Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний

інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженері

Звіт

з лабораторної роботи № 9 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

Дослідження алгоритмів обходу масивів

Варіант 31

Виконав студент ІП-12 Титаренко Данило Олегович

Перевірив \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( 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.

**Задача 31**

Задано матрицю дійсних чисел A[n,n]. При обході матриці по стовпчиках знайти в ній перший мінімальний елемент Х і його місцезнаходження. Обміняти знайдене значення Х з елементом над головною діагоналлю, більших за середньоарифметичне значення.

**Розв’язок**

1. **Постановка задачі.**

За допомогою вкладених ітераційних циклів робимо алгоритм обходу матриці по стовпчиках, а також знаходимо мінімальний елемент та його місцезнаходження. Потім знаходимо середнє арифметичне елементів над головною діагоналлю і в кінці обмінюємо мінімальний елемент з елементом над головною діагоналлю, який більший за середнє арифметичне. Результатом розв’язку буде масив дійсних чисел.

1. **Математична модель.** Складемо таблицю імен змінних

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Двовимірний масив | Дійсний | A | Початкове дане |
| К-ть рядків та стовпчиків | Цілий | n | Початкове дане |
| Мінімальний елемент | Дійсний | min | Результат |
| Рядок мінімального елемента | Цілий | B | Результат |
| Стовпчик Мінімального елемента | Цілий | C | Результат |
| Лічильник 1 | Цілий | I | Проміжне дане |
| Лічильник 2 | Цілий | J | Проміжне дане |
| Лічильник 3 | Цілий | K | Проміжне дане |
| Середнє арифметичне | Дійсний | S | Проміжне дане |
| Прапорець 1 | Логічний | F | Проміжне дане |
| Прапорець 2 | Цілий | Dir | Проміжне дане |

Таким чином математичне формулювання задачі зводиться до ініціалізування двовимірного масиву за допомогою вкладених ітераційних циклів та функції rand(). Застосування алгоритму для обходу матриці по стовпчиках, а також виведення двовимірного масиву, мінімального елементу та його місцезнаходження.

1. **Розв’язок**

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

*Крок 1.* Визначимо основні дії

*Крок 2*. Ініціалізуємо елементи двовимірного масиву

Крок 3. Деталізуємо дію обходу матриці по стовпчиках

Крок 4. Деталізуємо дію знаходження мінімального елементу та його місцезнаходження.

Крок 5. Деталізуємо дію знаходження середнього арифметичного над головною діагоналлю

Крок 6. Деталізуємо дію обміну елементів та вивід масиву

1. **Псевдокод**

Крок 1

**Початок**

Ініціалізуємо елементи двовимірного масиву

Деталізуємо дію обходу матриці по стовпчиках

Деталізуємо дію знаходження мінімального елементу та його місцезнаходження.

Деталізуємо дію знаходження середнього арифметичного над головною діагоналлю

Деталізуємо дію обміну елементів та вивід масиву

**Кінець**

Крок 2

**Початок**

**повторити**

**для і від 0 до n**

**повторити**

**для j від 0 до n**

a[i][j]=rand()%20+1;

**все повторити**

**все повторити**

Деталізуємо дію обходу матриці по стовпчиках

Деталізуємо дію знаходження мінімального елементу та його місцезнаходження.

Деталізуємо дію знаходження середнього арифметичного над головною діагоналлю

Деталізуємо дію обміну елементів та вивід масиву

**Кінець**

Крок 3

**Початок**

**повторити**

**для і від 0 до n**

**повторити**

**для j від 0 до n**

a[i][j]=rand()%20+1;

**все повторити**

**все повторити**

min=a[0][0]

dir=1

**повторити**

**для j від 0 до n**

**якщо(dir<0)**

**то**

**повторити**

**для і від n-1 до 0**

Деталізуємо дію знаходження мінімального елементу та його місцехнаходження

**все повторити**

**інакше**

**повторити**

**для і від 0 до n**

Деталізуємо дію знаходження мінімального елементу та його місцехнаходження

**все повторити**

**все якщо**

dir=-dir

**все повторити**

Деталізуємо дію знаходження середнього арифметичного над головною діагоналлю

Деталізуємо дію обміну елементів та вивід масиву

**Кінець**

Крок 4

**Початок**

**повторити**

**для і від 0 до n**

**повторити**

**для j від 0 до n**

a[i][j]=rand()%20+1;

