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 20

Виконав студент Медвідь Олександр Русланович

(шифр, прізвище, ім'я, по батькові)

Перевірив Доцент кафедри ІПІ Мартинова О. П.

( прізвище, ім'я, по батькові)

Київ 202 1

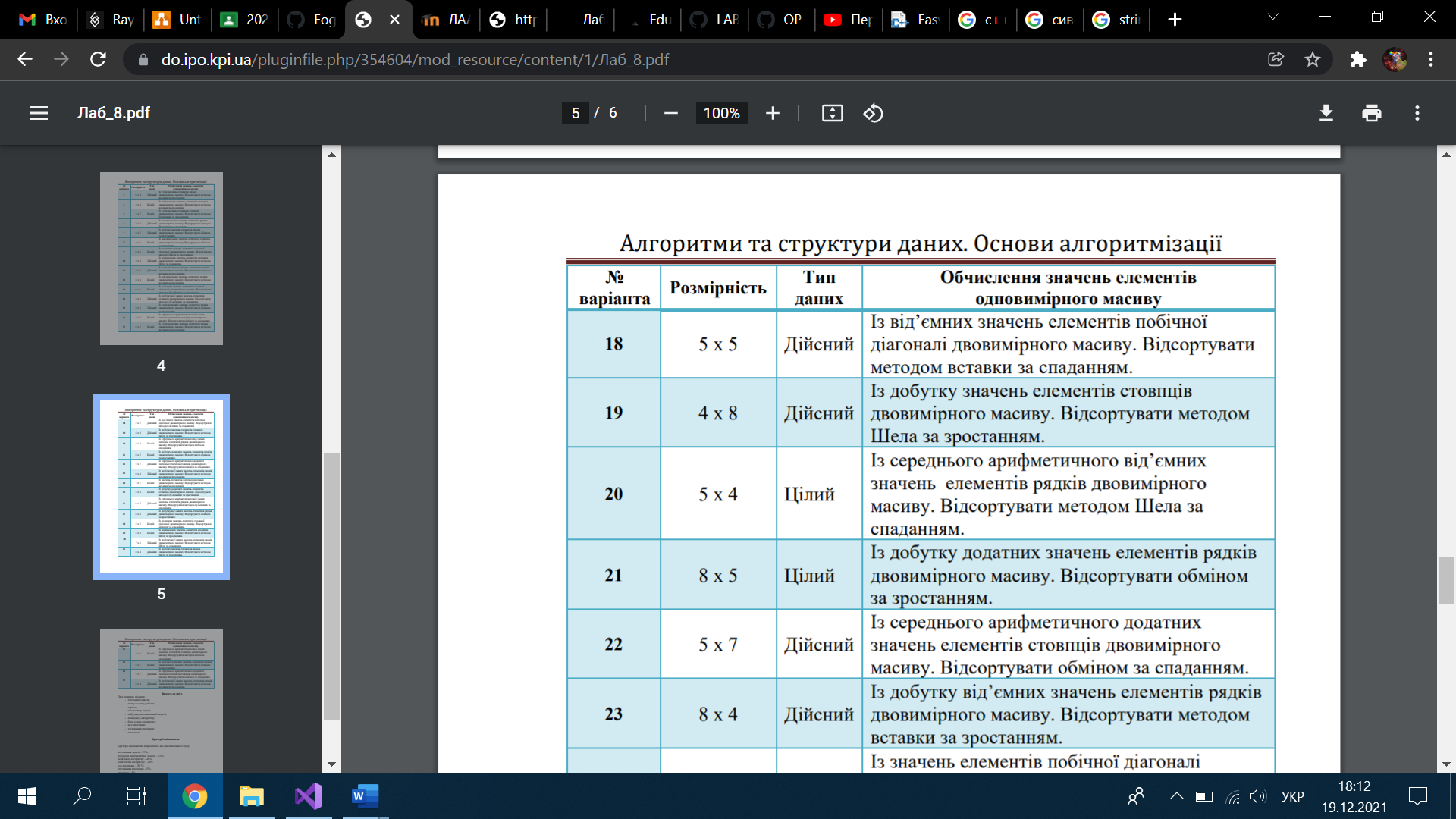
**Лабораторна робота №8**

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

**Завдання**:

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.



**Розв’язок:**

1)Постановка задачі. Результатом розв’язку є відсортований одновимірний масив зі значеннями середніх арифметичних двовимірного масиву. Треба відсортувати методом Шела за спаданням за допомогою алгоритмів пошуку та сортування. Треба ініціювати двовимірний масив значеннями, створеними випадковим чином. Потім треба знайти середнє арифметичне значення від від’ємних значень елементів рядків двовимірного масиву і потім ініціювати одновимірний масив з цими значеннями. За допомогою метода Шела треба відсортувати масив за спаданням.

**Побудова математичної моделі:**

Основна частина:

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Двовимірний масив arr | Цілий | arr | Проміжні дані |
| Одновимірний масив arr1 | Дійсний | arr1 | Результат |
| Розмір n | Цілий | n | Проміжні дані |
| Розмір m | Цілий | m | Проміжні дані |

Процедури initialization та vivod1:

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Процедура initialization | - | initialization | Проміжні дані |
| Процедура vivod1 | - | output | Проміжні дані |
| Одновимірний масив arr1 | Дійсний | arr1 | Результат |
| Лічильник i | Цілий | i | Проміжні дані |
| Розмір n | Цілий | n | Проміжні дані |
| Розмір m | Цілий | m | Проміжні дані |

Функція average

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Функція average | Дійсний | average | Проміжні дані |
| Двовимірний масив arr | Цілий | arr | Проміжні дані |
| Одновимірний масив arr1 | Дійсний | arr1 | Результат |
| Лічильник i | Цілий | i | Проміжні дані |
| Лічильник j | Цілий | j | Проміжні дані |
| Сума sum | Цілий | sum | Проміжні дані |
| Розмір n | Цілий | n | Проміжні дані |
| Розмір m | Цілий | m | Проміжні дані |

Процедура vivod2:

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Процедура vivod2 | - | vivod2 | Проміжні дані |
| Одновимірний масив arr1 | Дійсний | arr1 | Результат |
| Лічильник i | Цілий | i | Проміжні дані |
| Розмір n | Цілий | n | Проміжні дані |

Функція shell:

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Функція shell | - | shell | Проміжні дані |
| Двовимірний масив arr | Цілий | arr | Проміжні дані |
| Одновимірний масив arr1 | Дійсний | arr1 | Результат |
| Лічильник i | Цілий | i | Проміжні дані |
| Лічильник j | Цілий | j | Проміжні дані |
| Змінна temp | Дійсний | Temp | Проміжні дані |
| Крок d | Цілий | d | Проміжні дані |
| Розмір масиву n | Цілий | n | Проміжні дані |

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію ініціювання двовимірного масиву за допомогою процедури initialization.

