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 25

Виконав ІП-15, Плугатирьов Дмитро Валерійович

студент (шифр, прізвище, ім’я, по батькові)

Перевірив Вєчерковська Анастасія Сергіївна

(прізвище, ім’я, по батькові)

Київ 2021

**Лабораторна робота 9**

**Дослідження алгоритмів обходу масивів**

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

**Завдання**

Розробити алгоритм та написати програму, яка складається з наступних

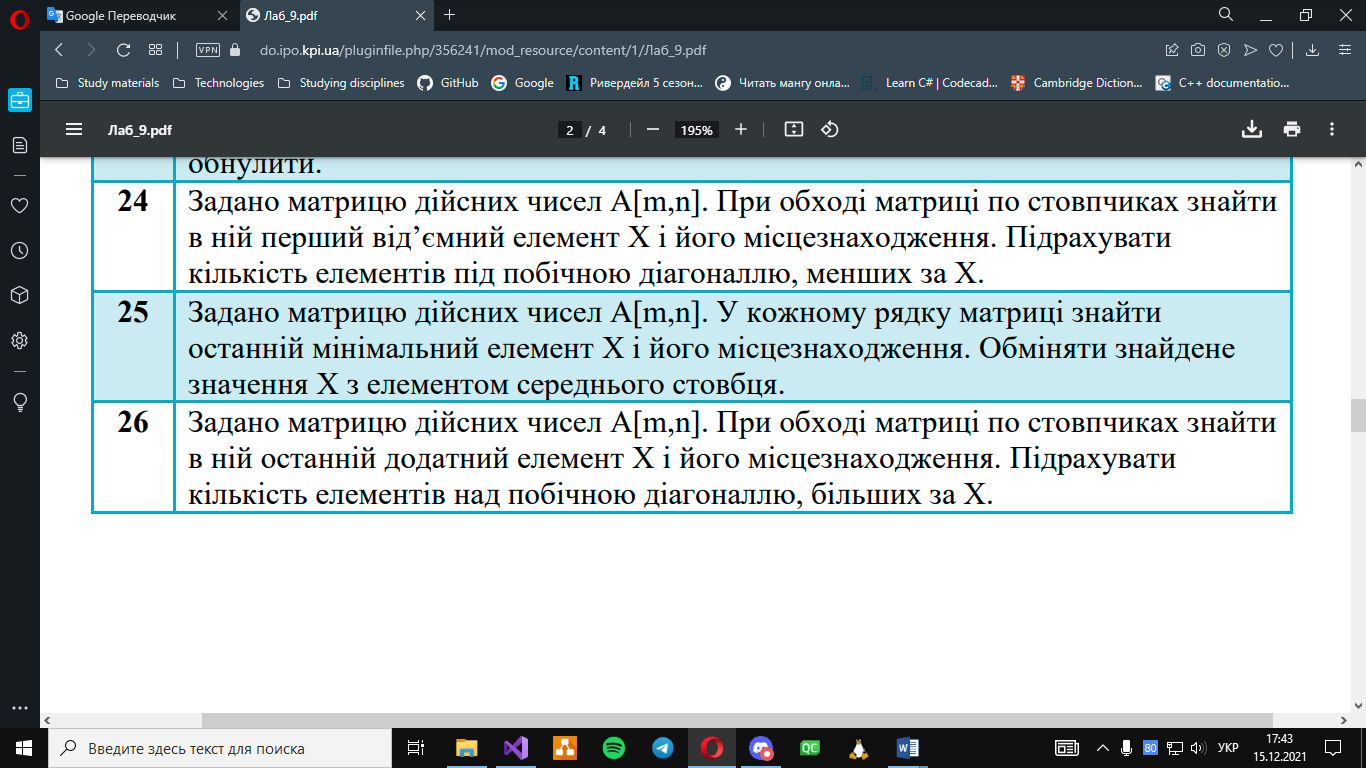
дій:

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

(табл. 1).

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

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



**1. П о с т а н о в к а з а д а ч і**

Створити матрицю дійсних чисел. В кожному рядку матриці знайти останній мінімальний елемент, його індекс та обміняти першозгаданого з іншим із середнього стовпця.

**2. П о б у д о в а м а т е м а т и ч н о ї м о д е л і**

|  |  |  |  |
| --- | --- | --- | --- |
| *Змінна* | *Тип* | *Ім’я* | *Призначення* |
| Матриця | Дійсний | matrix[row][col] | Початкові дані, результат |
| Рядки матриці | Цілочисельний | row | Початкові дані |
| Стовпці матриці | Цілочисельний | col | Початкові дані |
| Мінімальний елемент | Дійсний | min | Проміжні дані |
| Середній елемент | Дійсний | x | Проміжні дані |
| Лічильник циклу | Цілочисельний | i | Проміжні дані |
| Лічильник вкладеного циклу | Цілочисельний | y | Проміжні дані |
| Параметр для визначення випадкового числа | Цілочисельний | temp | Проміжні дані |
| Індекс середнього елемента рядка | Цілочисельний | midI | Проміжні дані |
| Індекс найменшого елемента рядка | Цілочисельний | colI | Проміжні дані |
| Індекс рядка з найменшим елементом | Цілочисельний | rowI | Проміжні дані |
| Пошук елемента, його індексу та обмін із середнім | Підпрограма | find\_elem\_and\_exchange | Проміжні дані |
| Обмін елементу з із середнім | Підпрограма | Exchange\_min\_mid | Проміжні дані |
| Створення матриці та її ініціювання | Підпрограма | create\_and\_init\_matrix | Проміжні дані |

Розмірність матриці визначається користувачем. Прохід по рядкам матриці відбувається з використанням вкладених циклів з умовами на відбір елементу та індексу. Мінімальний елемент та його індекс заносяться у змінні-утримувачі. Виклик підпрограми обмінює середній елемент рядка місцями зі знайденим.