**все повторити**

**все повторити**

min=a[0][0]

dir=1

**повторити**

**для j від 0 до n**

**якщо(dir<0)**

**то**

**повторити**

**для і від n-1 до 0**

**якщо** **(a[i][j] < min)**

**то**

min = a[i][j];

b = i;

c = j;

**все якщо**

**все повторити**

**інакше**

**повторити**

**для і від 0 до n**

**якщо** **(a[i][j] < min)**

**то**

min = a[i][j];

b = i;

c = j;

**все якщо**

**все повторити**

**все якщо**

dir=-dir

**все повторити**

Деталізуємо дію знаходження середнього арифметичного над головною діагоналлю

Деталізуємо дію обміну елементів та вивід масиву

**Кінець**

Крок 5

**Початок**

**повторити**

**для і від 0 до n**

**повторити**

**для j від 0 до n**

a[i][j]=rand()%20+1;

**все повторити**

**все повторити**

min=a[0][0]

dir=1

**повторити**

**для j від 0 до n**

**якщо(dir<0)**

**то**

**повторити**

**для і від n-1 до 0**

**якщо** **(a[i][j] < min)**

**то**

min = a[i][j];

b = i;

c = j;

**все якщо**

**все повторити**

**інакше**

**повторити**

**для і від 0 до n**

**якщо** **(a[i][j] < min)**

**то**

min = a[i][j];

b = i;

c = j;

**все якщо**

**все повторити**

**все якщо**

dir=-dir

**все повторити**

s=0

k=0

**повторити**

**для і від 0 до n**

**повторити**

**для j від i+1 до n**

s = s + a[i][j];

k = k + 1;

**все повторити**

**все повторити**

s=s/k

Деталізуємо дію обміну елементів та вивід масиву

**Кінець**

Крок 6

**Початок**

**повторити**

**для і від 0 до n**

**повторити**

**для j від 0 до n**

a[i][j]=rand()%20+1;

**все повторити**

**все повторити**

min=a[0][0]

dir=1

**повторити**

**для j від 0 до n**

**якщо(dir<0)**

**то**

**повторити**

**для і від n-1 до 0**

**якщо** **(a[i][j] < min)**

**то**

min = a[i][j];

b = i;

c = j;

**все якщо**

**все повторити**

**інакше**

**повторити**

**для і від 0 до n**

**якщо** **(a[i][j] < min)**

**то**

min = a[i][j];

b = i;

c = j;

**все якщо**

**все повторити**

**все якщо**

dir=-dir

**все повторити**

s=0

k=0

**повторити**

**для і від 0 до n**

**повторити**

**для j від i+1 до n**

s = s + a[i][j];

k = k + 1;

**все повторити**

**все повторити**

s=s/k

f=true

**повторити**

**для і від 0 до n**

**повторити**

**для j від i+1 до n**

**якщо** (a[i][j] > s)

**то**

**якщо** f

**то**

a[b][c] = a[i][j];

a[i][j] = min;

f = false;