Крок 3. Деталізуємо дію знаходження одновимірного масиву за допомогою функції average.

Крок 4. Деталізуємо дію виведення масивів за допомогою процедури vivod1 та vivod2.

Крок 5. Деталізуємо дію сортування одновимірного масиву за допомогою процедури shell.

Крок 6. Деталізуємо дію сортування одновимірного масиву в основному тілі програми.

**Псевдокод:**

**Крок 1**

**Початок**

n, m, arr[n][m], arr1[n][m]

дію ініціювання двовимірного масиву за допомогою процедури initialization.

Деталізуємо дію знаходження одновимірного масиву за допомогою функції average.

Деталізуємо дію сортування одновимірного масиву за допомогою процедури shell.

Деталізуємо дію виведення масивів за допомогою процедури vivod1 та vivod2.

Виведення arr1.

**Кінець**

**Процедура**

**initialization(arr[], n, m):**

Деталізуємо дію ініціювання двовимірного масиву за допомогою процедури initialization.

**Все процедура**

**Процедура**

**vivod1(arr[], n, m):**

Деталізуємо дію виведення двовимірного масиву за допомогою процедури initialization.

**Все процедура**

**Функція**

**average(arr[], n, m):**

Деталізуємо дію знаходження одновимірного масиву за допомогою функції average.

**Все функція**

**Процедура**

**vivod2(arr1[], n):**

Деталізуємо дію виведення одновимірного масиву за допомогою процедури vivod2.

**Все процедура**

**Процедура**

**shell(arr1[], n):**

Деталізуємо дію сортування одновимірного масиву за допомогою процедури shell.

**Все процедура**

**Крок 2**

**Початок**

n=5, m=4, arr[n][m], arr1[n][m]

initialization(arr, n, m ).

average(arr, n, m ).

shell(arr, n, m).

vivod1(arr1, n).

Shell (arr1, n).

vivod2(arr1, n).

Виведення arr1.

**Кінець**

**Процедура**

**initialization(arr[], n, m):**

**повторити для і від 0 до n не включно з кроком 1:**

**повторити для j від 0 до m не включно з кроком 1:**

arr[i][j]=rand()%21-10

**все повторити**

**все повторити**

**Все процедура**

**Процедура**

**vivod1(arr[], n, m):**

**повторити для і від 0 до n не включно з кроком 1:**

**повторити для j від 0 до m не включно з кроком 1:**

виведення arr[i][j]

**все повторити**

**все повторити**

**Все процедура**

**Функція**

**average(arr[], n, m):**

k=0; sum=0; arr1[5]

**повторити для і від 0 до n не включно з кроком 1:**

**повторити для j від 0 до m не включно з кроком 1:**

**якщо** (arr[i][j]<0)

**то**

sum+=arr[i][j]

k+=1

**все якщо**

**якщо** (k==0)

**то**

arr1[i]=0

**інакше**

arr1[i]=sum/k

**все якщо**

k=0

sum=0

**все повторити**

**все повторити**

**return arr1**

**Все функція**

**Процедура**

**vivod2(arr1[], n):**

**повторити для і від 0 до n не включно з кроком 1:**

виведенняarr1[i]

**все повторити**

**Все процедура**

**Процедура**

**shell(arr1[], n):**

temp, d=n/2

**поки d>0**

**повторити для і від 0 до (n-d) не включно з кроком 1:**

j=1

**поки (j>=0 && arr1[j]<arr1[j+d])**

temp=arr1[j]

arr1[j]=arr1[j+d]