За парної кількості стовпців матриці середнім елементом вважається крайній справа.

Дія x := rand() % num1 – num2 означає присвоєння згенерованого псевдовипадковим чином цілого числа до змінної x в межах від num1 – num2 - 1 до –num2.

**3. Р о з в ' я з а н н я**

Програмні специфікації записати у псевдокоді та графічній формі у вигляді блок-схеми.

*Крок 1*. Визначити основні дії.

*Крок 2*. Визначення розмірності та ініціалізація матриці.

*Крок 3*. Знаходження мінімальних елементів, їх індексів та обмін між із середніми в рядках.

**4. П с е в д о к о д**

**О с н о в н а п р о г р а м а:**

*Крок 1*

**початок**

визначення розмірності та ініціалізація матриці

знаходження мінімальних елементів, їх індексів та обмін між із середніми в рядках

**кінець**

*Крок 2*

**початок**

**ввести** row

**ввести** col

**create\_and\_init\_matrix(matrix, row, col)**

знаходження мінімальних елементів, їх індексів та обмін між із середніми в рядках

**кінець**

*Крок 3*

**початок**

**ввести** row

**ввести** col

**create\_and\_init\_matrix(matrix, row, col)**

**find\_elem\_and\_exchange(matrix, row, col, x)**

**кінець**

**П і д п р о г р а м и:**

**create\_and\_init\_matrix(matrix, row, col)**

**повторити для** i **від** 0 **до** row

**повторити для** y **від** 0 **до** col

temp := rand() % 3

**якщо** temp == 2

**то**

matrix[i][y] := rand() % 1000 \* 2 – 1000

**все якщо**

**інакше якщо** temp == 1

**то**

matrix[i][y] := (rand() % 1000 \* 2 - 1000) / 12.4

**все якщо**

**інакше**

matrix[i][y] := 0

**все якщо**

**все повторити**

**все повторити**

**кінець**

**find\_elem\_and\_exchange(matrix, row, col, x)**

midI := row / 2

**повторити для** i **від** 0 **до** row

rowI := i

x := matrix[i][0]

**повторити для** y **від** 0 **до** col

**якщо** x >= matrix[i][y]

**то**

x := matrix[i][y]

colI := y

**все якщо**

**якщо** colI != row / 2 **або** colI != row / 2 + 1

**то**

**Exchange\_min\_mid(matrix, rowI, colI, x, midI)**

**все якщо**

**все повторити**

**все повторити**

**кінець**

**Exchange\_min\_mid(matrix, rowI, colI, x, midI)**

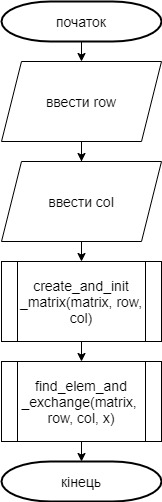
matrix[rowI][colI] := matrix[rowI][midI]

