**int** array2[8];

**для всіх** і **від** 1 **до** 8 **з кроком** 1:

**{ int** product=1;

**для всіх** j **від** 1 **до** 5 **з кроком** 1:

**{ якщо** array1[i][j]>0:

product=product\*array1[i][j]; **}**

array2[i]=product; **}**

викон. сорт. за зрост. одновим. масиву

виводимо одновимірний масив

**кінець**

*Крок 4*

**початок**

**int** array1[8][5];

**для всіх** і **від** 1 **до** 8 **з кроком** 1:

**для всіх** j **від** 1 **до** 5 **з кроком** 1:

array1[i][j]=-99+rand()%199;

**int** array2[8];

**для всіх** і **від** 1 **до** 8 **з кроком** 1:

**{ int** product=1;

**для всіх** j **від** 1 **до** 5 **з кроком** 1:

**{ якщо** array1[i][j]>0:

product=product\*array1[i][j]; **}**

array2[i]=product; **}**

**для всіх** prohid **від** 1 **до** 8-1 **з кроком** 1:

**для всіх** і **від** 1 **до** 8-1 **з кроком** 1:

**якщо** array2[i]>array2[i+1]:

**{** int memory=array2[i];

array2[i]=array2[i+1];

array2[i+1]=memory; **}**

виводимо одновимірний масив

**кінець**

*Крок 5*

**початок**

**int** array1[8][5];

**для всіх** і **від** 1 **до** 8 **з кроком** 1:

**для всіх** j **від** 1 **до** 5 **з кроком** 1:

array1[i][j]=-99+rand()%199;

**int** array2[8];

**для всіх** і **від** 1 **до** 8 **з кроком** 1:

**{ int** product=1;

**для всіх** j **від** 1 **до** 5 **з кроком** 1:

**{ якщо** array1[i][j]>0:

product=product\*array1[i][j]; **}**

array2[i]=product; **}**

**для всіх** prohid **від** 1 **до** 8-1 **з кроком** 1:

**для всіх** і **від** 1 **до** 8-1 **з кроком** 1:

**якщо** array2[i]>array2[i+1]:

**{** int memory=array2[i];

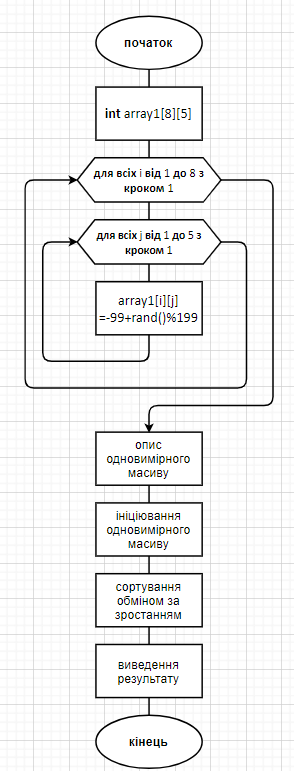
array2[i]=array2[i+1];