arr1[j+d]=temp

j--

**все повторити**

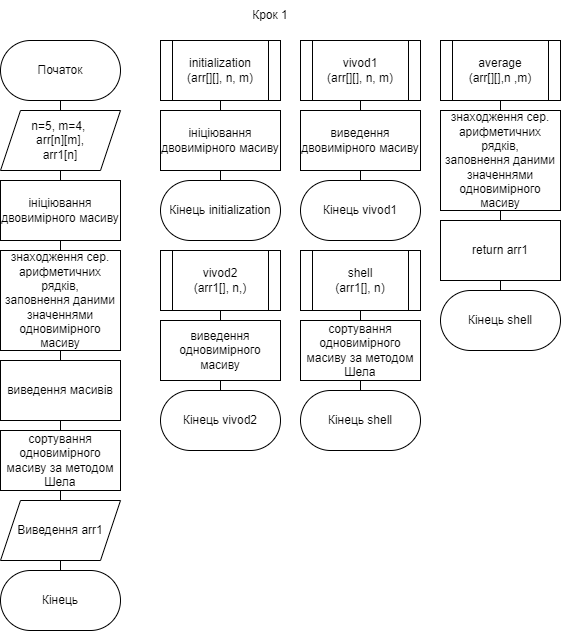
**все повторити**

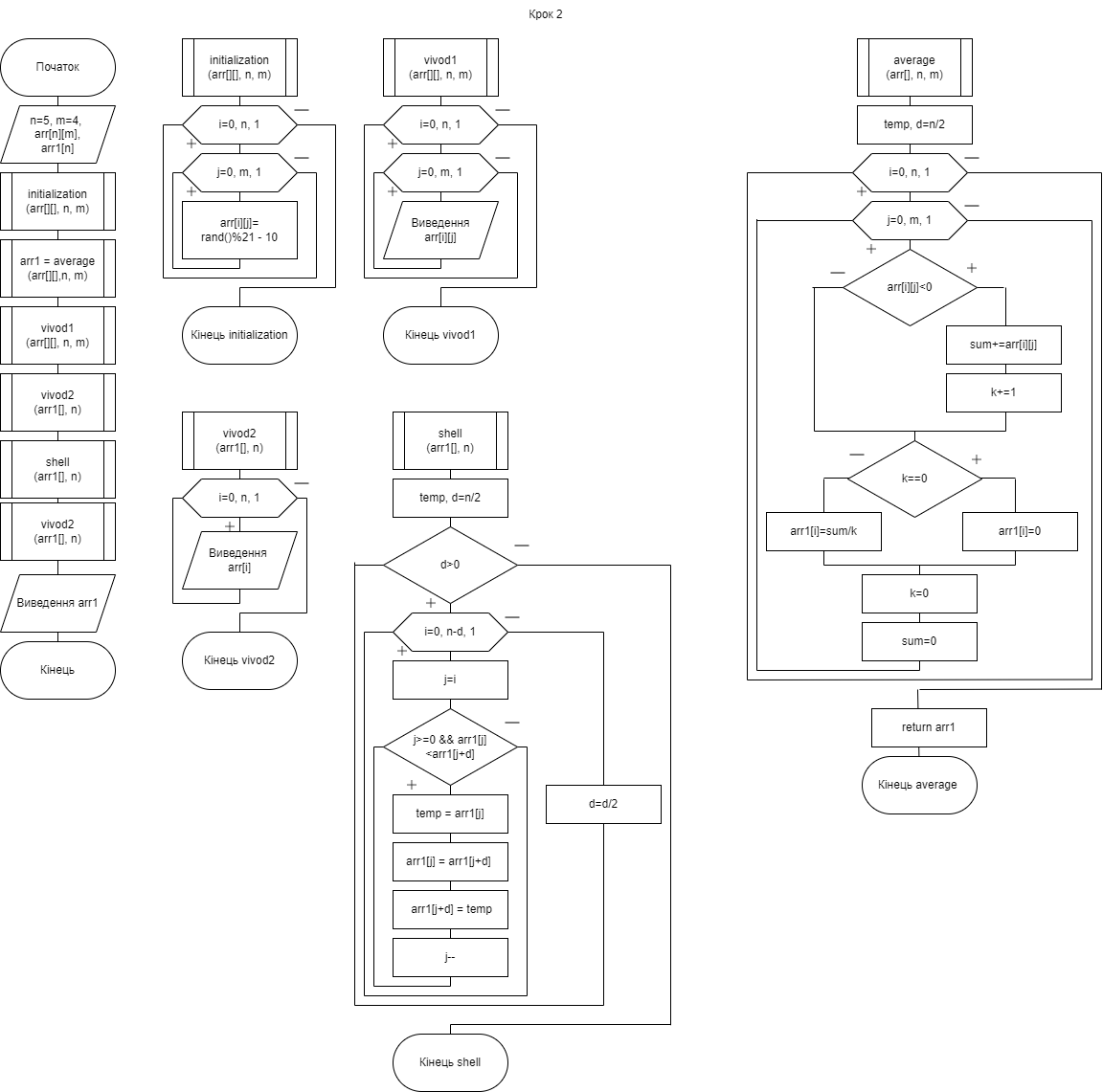
**все повторити**

**return arr1**