matrix[rowI][midI] := x

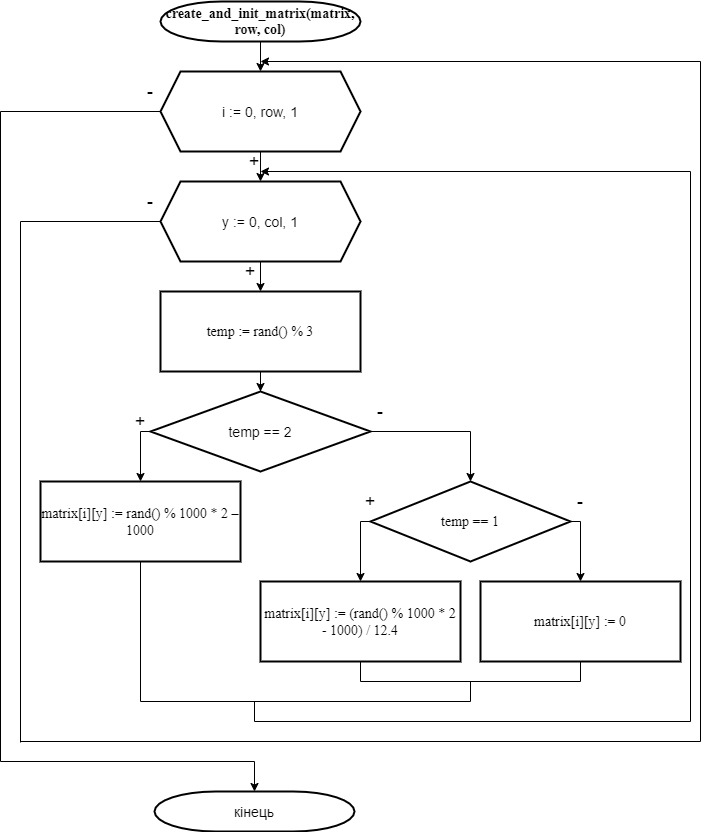
**кінець**

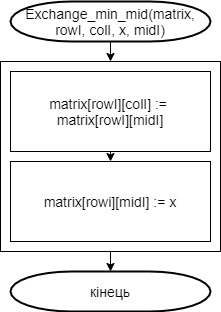
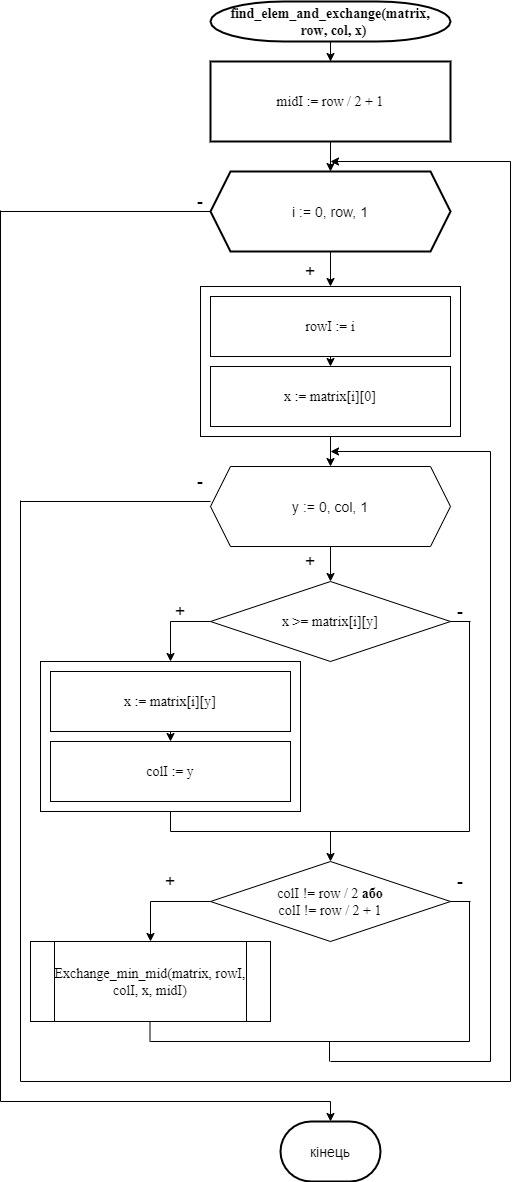
*Блок-схема*

