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 № 9 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження алгоритмів обходу масивів»

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 №9**

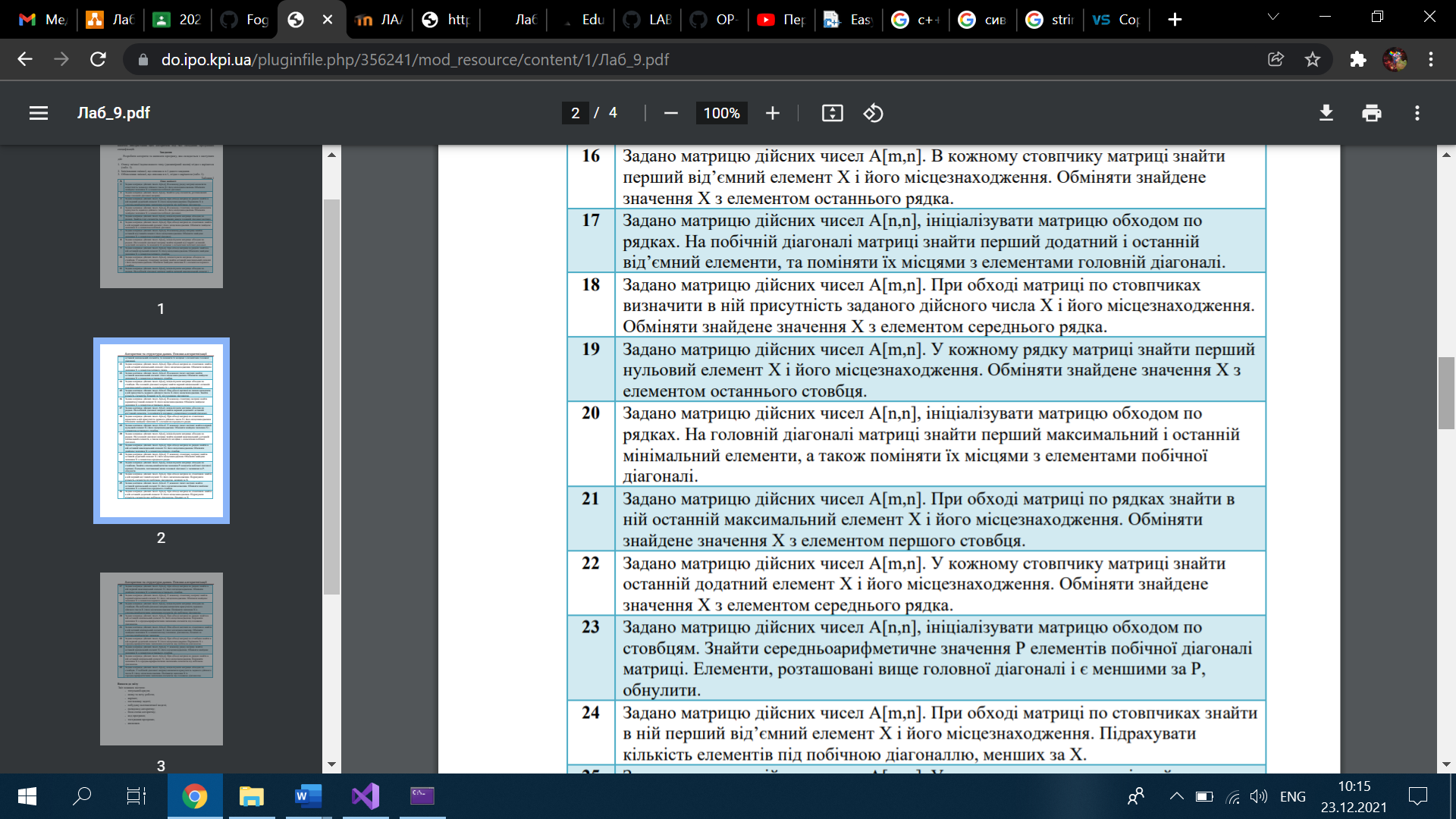
**Мета**: дослідити алгоритми обходу масивів набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

**Завдання**:

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

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

3. Обчислення змінної, що описана в п.1, згідно з варіантом.



**Розв’язок:**

1)Постановка задачі. Знайти значення першого максимального та останнього мінімальних елементів головної діагоналі двовимірного масиву та поміняти їх місцями з крайніми значеннями побічної діагоналі. Ініціалізувати матрицю обходом по рядках(змійкою). Ініціюємо двовимірний масив випадковими дійсними значеннями. Далі знаходимо перший максимальний та останній мінімальний елементи головної діагоналі та їх позиції. Замінюємо ці значення крайніми значеннями побічної діагоналі матриці, якщо позиція максимального елемента менша за позицію мінімального, то замінюємо максимальний елемент головної діагоналі на перший елемент побічної, з мінімальним аналогічно та навпаки. Результатом розв’язку буде змінений двовимірний масив, де перший максимальний та останній мінімальний елементи поміняні з крайніми елементами побічної діагоналі.

**Побудова математичної моделі:**

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Процедура задання значень | void | Birth | Проміжні дані |
| Процедура виведення масиву | Void | Vivod | Проміжні дані |
| Функція задання розміру масиву | Int | Size | Проміжні дані |
| Функція створення динамічного масиву | Double\*\* | Vvod | Проміжні дані |
| Функція знаходження мінімального елементу масиву та його позиції | double | min | Проміжні дані |
| Процедура форматування масиву | Void | Newmat | Проміжні дані |
| Процедура видалення памяті під масив | Void | Extermination | Проміжні дані |
| Двовимірний масив | Double | Matrix | Результат |
| Позиція максимального елементу | Int | Pos1 | Проміжні дані |
| Позиція мінімального елементу | Int | Pos2 | Проміжні дані |
| Розмір n | Int | N | Вхідні дані |
| Максимальний елемент | Double | Max1 | Проміжні дані |
| Мінімальний елемент | double | Min1 | Проміжні дані |

Таким чином математичне формулювання задачі зводиться до створення динамічного двовимірного масиву розмірністю n, потім ініціалізація масиву обходом по рядках(змійкою) випадковими дійсними числами. Знаходження максимального та мінімального елементів та заміна їх на перший та останній елементи побічної діагоналі у відповідності до позиції макс. та мін. елементів. Використовую функцію rand() – за допомогою, якої заповняю масив випадковими значеннями.