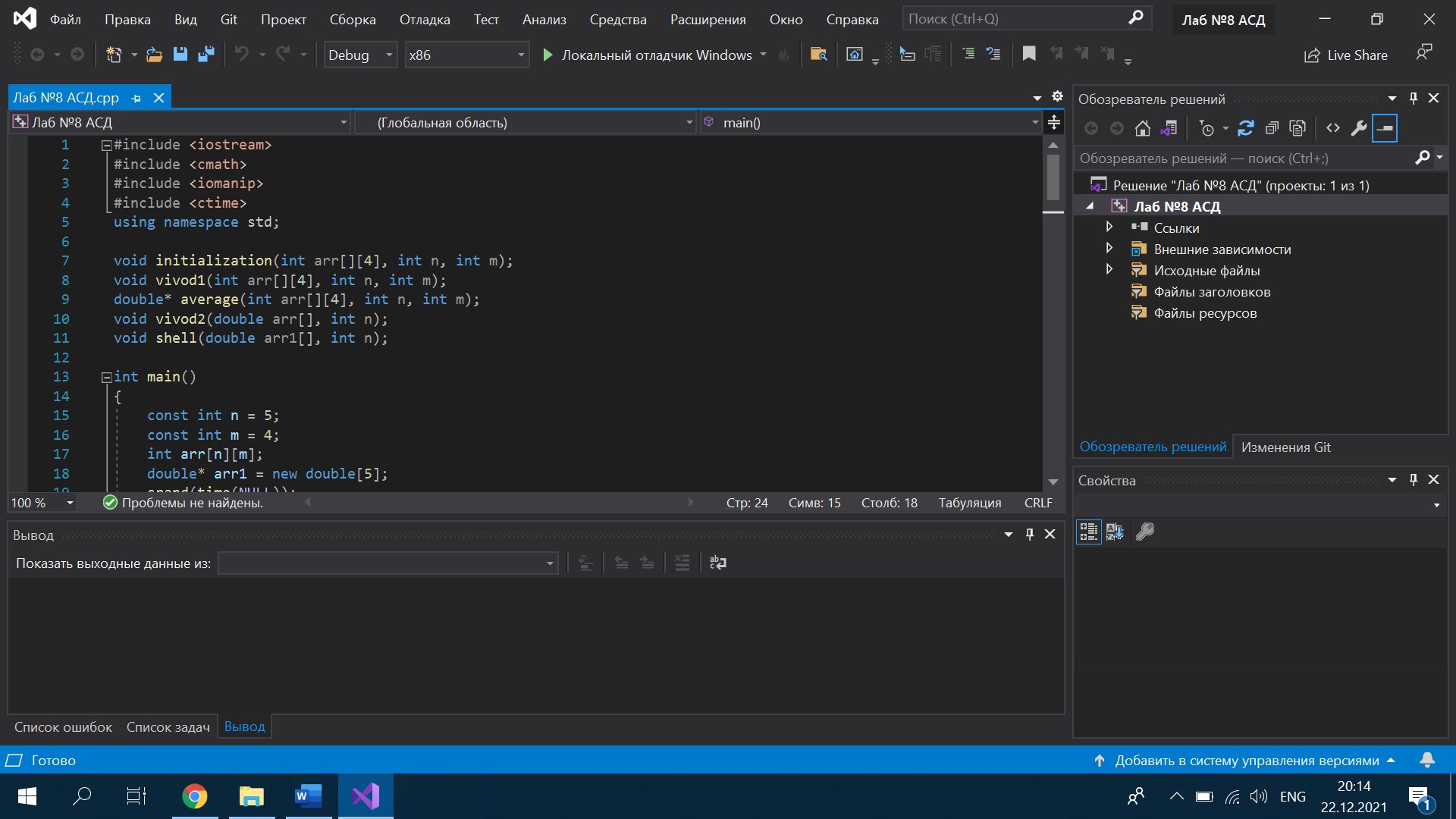
**Все процедура**

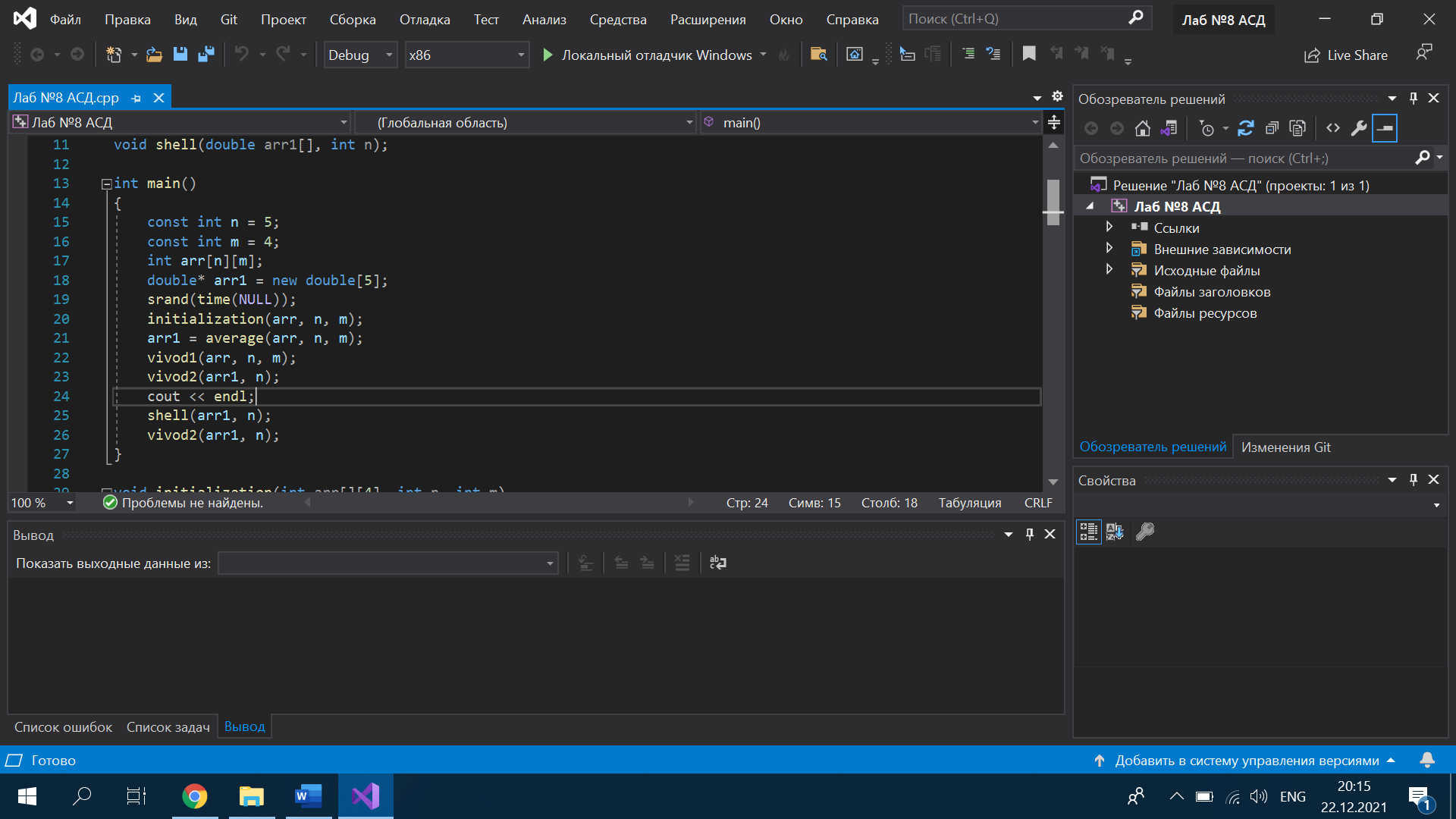
**Блок-схема:**

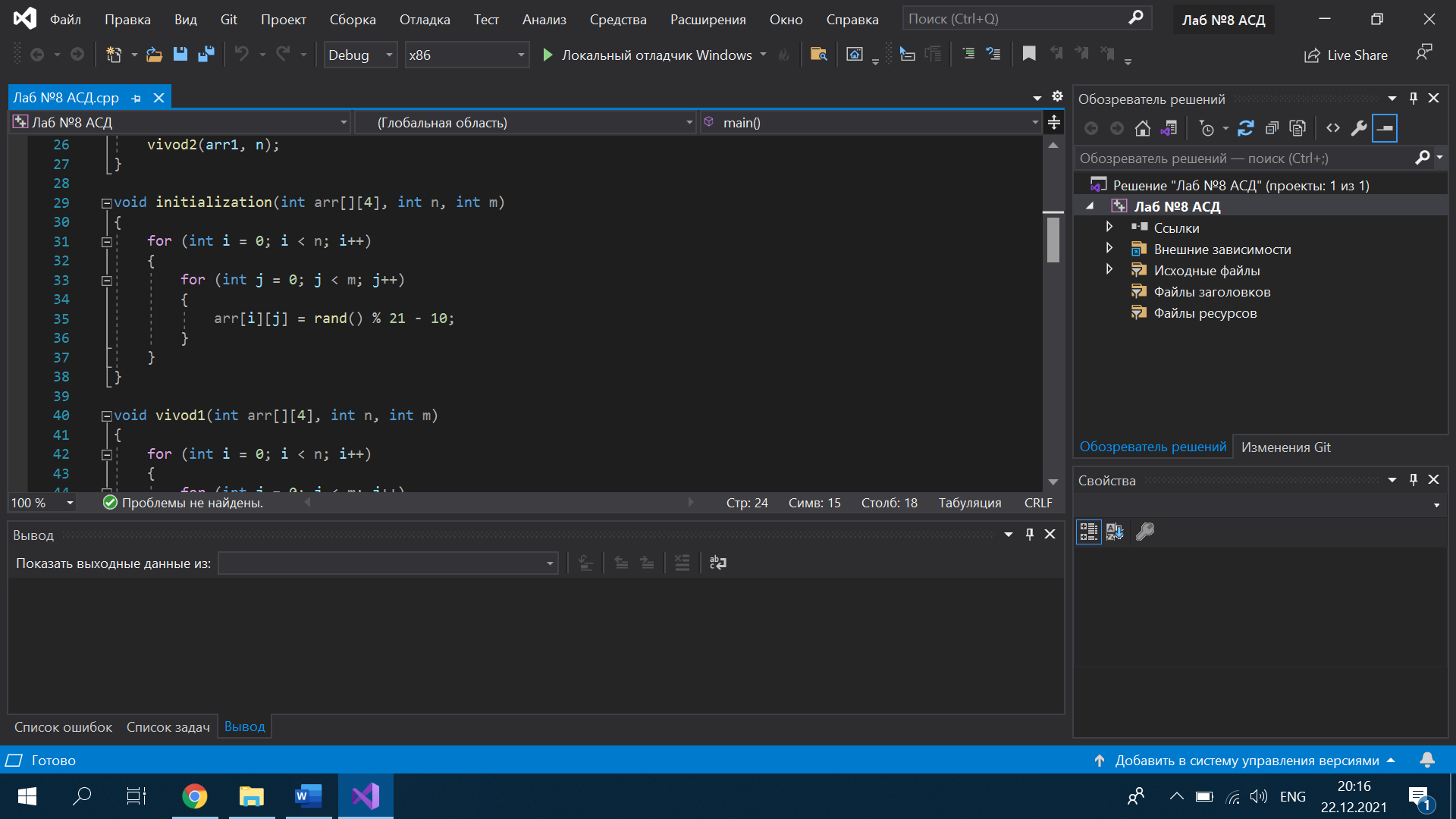