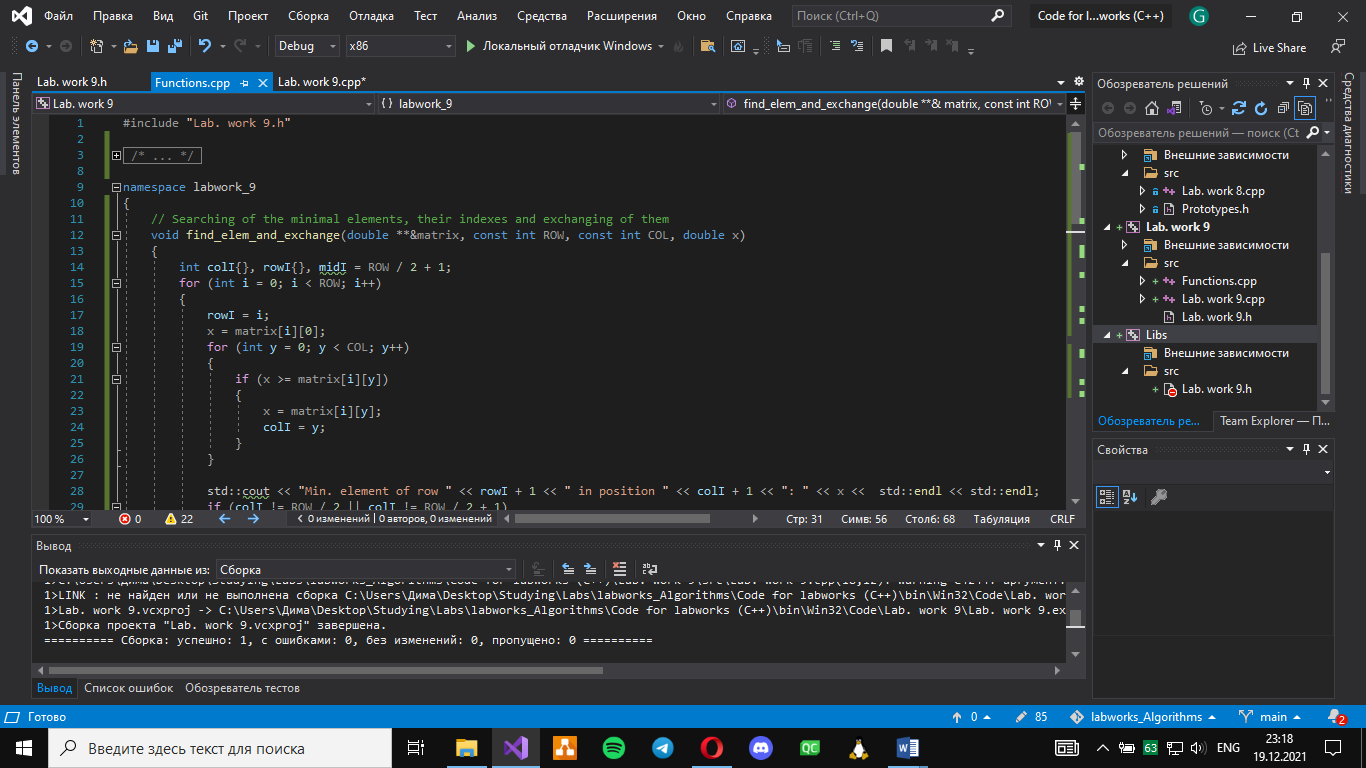
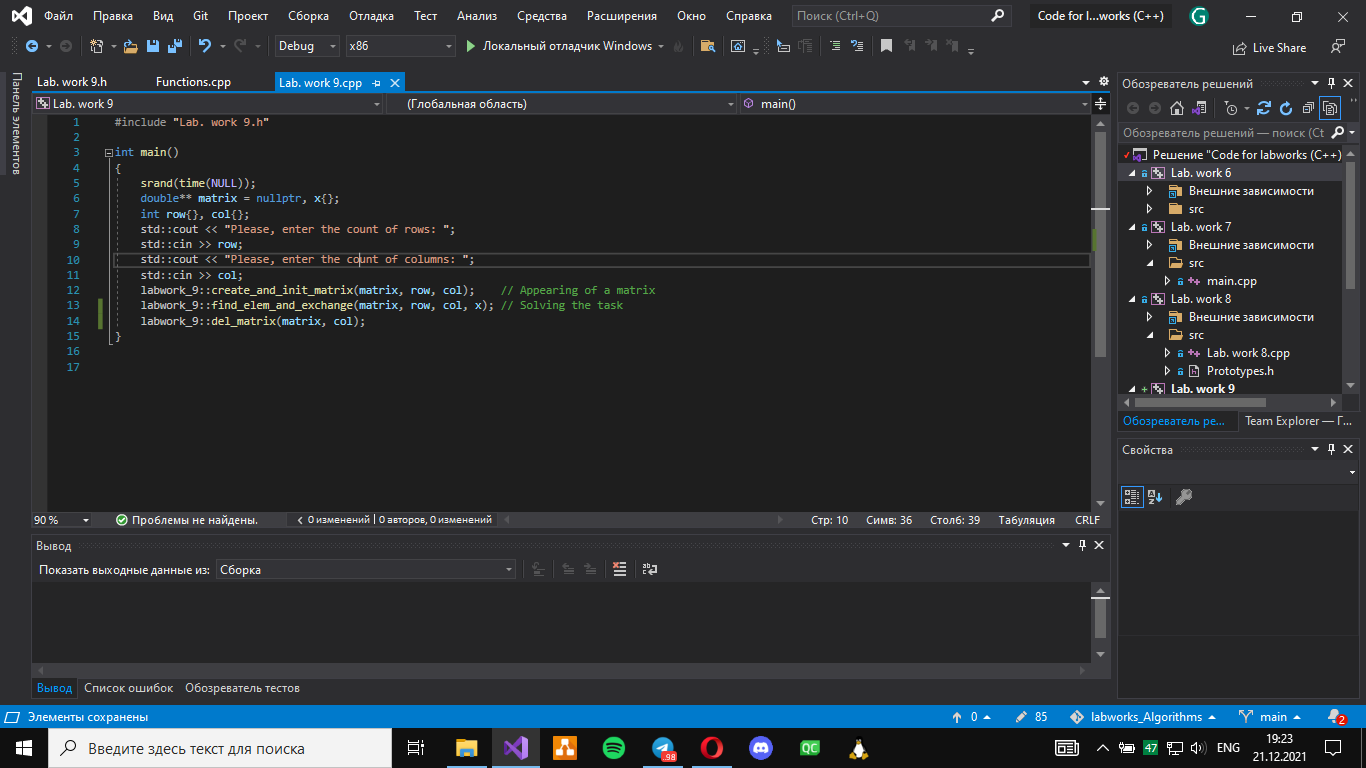
**О с н о в н а п р о г р а м а:**