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію знаходження розміру двовимірного масиву.

Крок 3. Деталізуємо дію створення двовимірного масиву.

Крок 4. Деталізуємо дію ініціювання масиву обходом по рядках.

Крок 5. Деталізуємо дію виведення початкового масиву.

Крок 6. Деталізуємо дію знаходження першого максимального елемента та його позиції.

Крок 7. Деталізуємо дію знаходження останнього мінімального елемента та його позиції.

Крок 8. Деталізуємо дію форматування двовимірного массиву(заміни значень).

Крок 9. Деталізуємо дію виведення форматованого масиву.

Крок 10. Деталізуємо дію очищення пам’яті.

**Псевдокод:**

Початок

pos1, pos2,

n = size()

matrix = vvod(n)

birth(matrix, n)

vivod(matrix, n)

max1 = max(matrix, n, pos\_1)

min1 = min(matrix, n, pos\_2)

вивід max1, pos1, min1, pos2

newmat(matrix, n, pos1, pos2, max1, min1)

vivod(matrix, n)

extermination(matrix, n)

Кінець

Функція size():

n

ввід n

return n

все функція

Функція vvod(n):

\*\*matrix = new double\* [n]

повторити для і від 0 до n:

matrix[n] = new double[n]

все повторити

return matrix

все функція

Процедура birth(\*\*matrix, n):

повторити для і від 0 до n:

якщо (i%2 == 0) то

повторити для j від 0 до n з кроком 1:

matrix[i][j]= 0,1\*(rand() % 101)

все повторити

інакше повторити для j від n-1 до 0 з кроком -1:

matrix[i][j]= 0,1\*(rand() % 101)

все повторити

все якщо

все повторити

Все процедура

Процедура vivod(\*\*matrix, n):

повторити для і від 0 до n: повторити для j від 0 до n:

виведення matrix[i][j]

все повторити

все повторити

Все процедура

Функція max(\*\*matrix, n, &pos\_1)

max = matrix[0][0], pos\_1 = 0

повторити для і від 0 до n:

повторити для j від 0 до n:

якщо (i == j) то

якщо (matrix[i][j] >max)