**все якщо**

**все якщо**

**все повторити**

**все повторити**

**повторити**

**для і від 0 до n**

**повторити**

**для j від 0 до n**

**Вивід a[i][j]**

**все повторити**

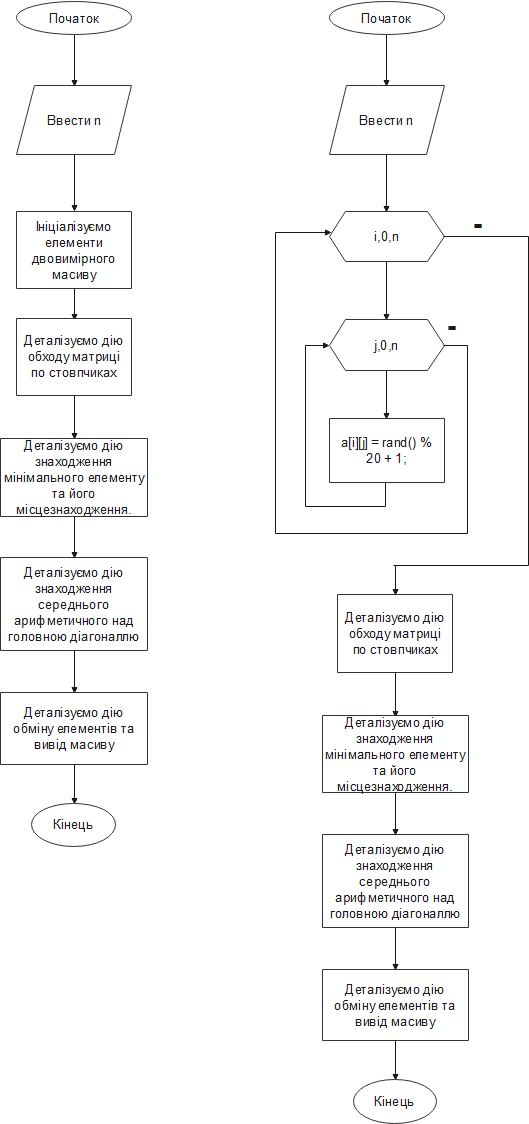
**все повторити**

**Вивід min, b,c**