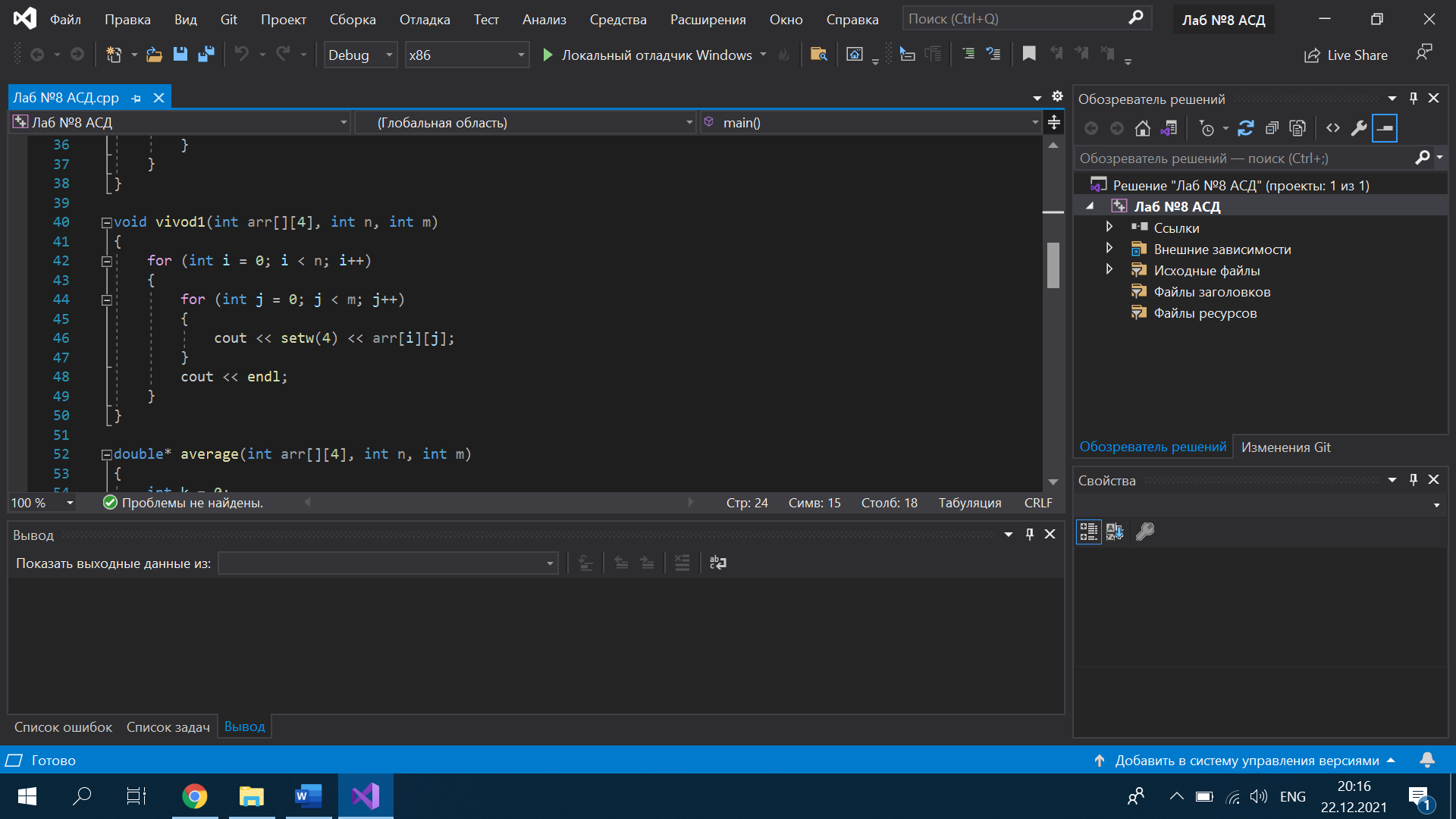


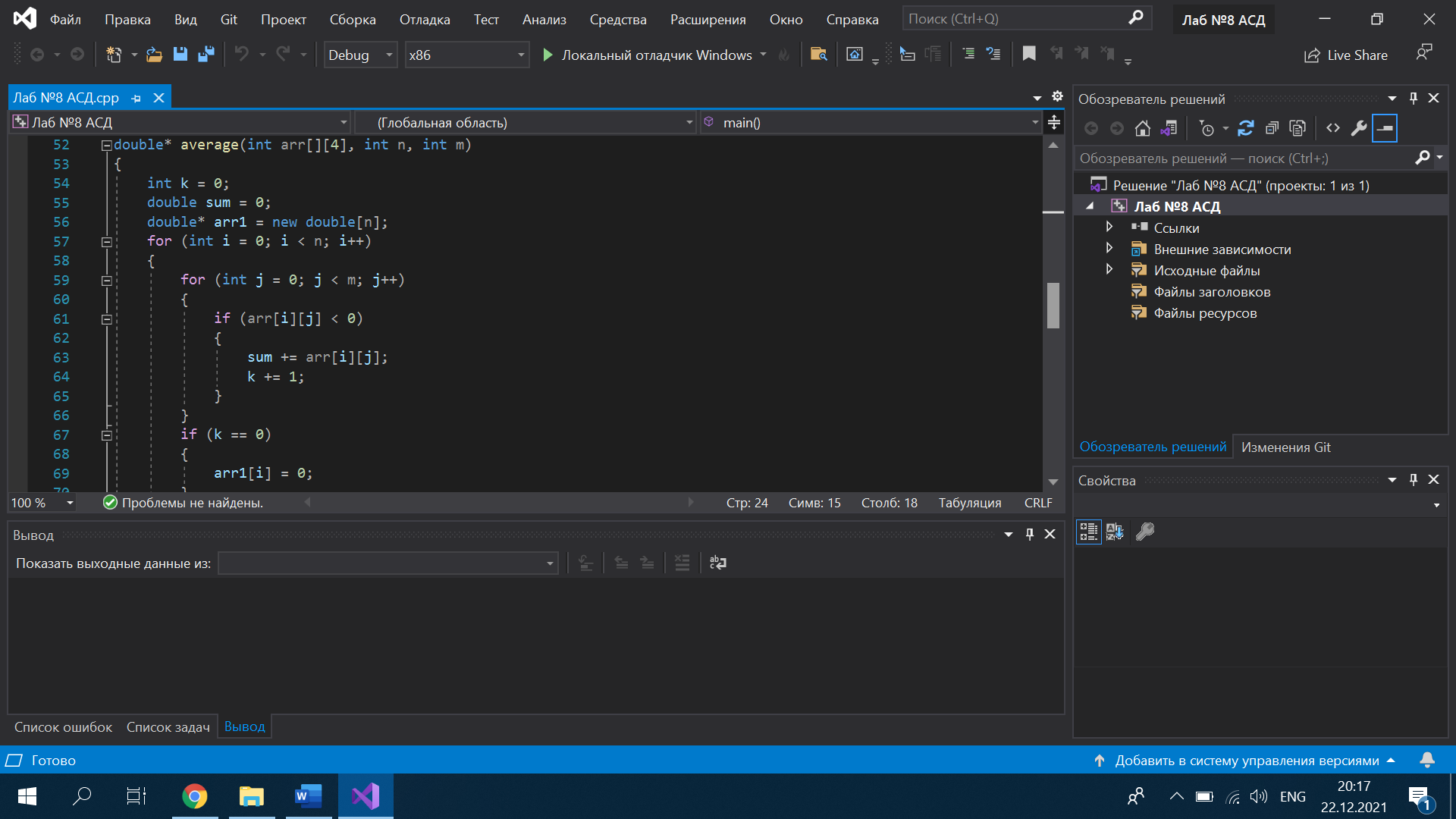
**Код на С++:**

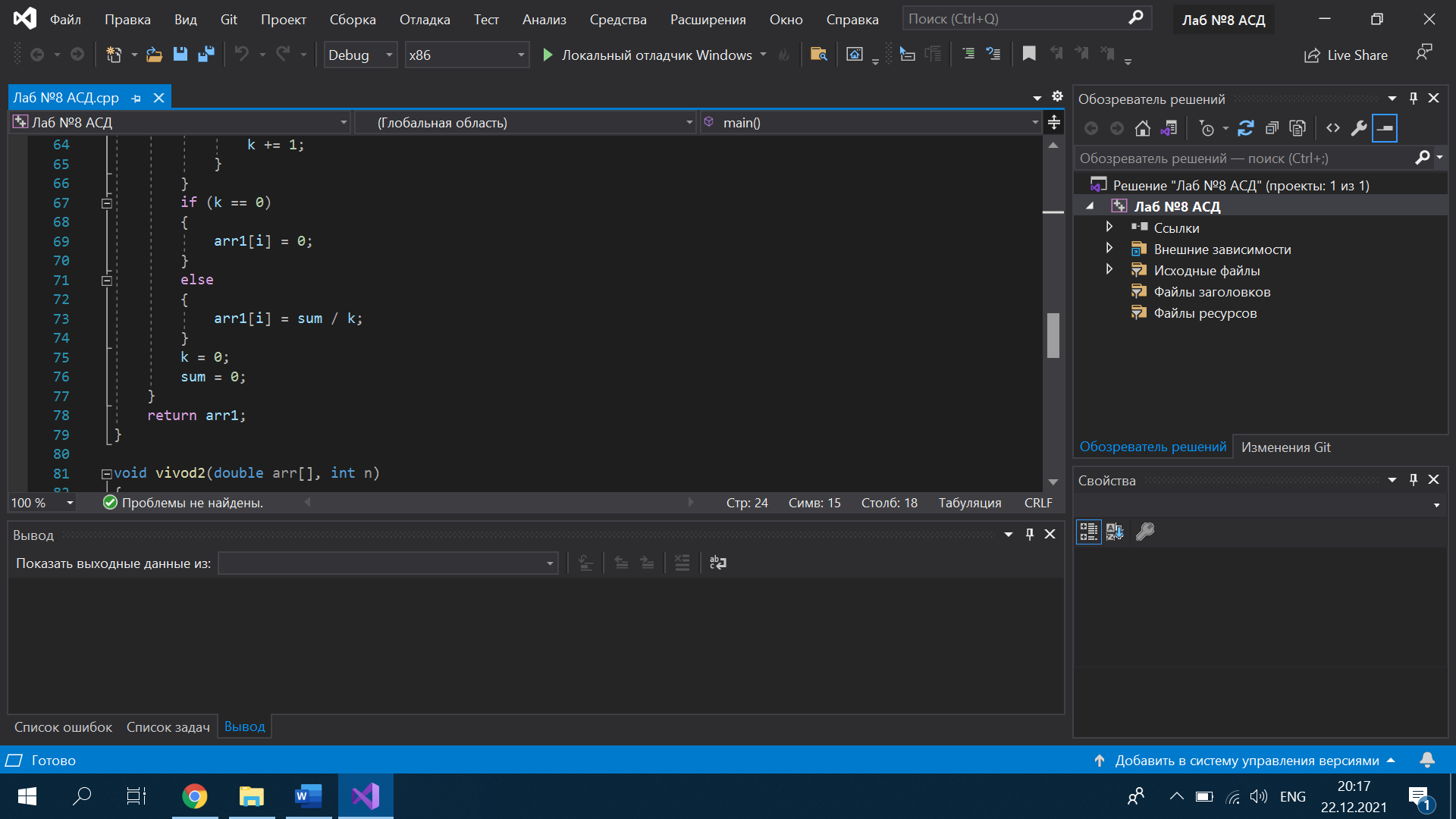


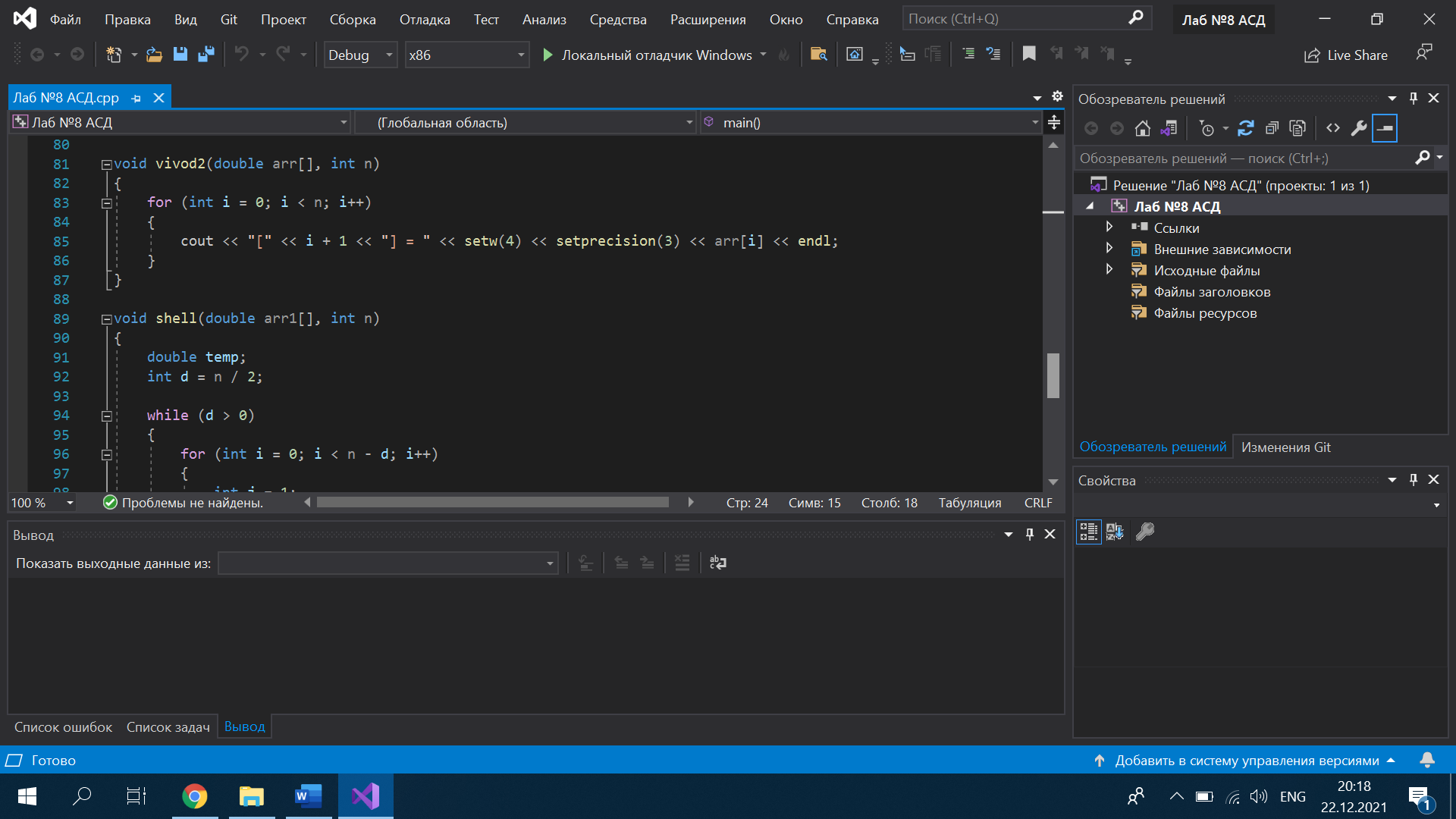