то max = matrix[i][j] pos\_1 = i

все якщо

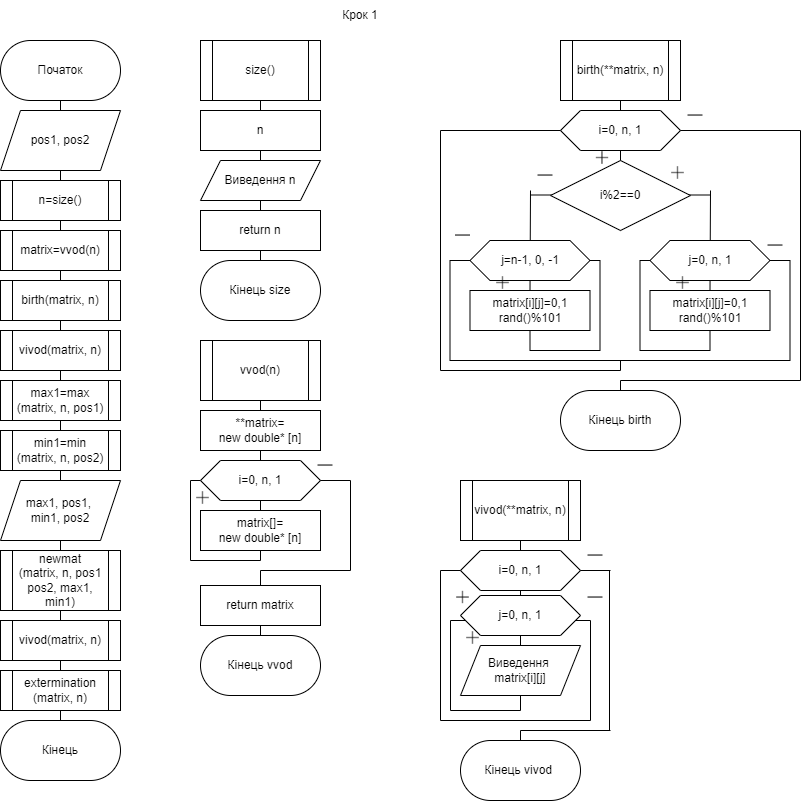
все якщо