array2[i+1]=memory; **}**

**для всіх** і **від** 1 **до** 8 **з кроком** 1:

**вивести** array2[i];

**кінець**

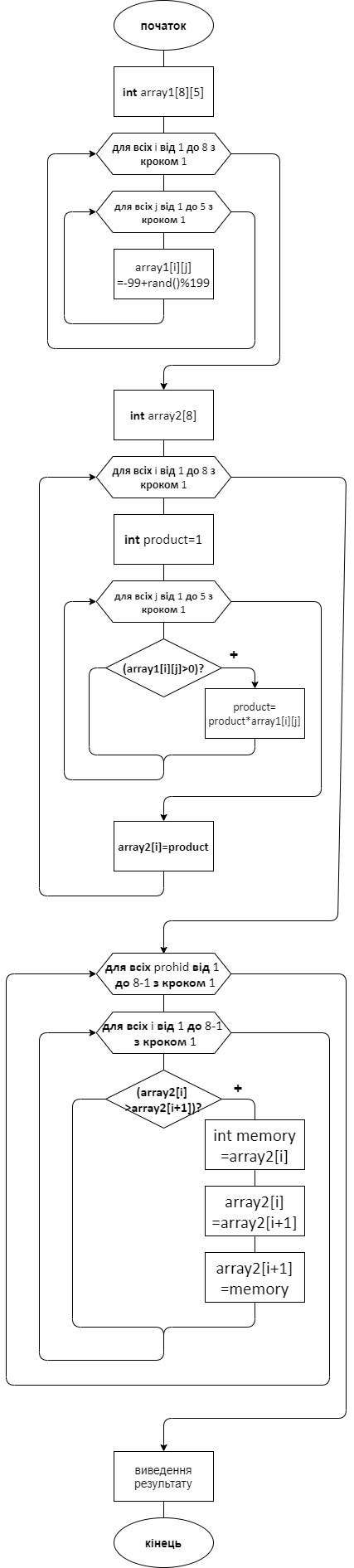
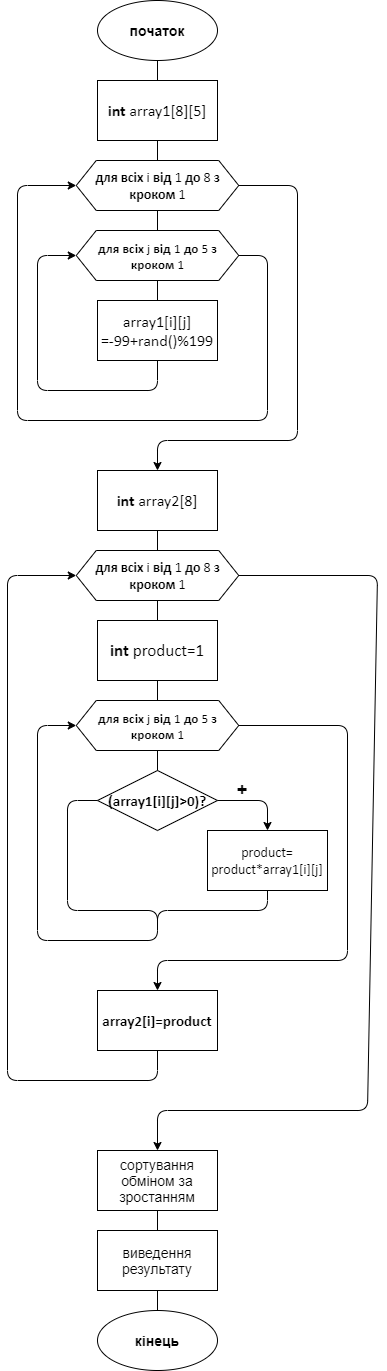
** **

**КРОК 1**

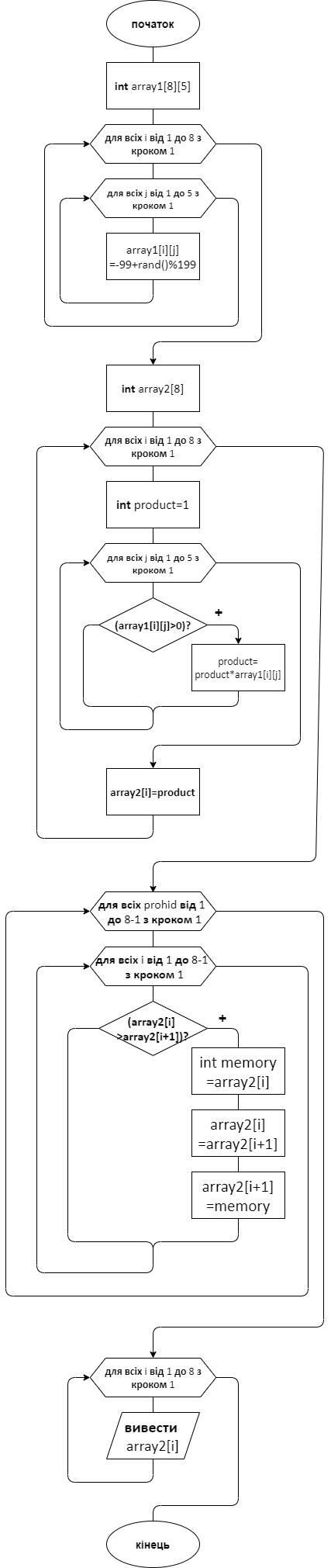
**КРОК 2**

**КРОК 4**

**КРОК 3**



**КРОК 5**



**Код і тестування програми на C++:**

#include <iostream>

#include <ctime>

#include <iomanip>

using namespace std;

int main()

{

int array1[8][5]; //опис змінної індексованого типу згідно варіанту

srand(time(NULL));

cout << "Matrix:"<< endl;

for (int i = 0; i < 8; i++)

{

for (int j = 0; j < 5; j++) //ініціювання двовимірного масиву

{

array1[i][j] = -99 + rand() % 199;

cout << setw(4) << array1[i][j]; //випадкові цілі в проміжку [-99;99]

}

cout << "\n";

}

int array2[8]; //створення нової змінної індексованого типу

cout << "Vector:" << endl;

for (int i = 0; i < 8; i++) //ініціювання нової змінної індексованого типу

{

int product = 1;

for (int j = 0; j < 5; j++)

if (array1[i][j] > 0) product \*= array1[i][j];

array2[i] = product;

if (product != 1) cout << setw(8) << array2[i];

else cout << setw(8) << "-";