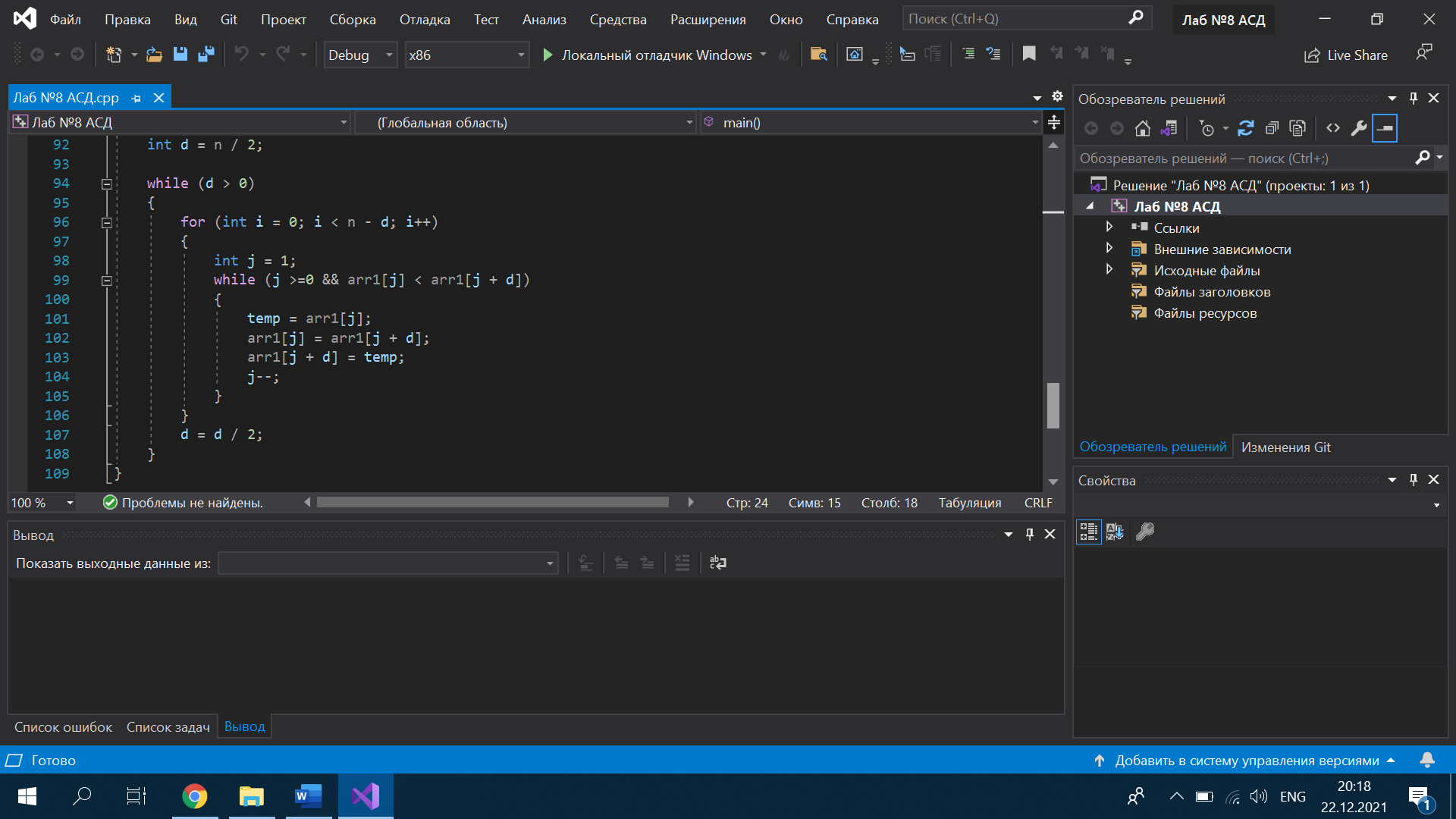


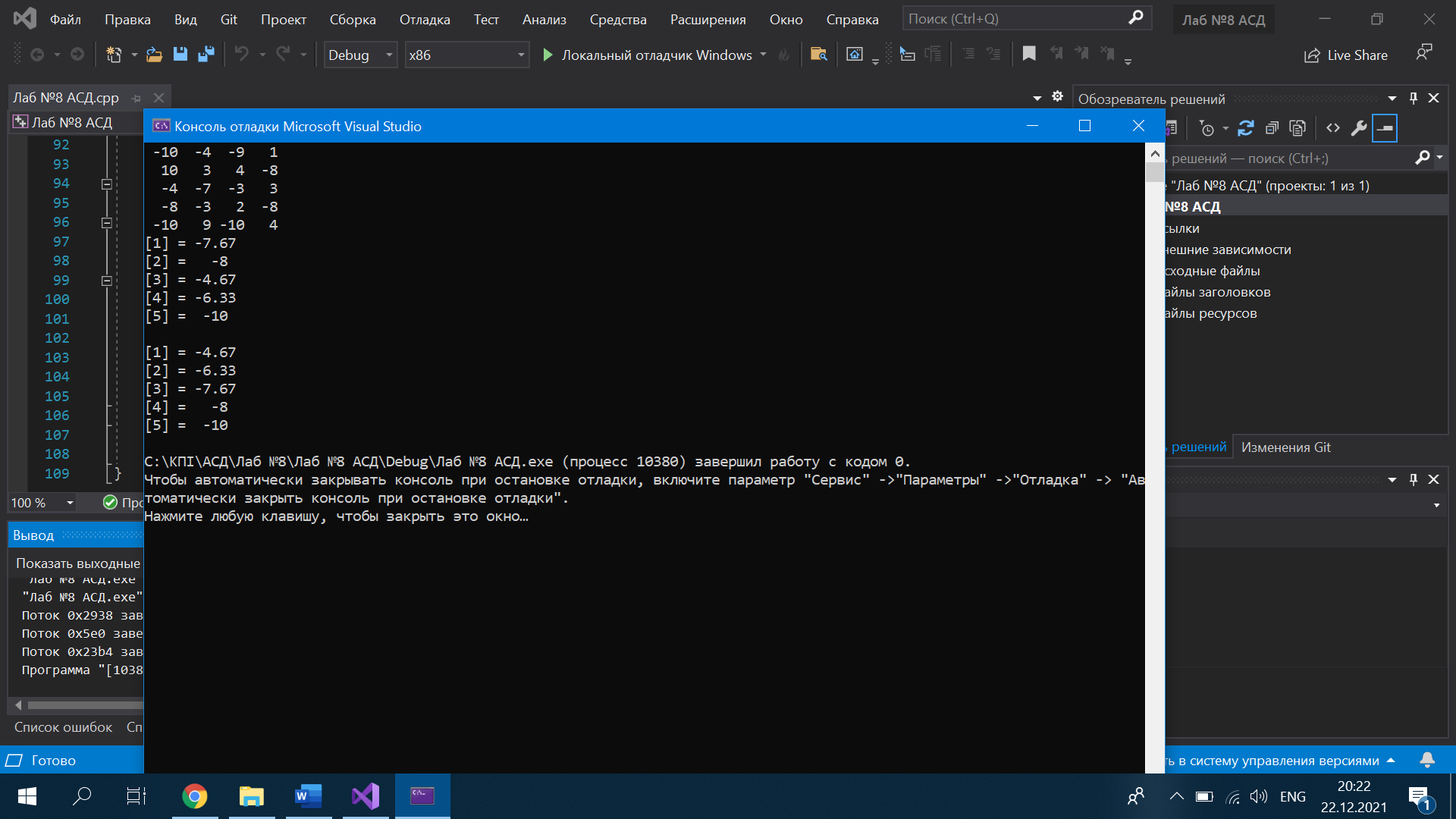












**Висновок:** Під час даної лабораторної роботи були досліджені алгоритми пошуку та сортування. Масив був відсортований за допомогою методу Шела. Були набуті навички використання їх у програмних специфікаціях. Були побудовані блок схема та математичка модель. Для обчислення виразу була створена елементарна програма. Результати програми виявилися правильними, що стверджує на її дієвість. Завдання було виконано на мові програмування: C++.