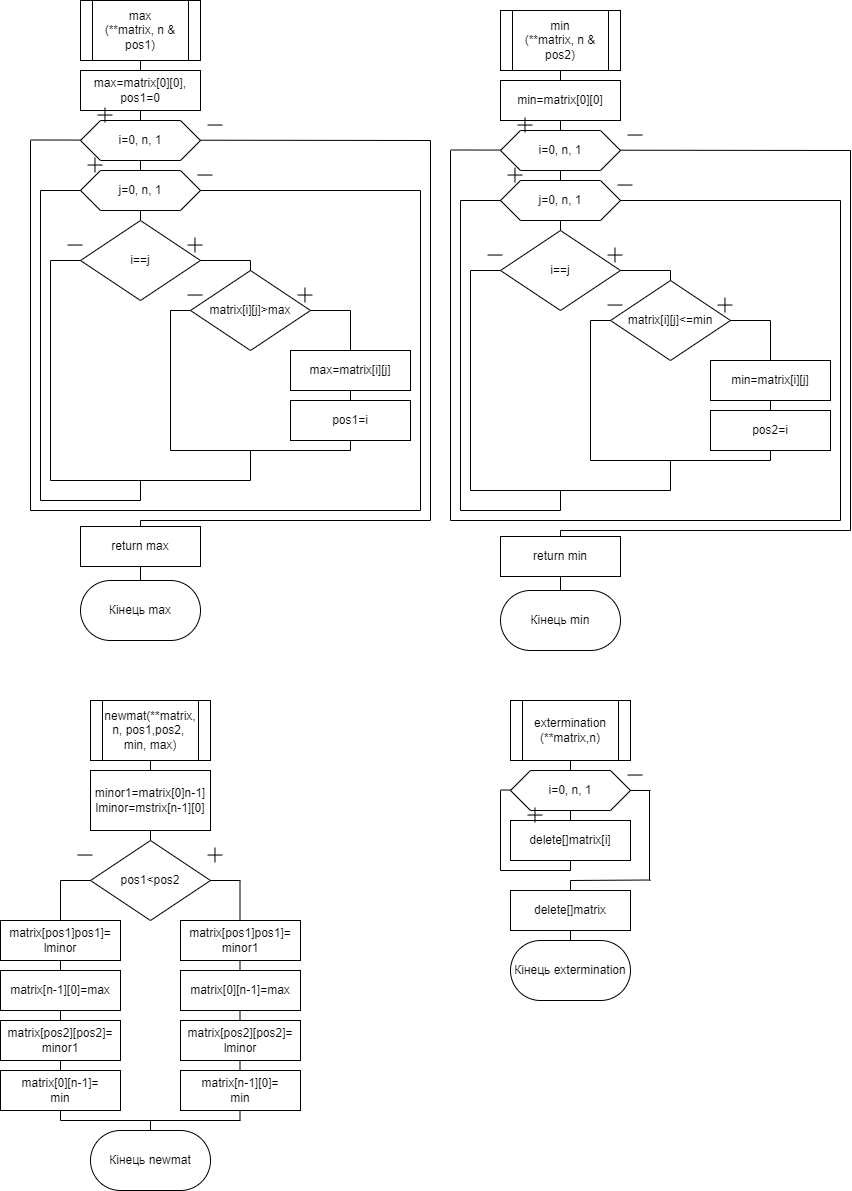
все повторити

все повторити

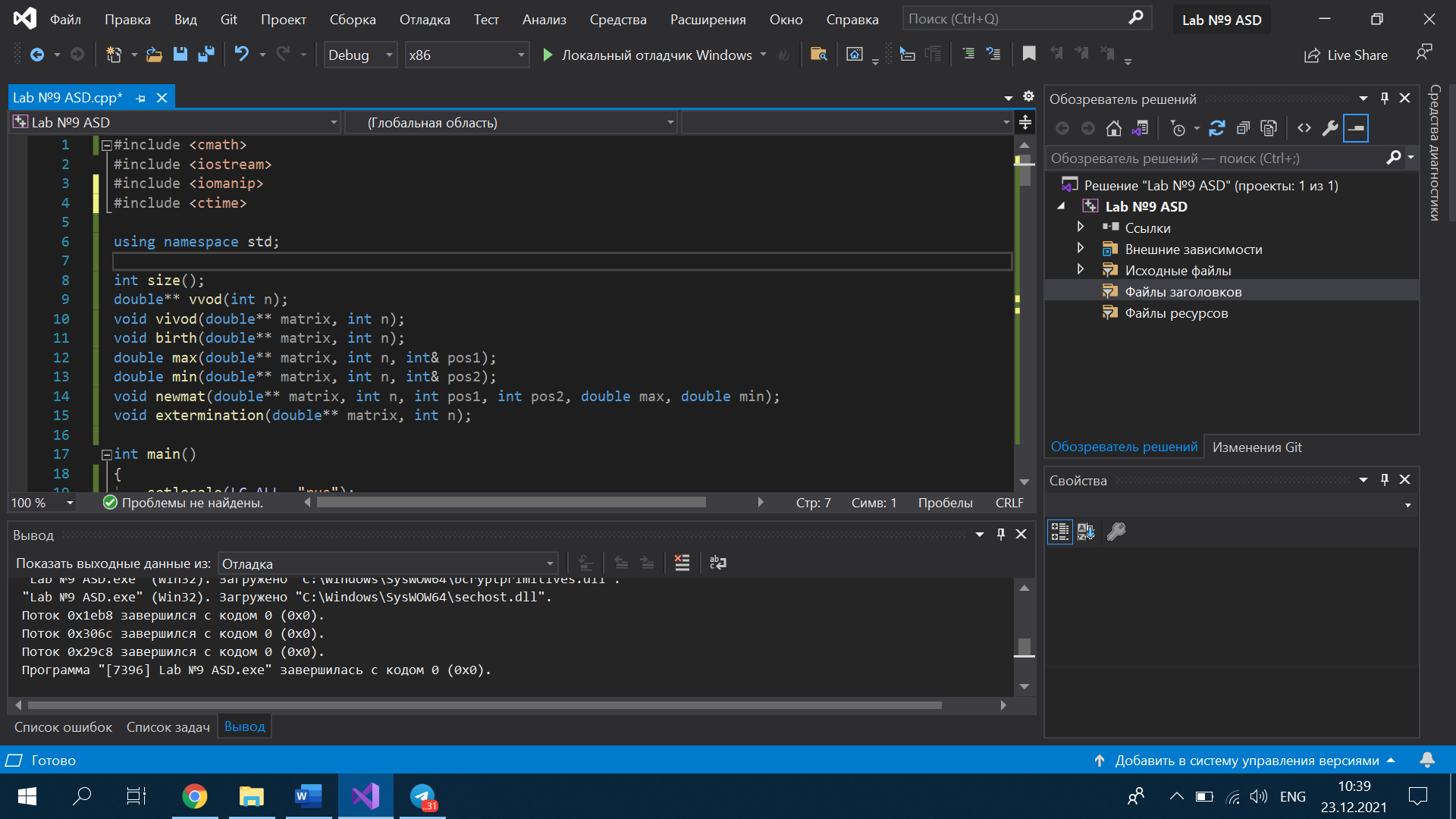