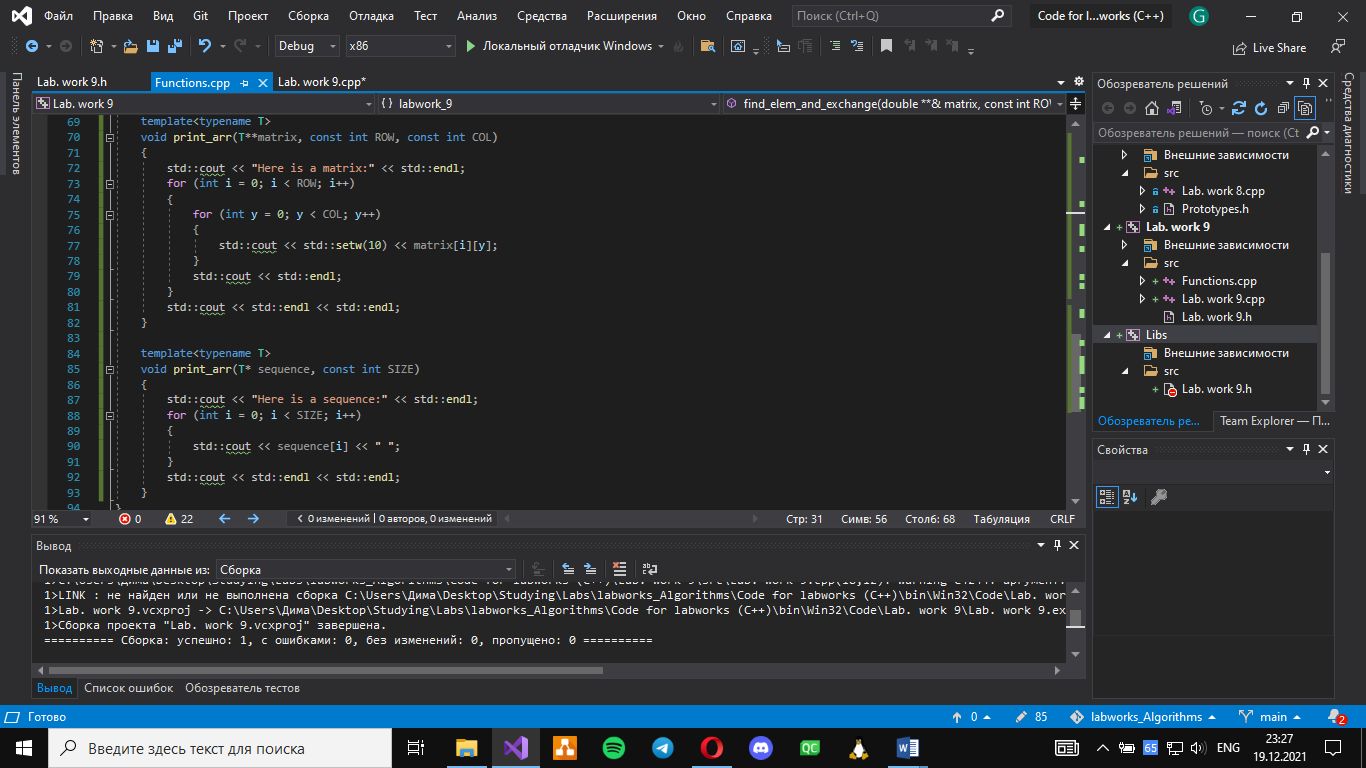


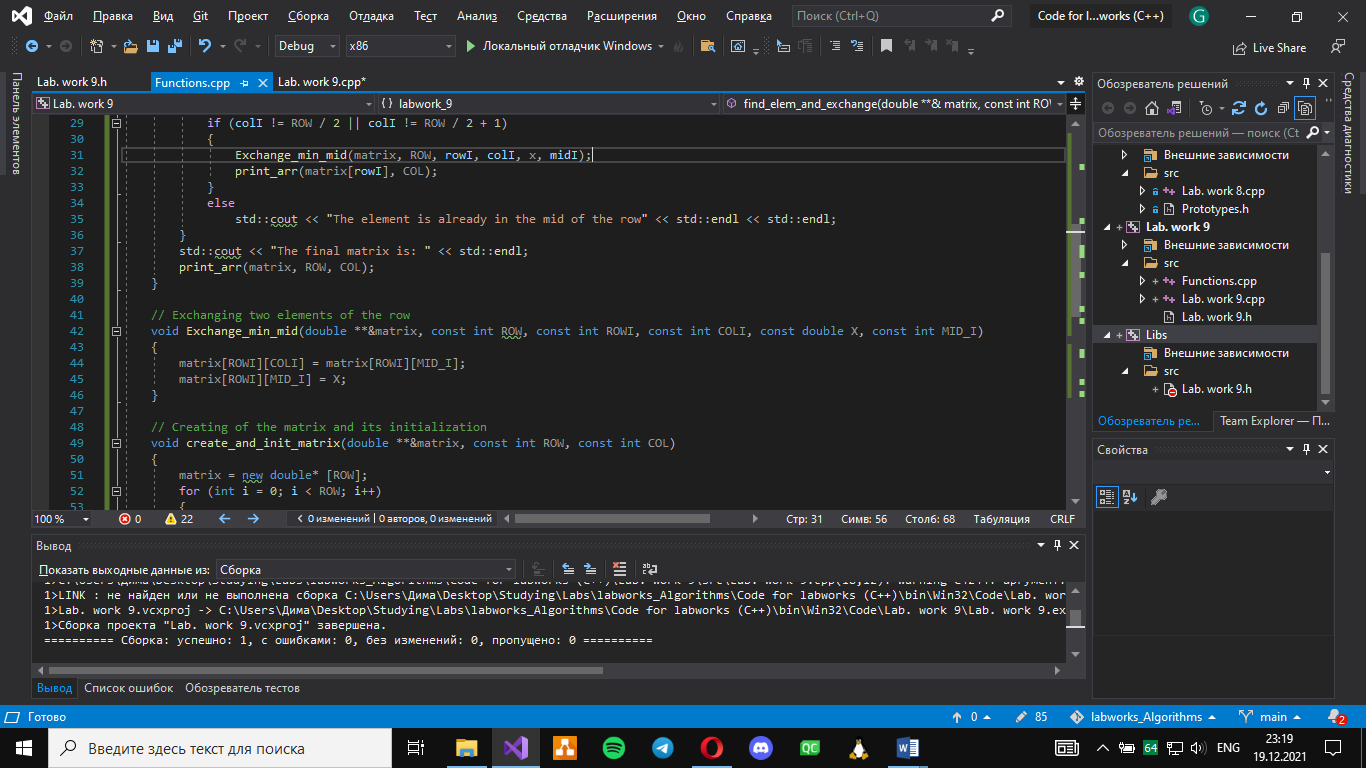