}

for (int prohid=0; prohid < 8-1; prohid++) //сортування обміном за зростанням одновимірного масиву

for (int i = 0; i < 8-1; i++)

if (array2[i] > array2[i + 1])

{

int memory = array2[i];

array2[i] = array2[i + 1];

array2[i + 1] = memory;

}

cout << "\nSorted vector:" << endl;

for (int i = 0; i < 8; i++) //вивести результат

{

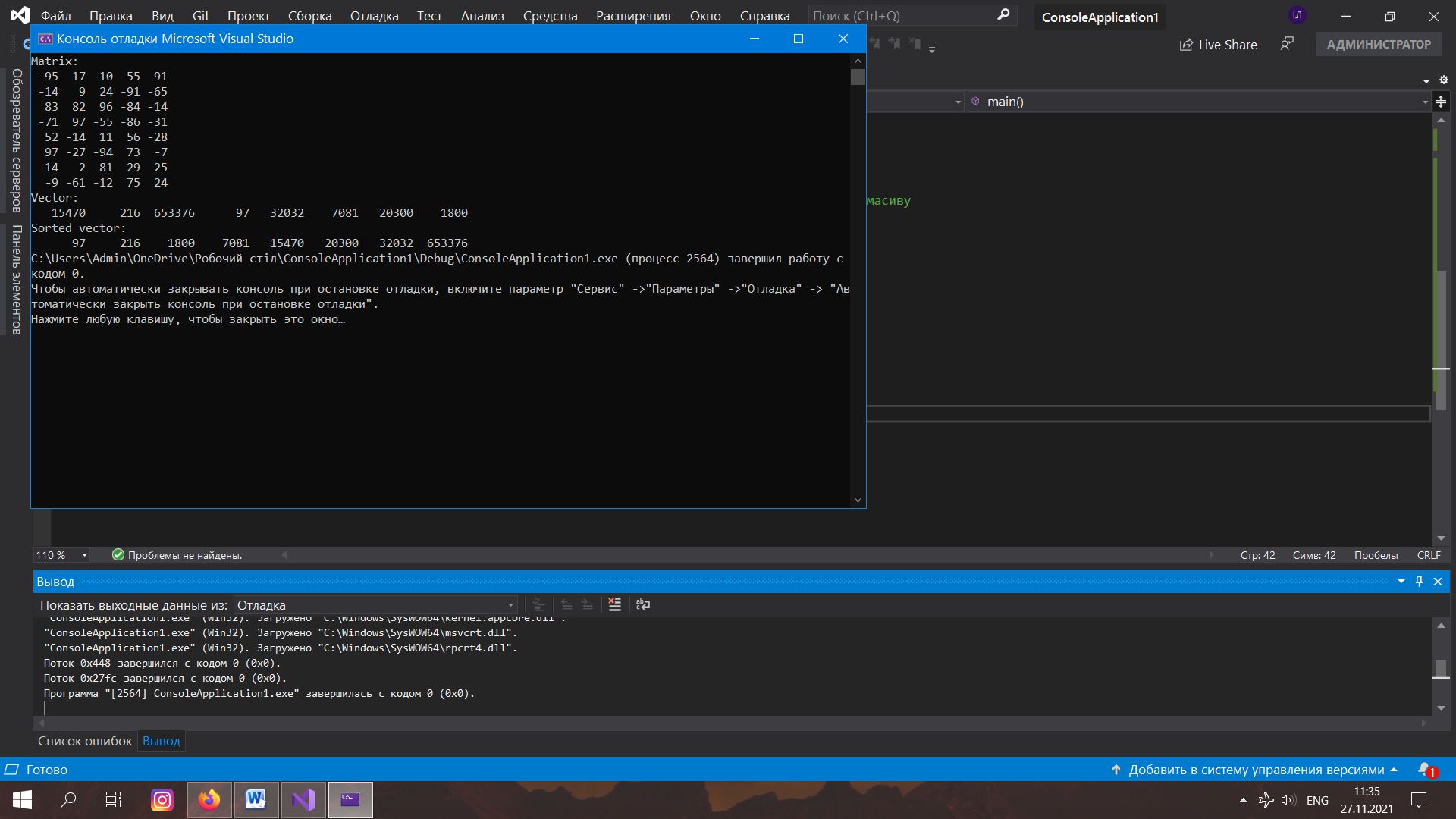
if (array2[i] == 1) cout << setw(8) << "-";