return max

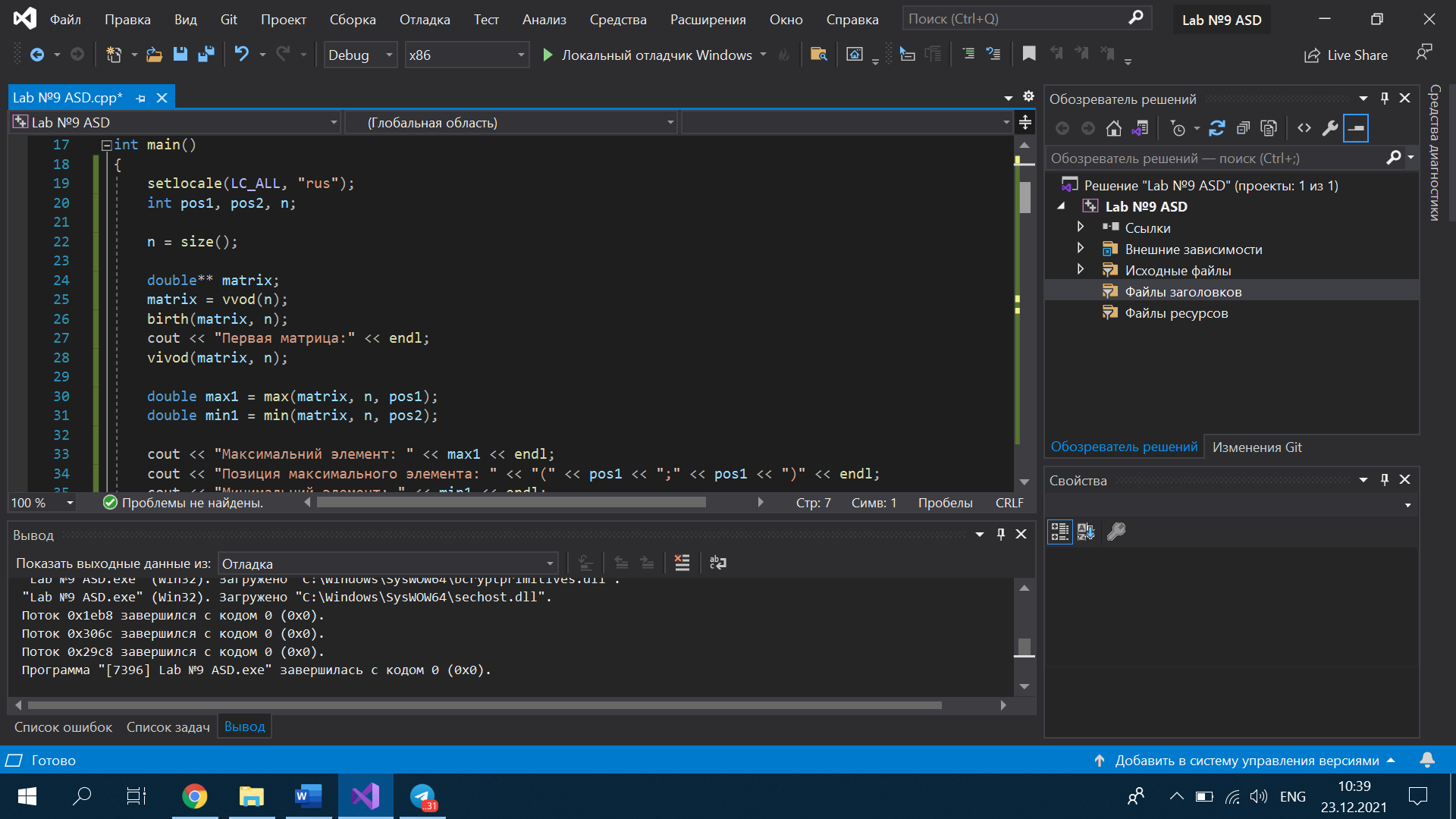
все функція

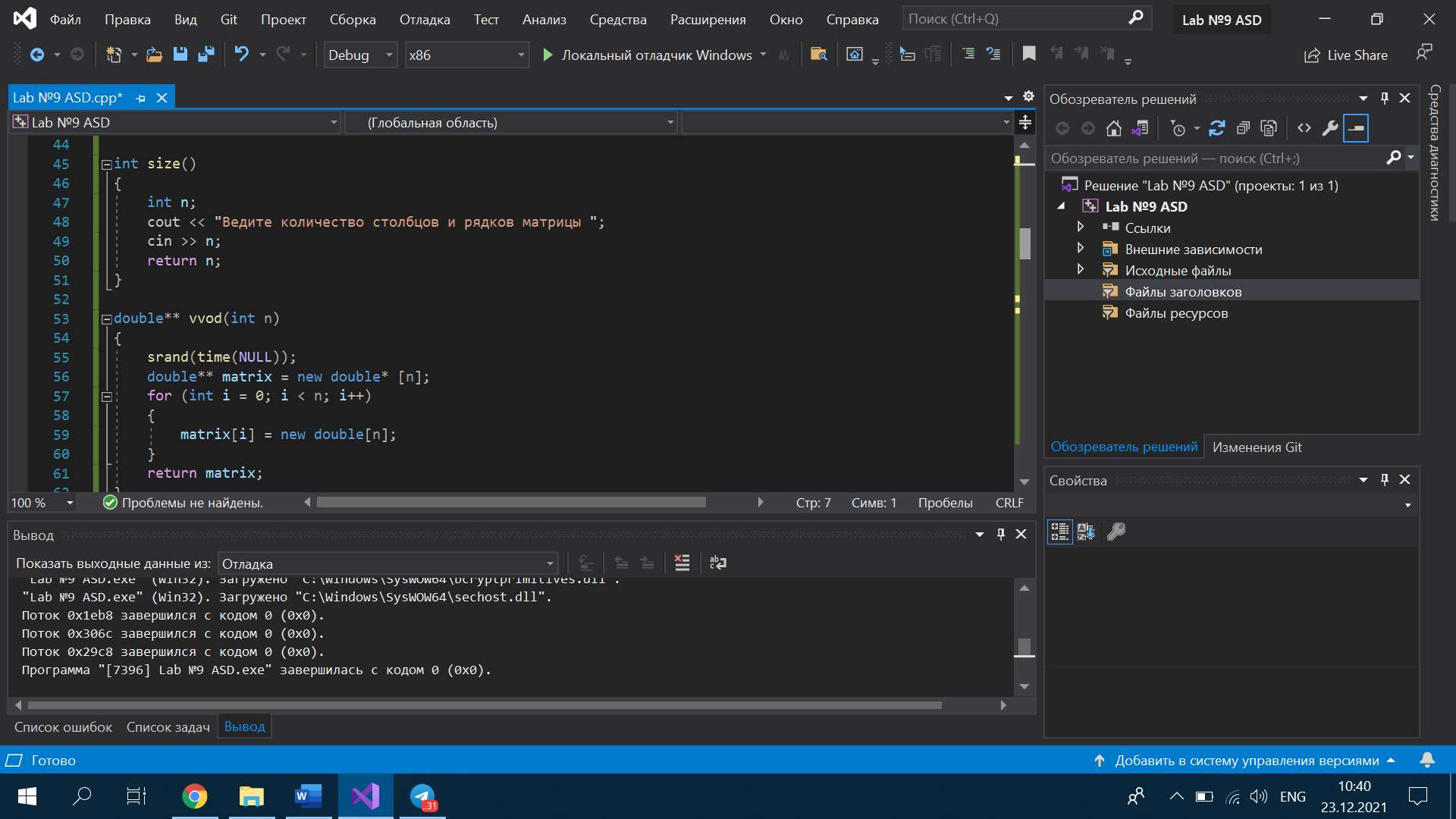