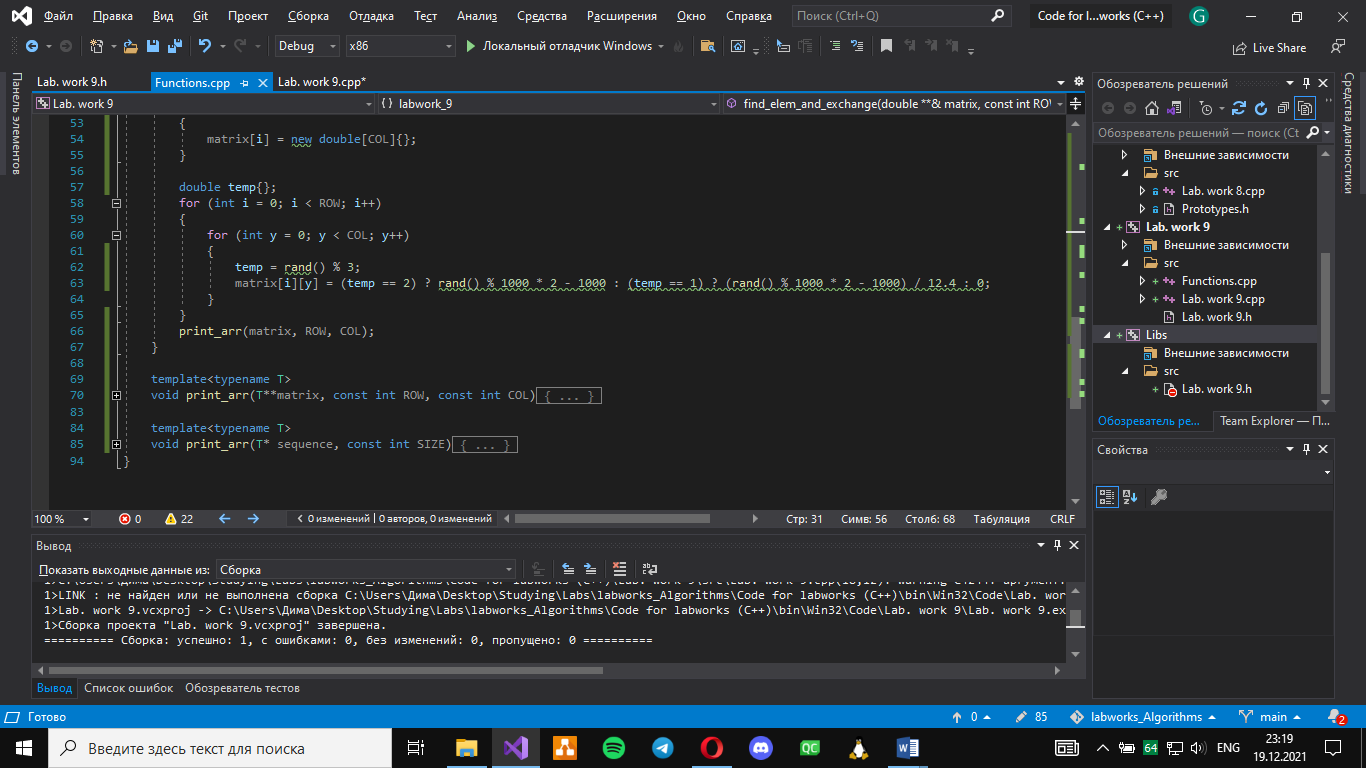