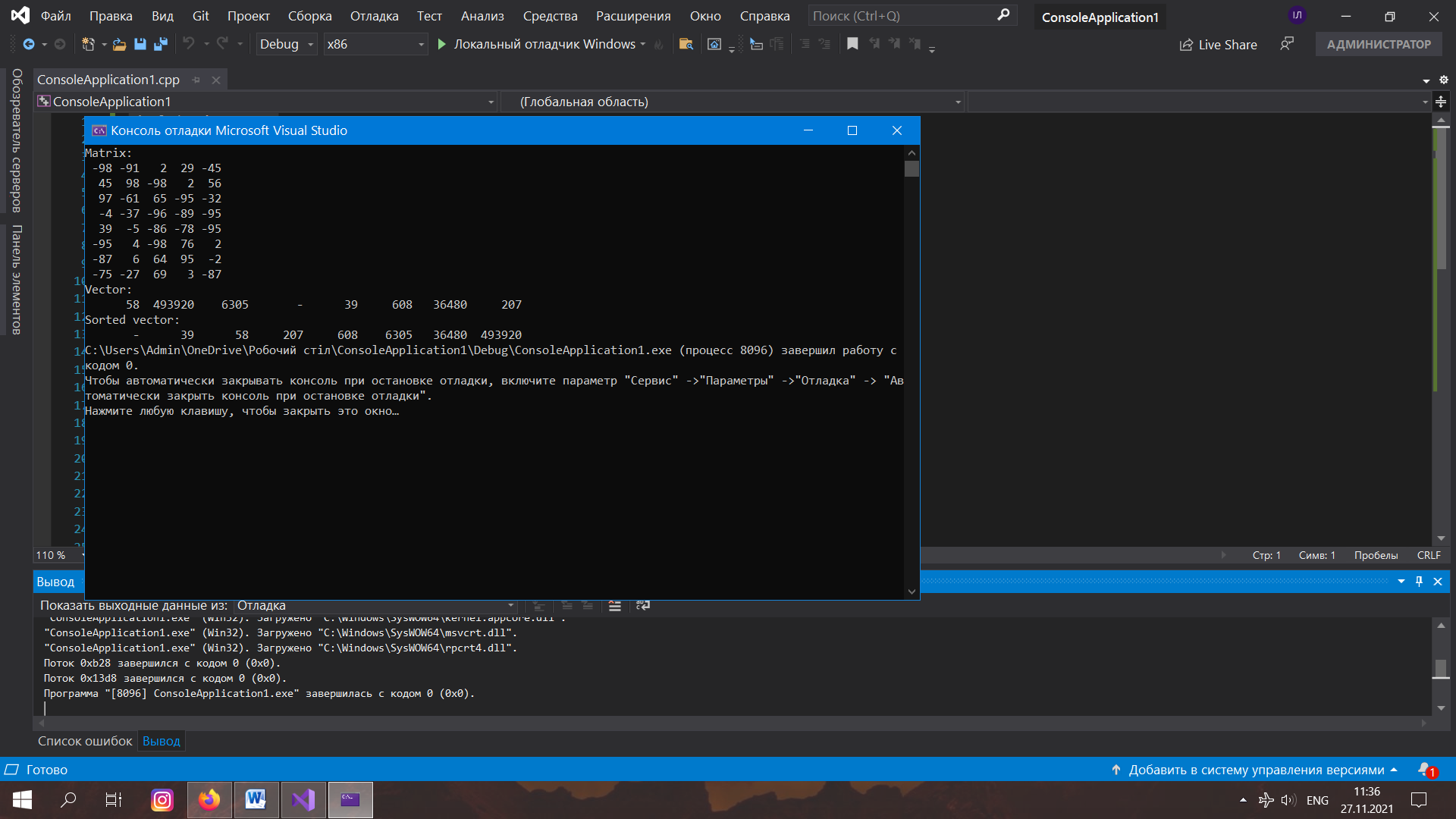
else cout << setw(8) << array2[i];

}

return 0;

}





*Програма виводить спочатку випадково згенерований двовимірний масив, потім обчислений з добутків додатніх елементів по рядках одновимірний масив, а потім цей же масив, але вже сортований за зростанням. Програма була написана так, щоб у разі відсутності додатніх елементів у якомусь з згенерованих випадково рядків, у відповідному елементі одновимірного масиву виводило “-”. Перерахувавши вручну, я переконався що програма працює правильно.*

**Висновок:**

На лабораторній роботі мені довелося попрацювати з пошуком елементів по рядкам матриці цілих чисел за ключем «додатні», їх добутком окремо по кожному рядку, записом знайдених добутків в новий одновимірний масив та сортуванням обміном (бульбашками) цього одновимірного масиву за зростанням. Весь алгоритм ділиться на 4 етапи; на всіх етапах виконання алгоритму, що часто характерно при роботі з масивами, використовується арифметичний цикл (цикл for).