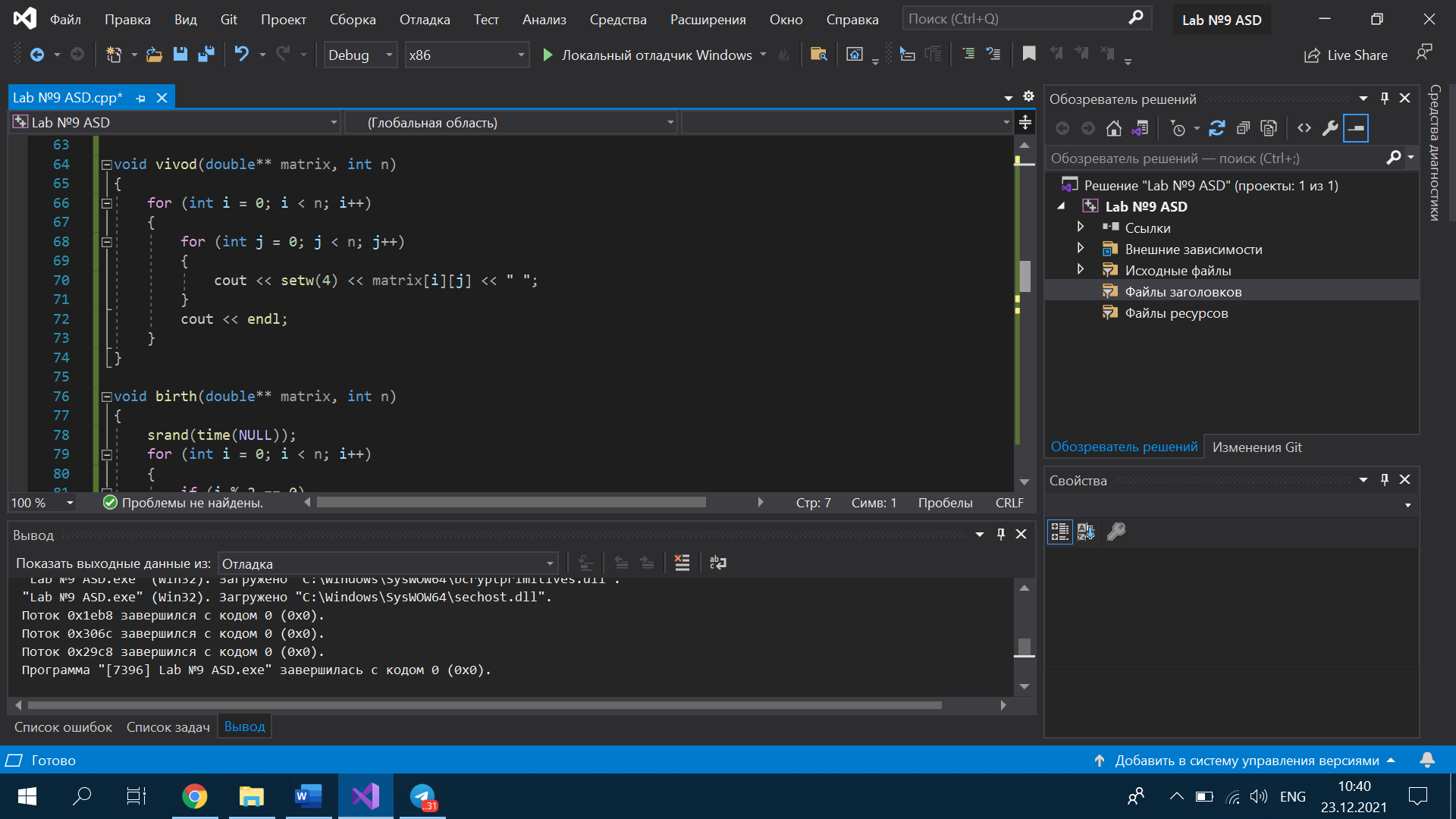


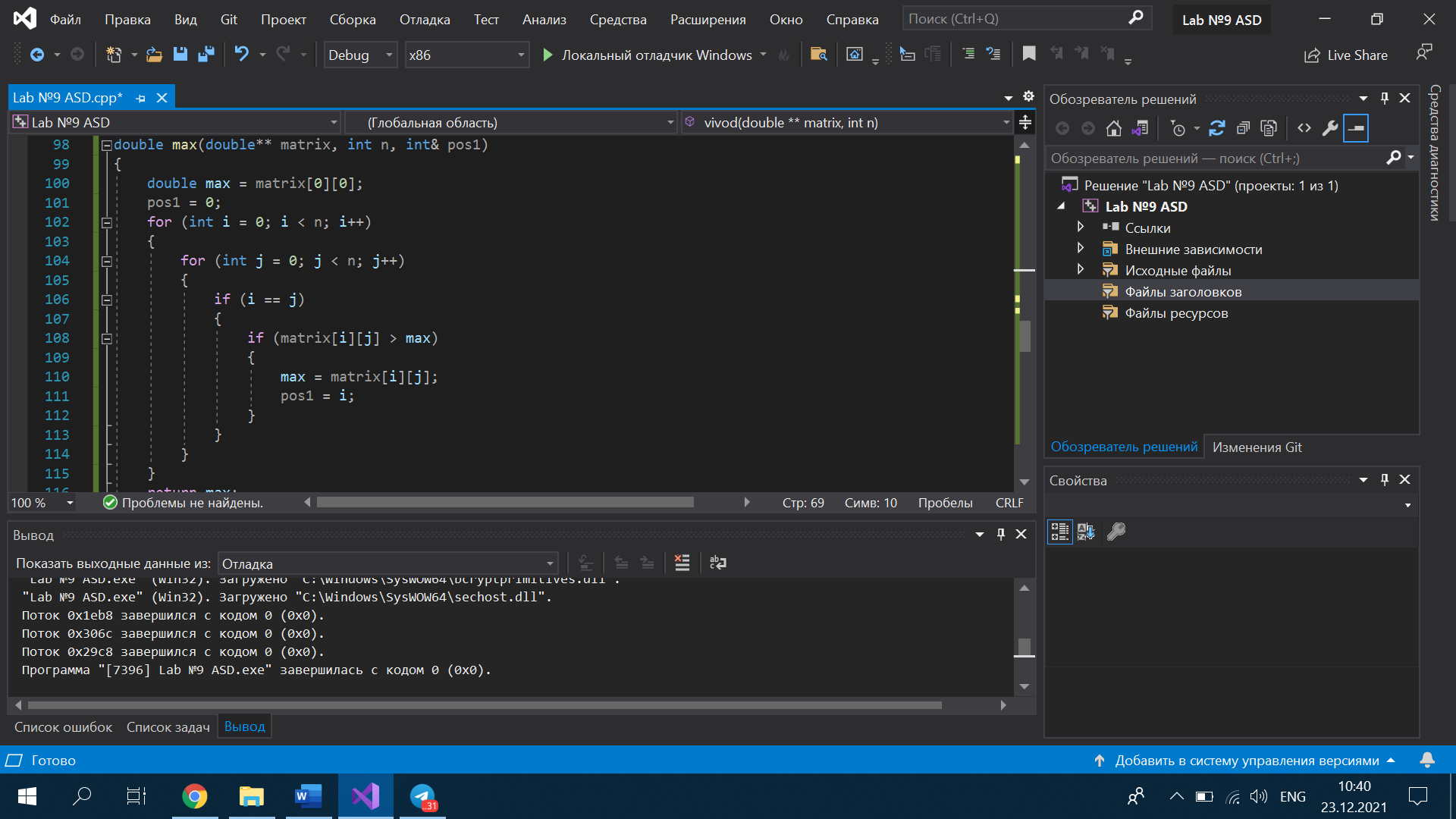
**Код на С++** :