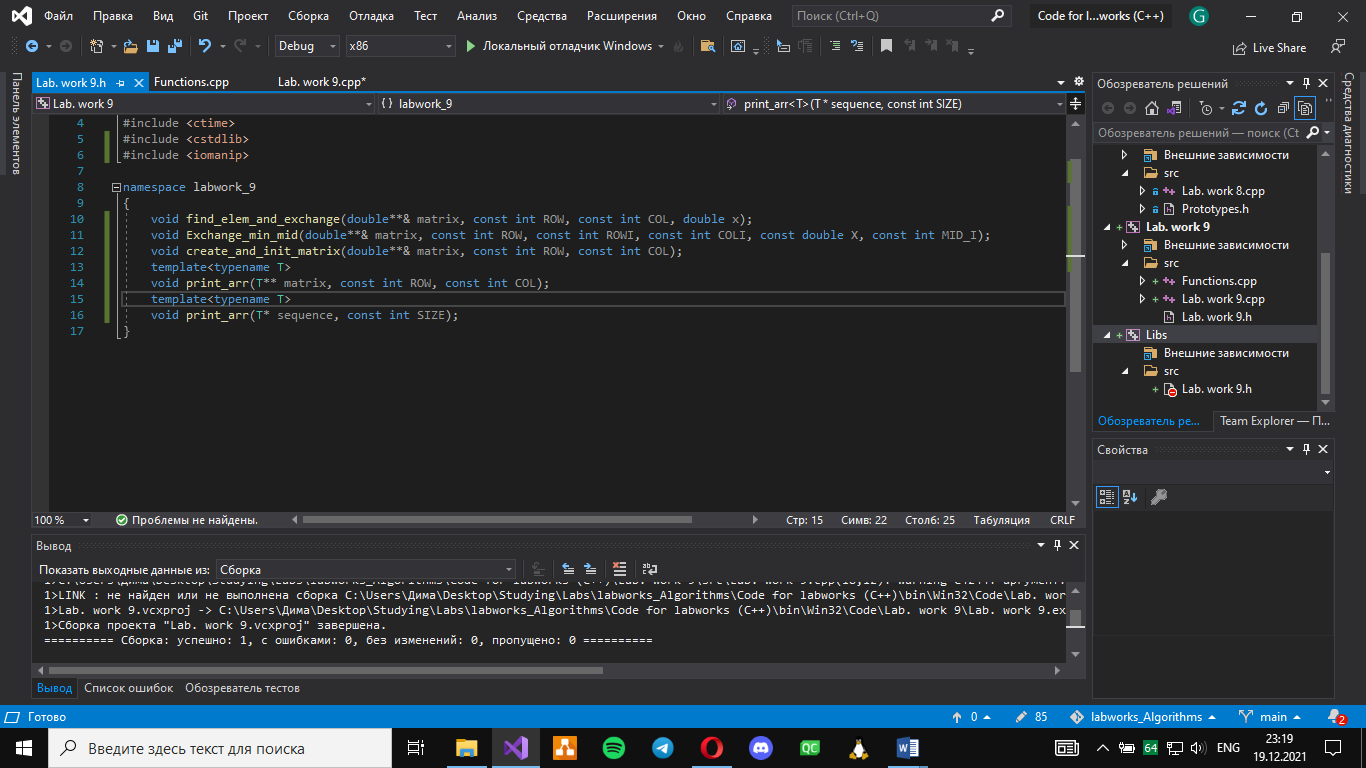
**П і д п р о г р а м и:**