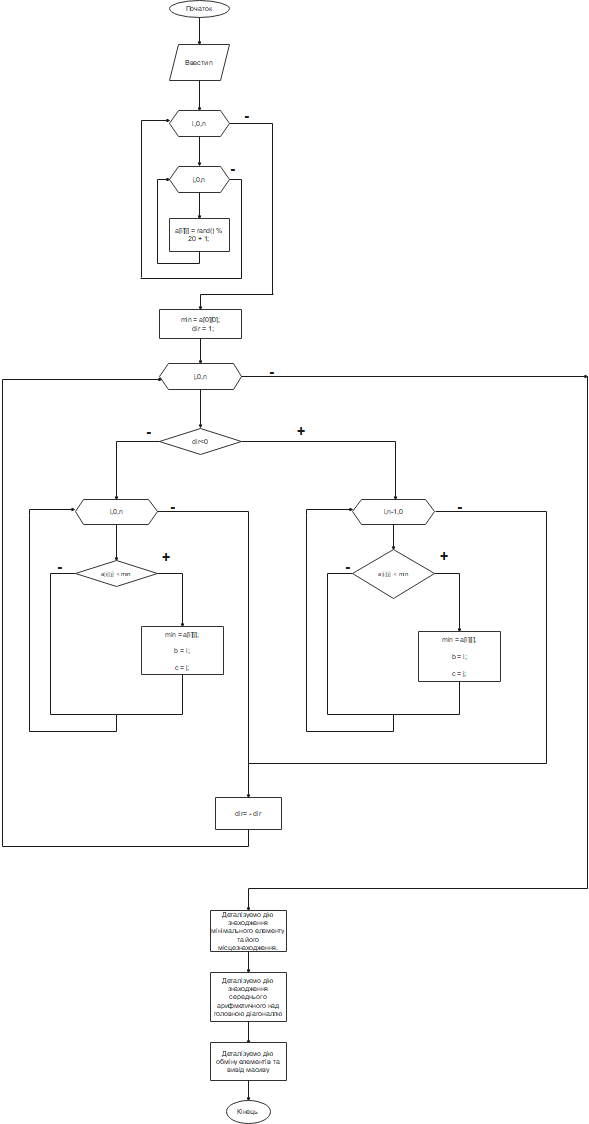
**Кінець**

1. **Блок-схема**

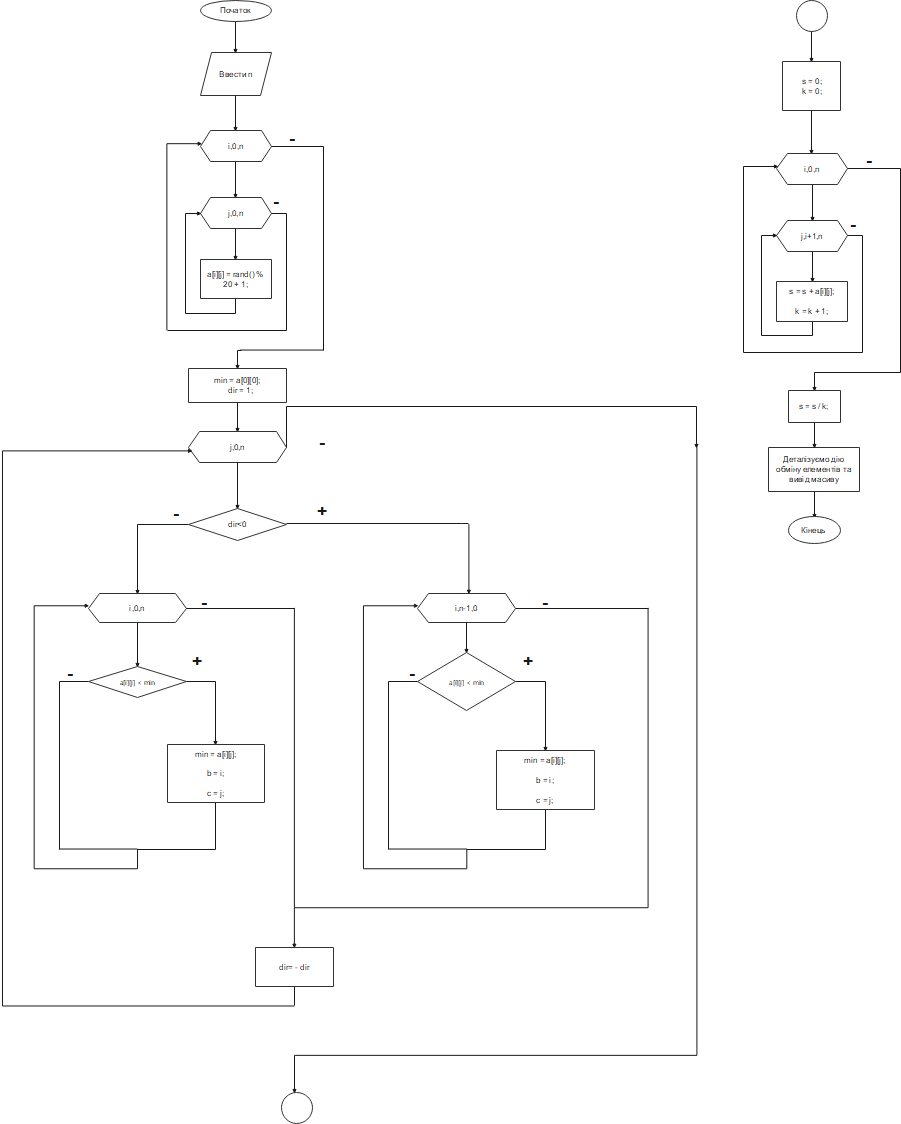
**Крок 1 Крок 2**