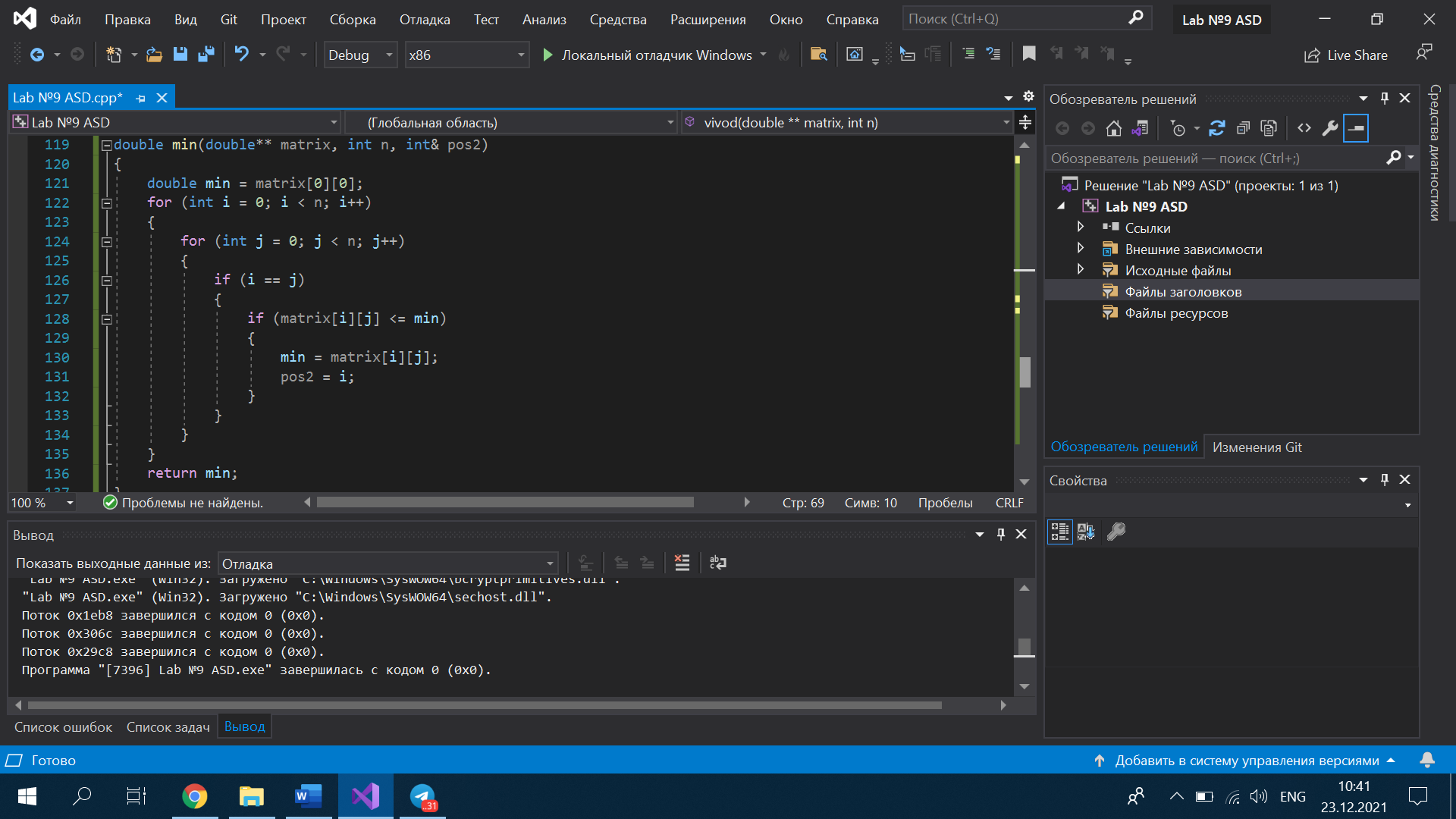


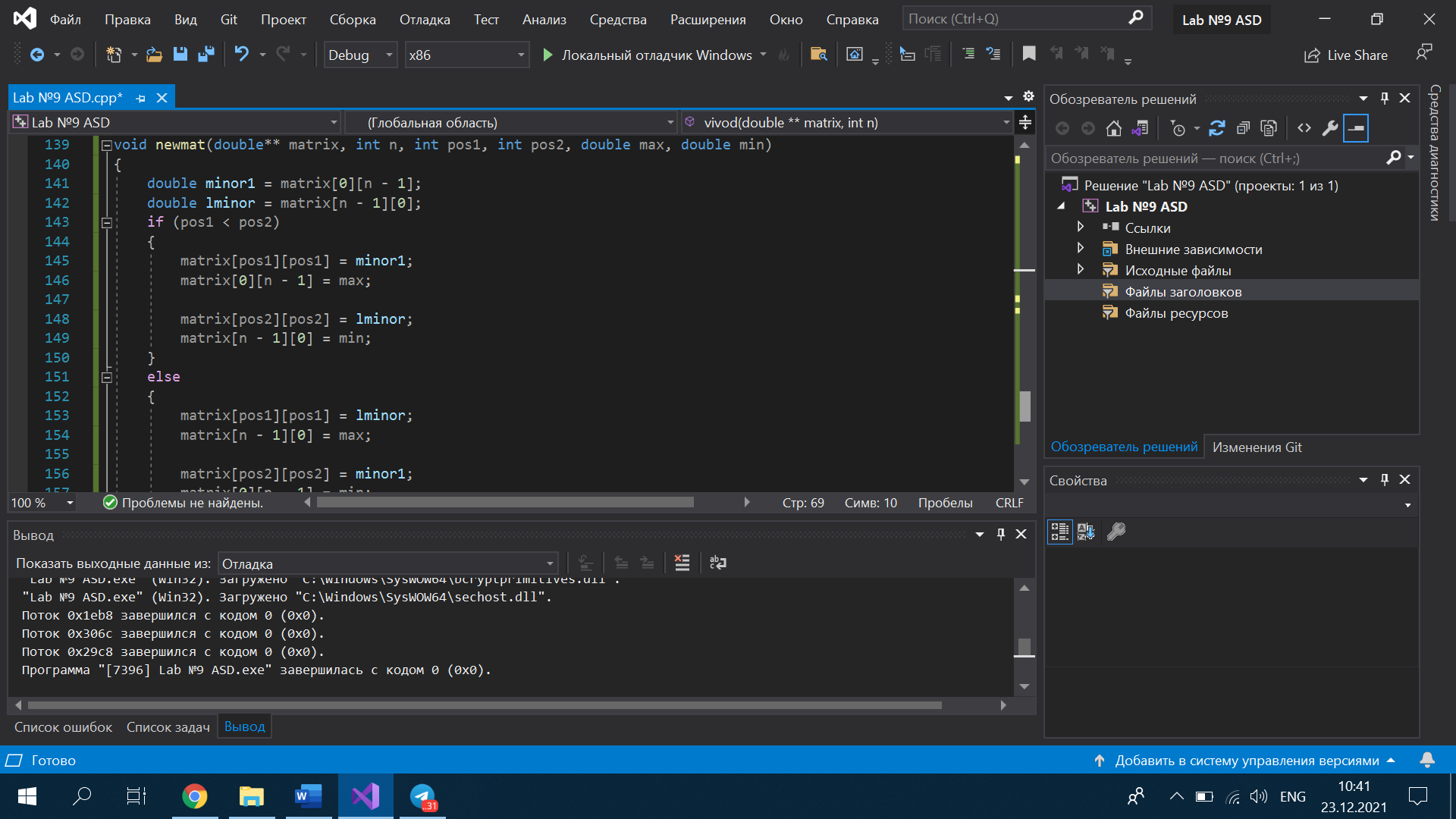


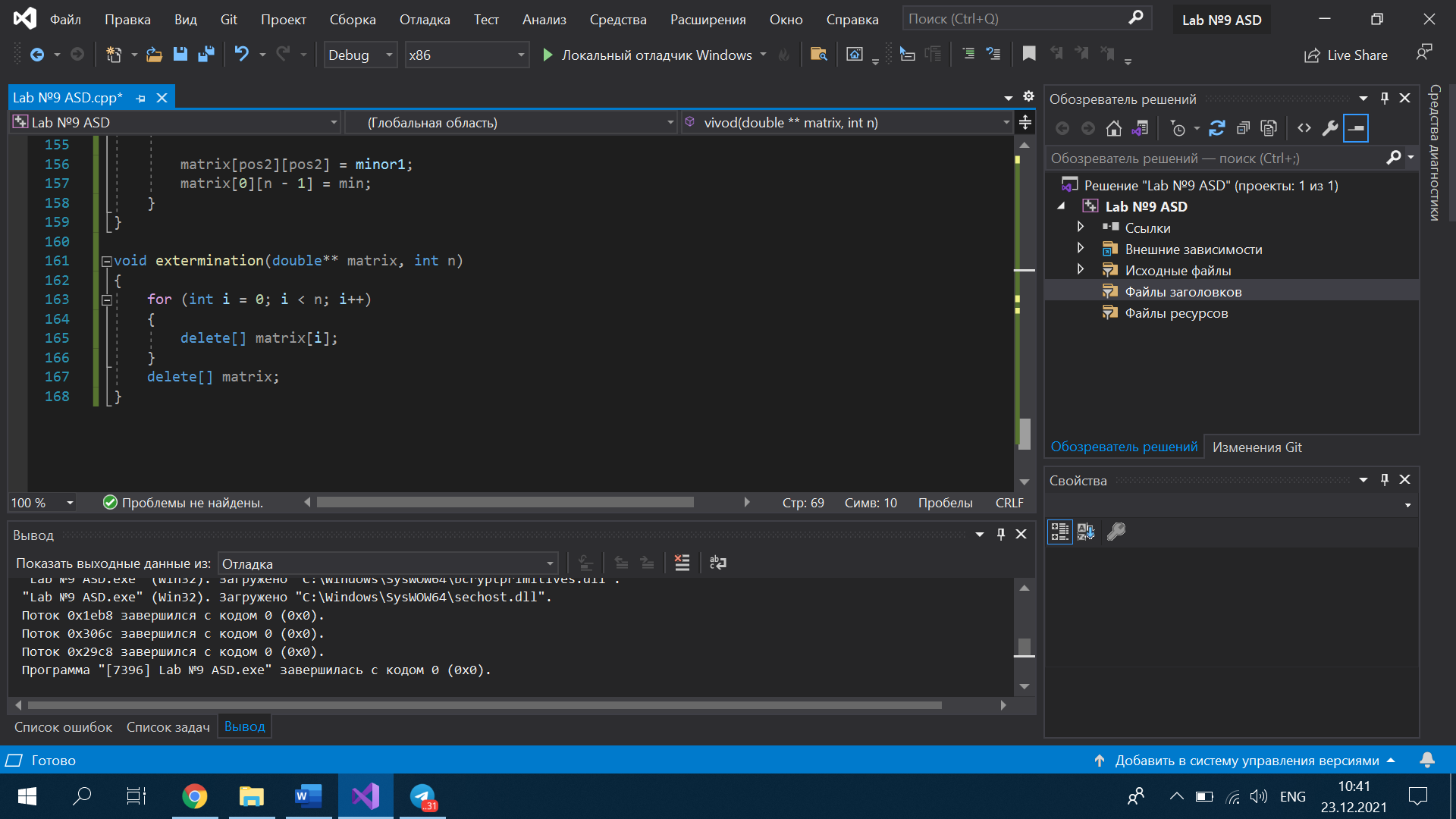


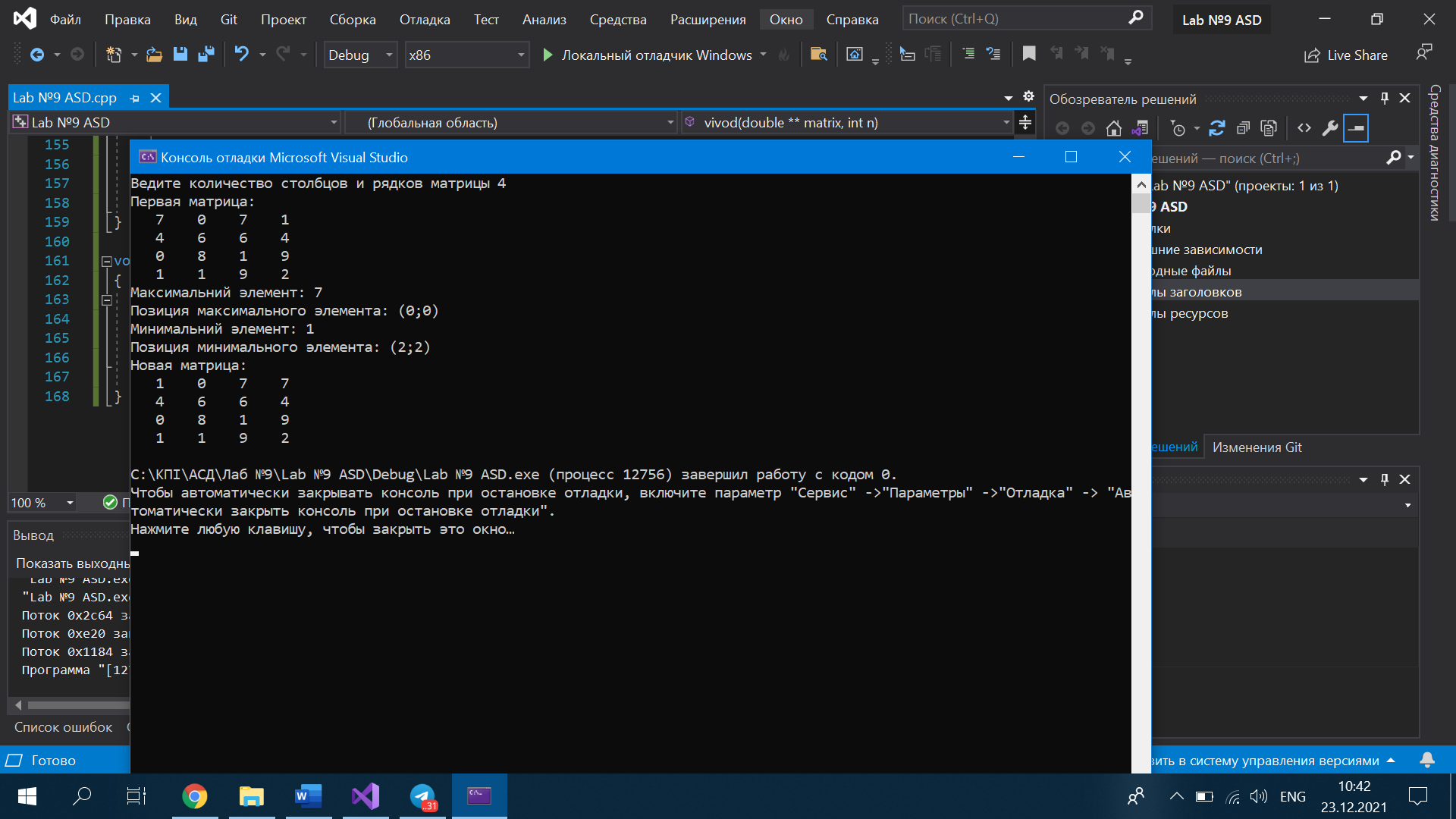












**Висновок:** Під час даної лабораторної роботи були досліджені алгоритми обходу масивів. У даній задачі масив обійшли по рядках(змійкою). Були набуті навички використання їх у програмних специфікаціях. Були побудовані блок схема та математичка модель. Для обчислення виразу була створена елементарна програма. Результати програми виявилися правильними, що стверджує на її дієвість. Завдання було виконано на мові програмування: C++.