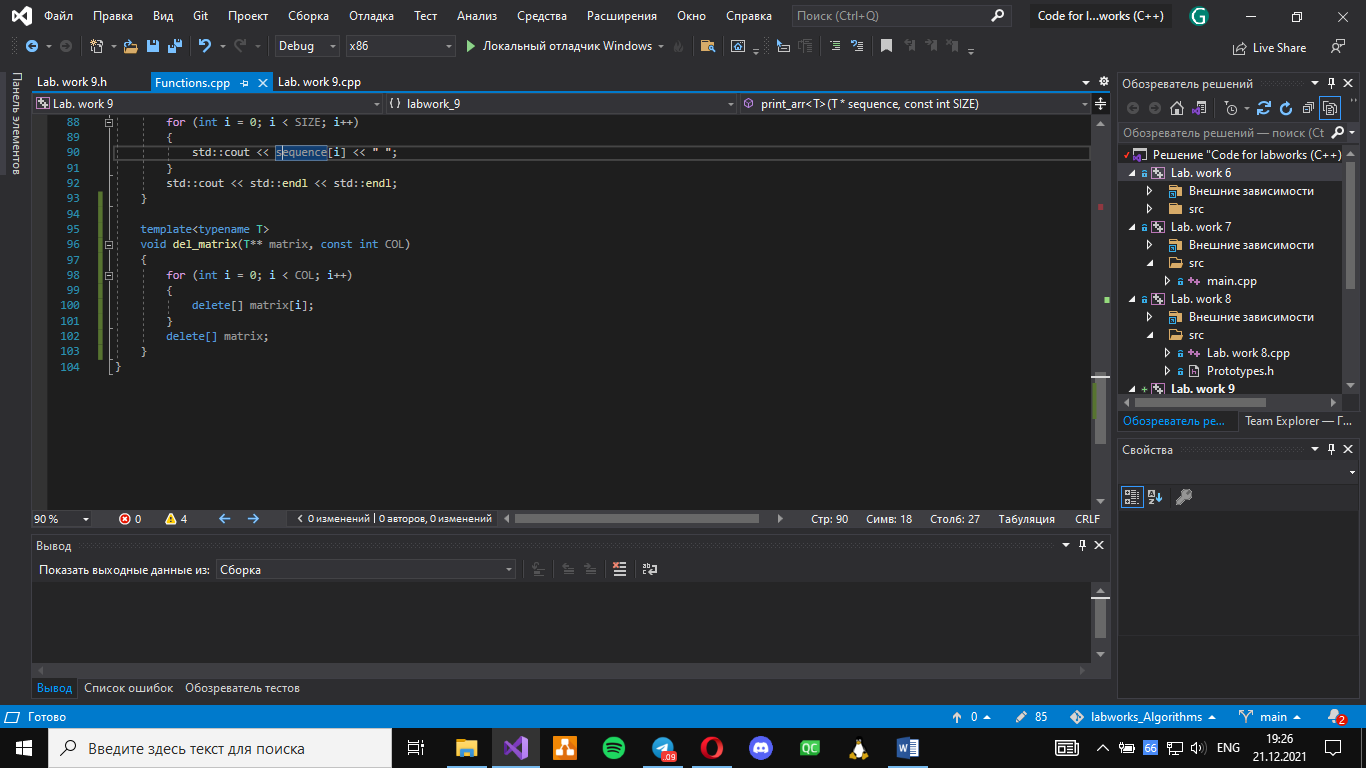




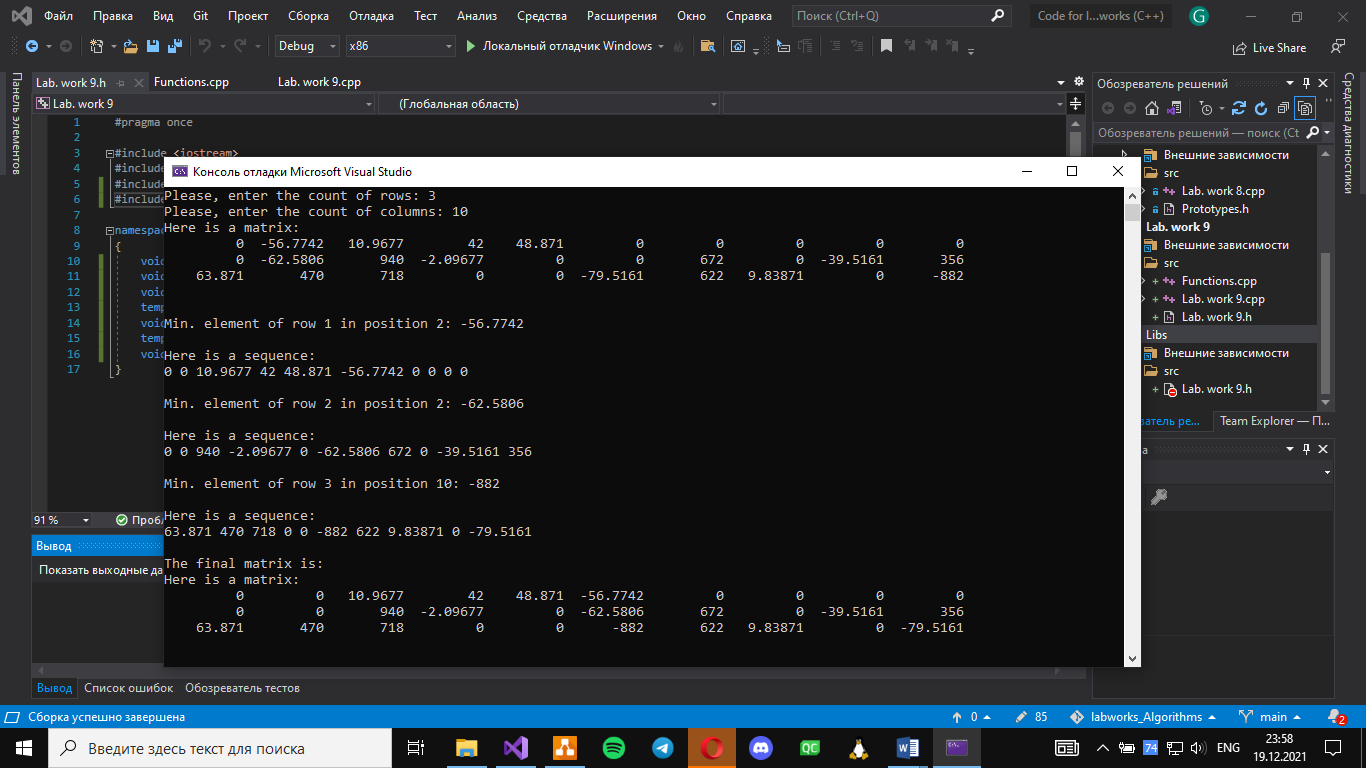
**5. К о д п р о г р а м и**







**Т е с т у в а н н я**



**6. В и с н о в о к**

На цій лабораторній роботі я дослідив алгоритми обходу масивів та набув практичних навичок використання цих алгоритмів під час складання програмних специфікацій. Мені довелося працювати з елементами матриці: використання пошуку, порівнянь та перестановок.