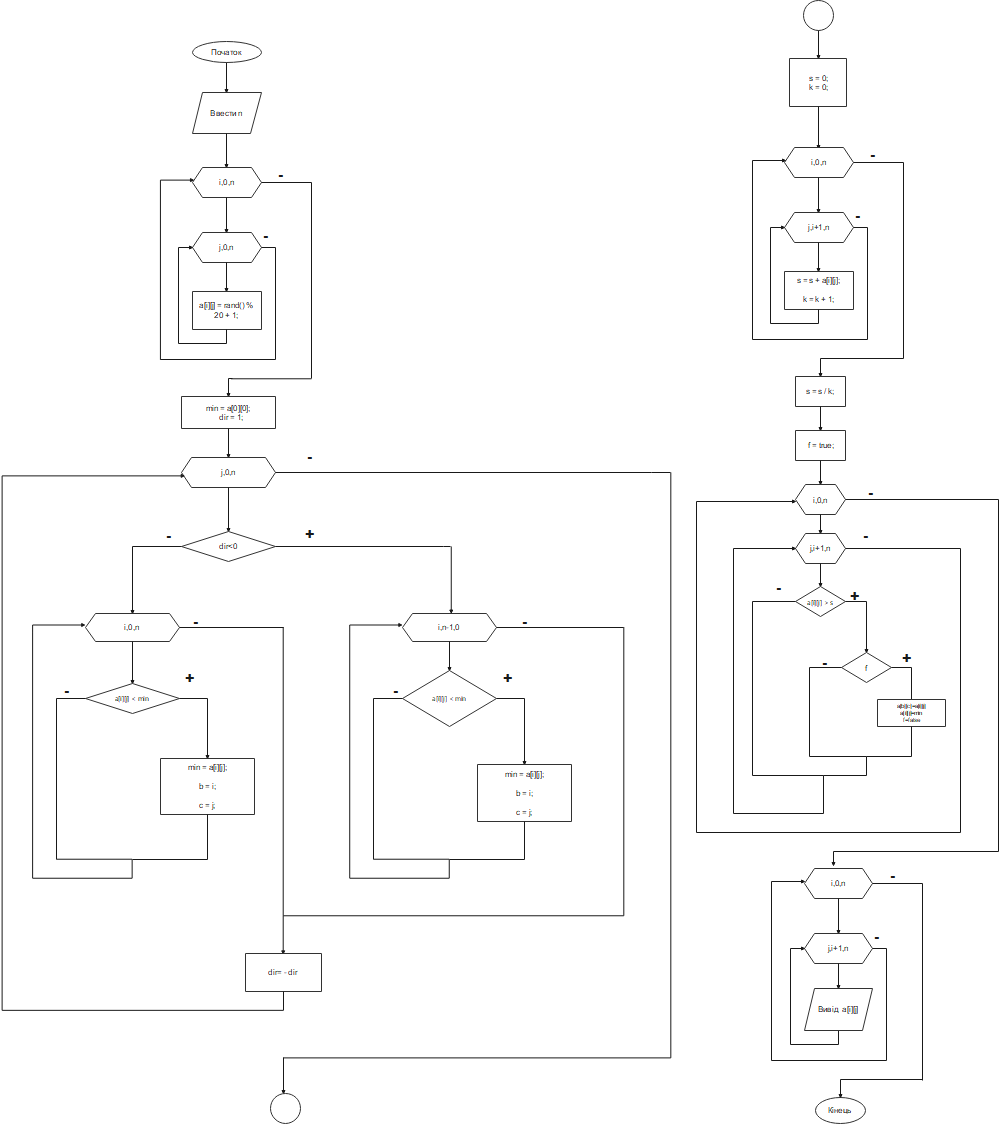
**Крок 3**



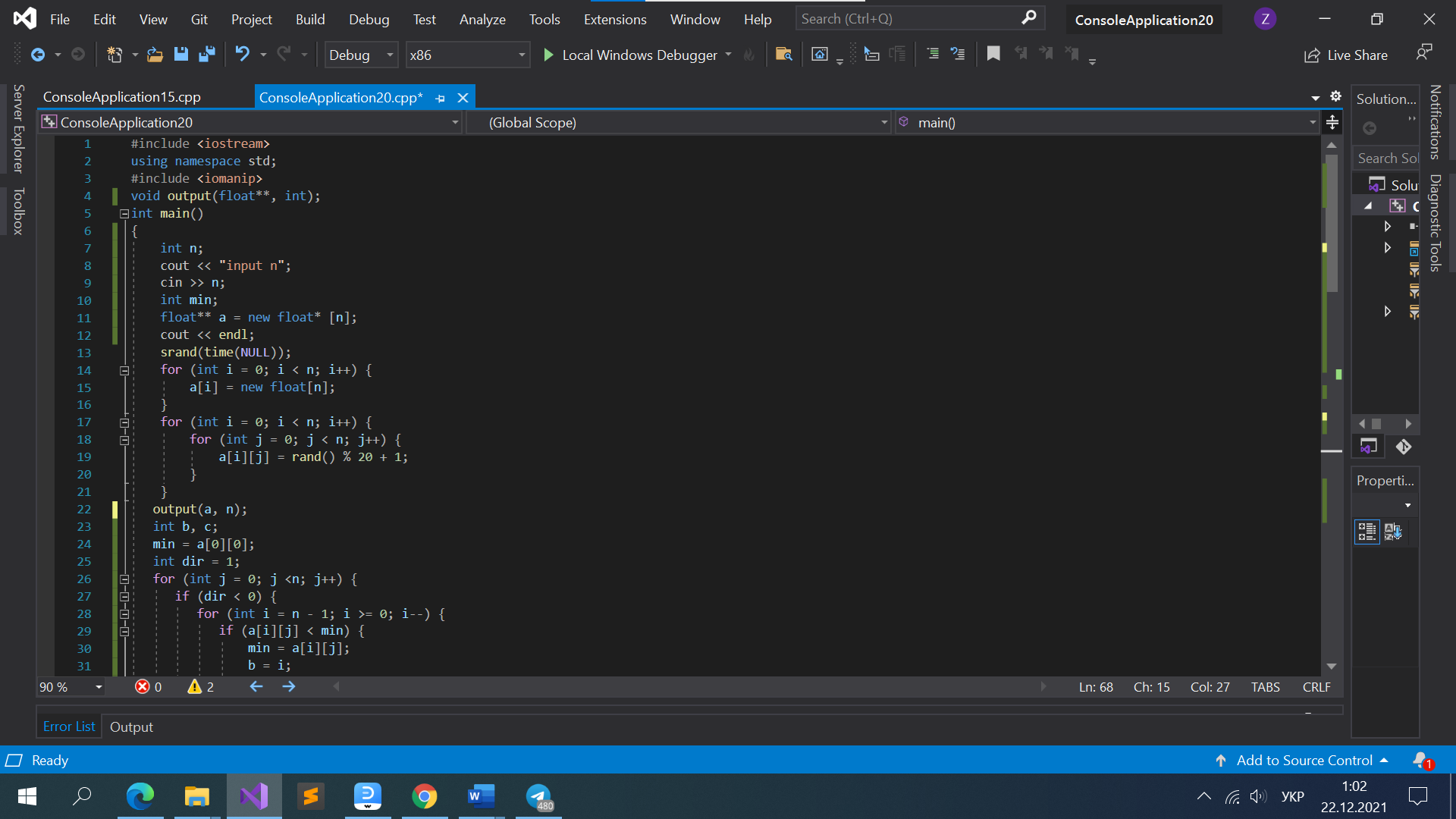
**Крок 4**

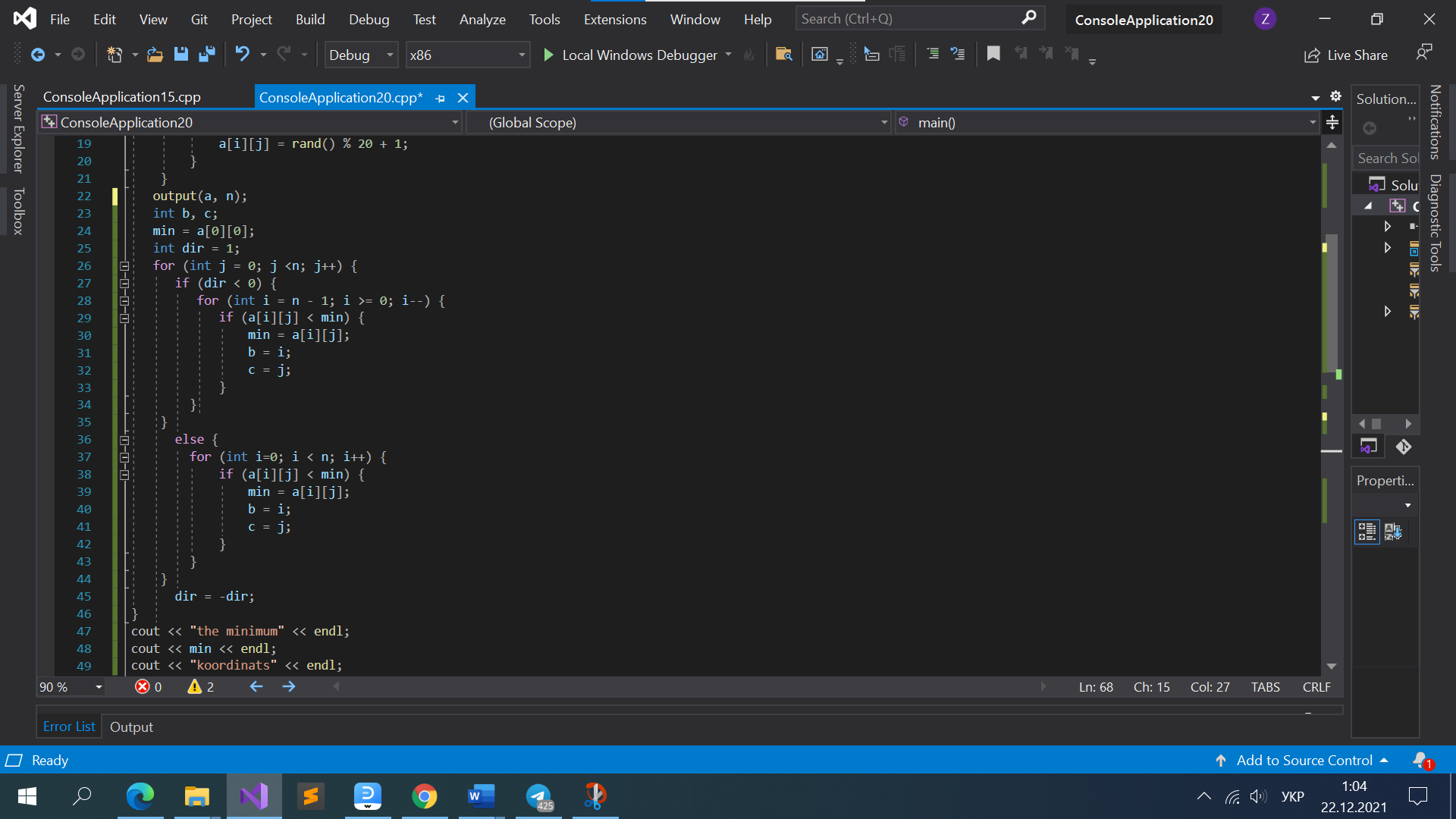


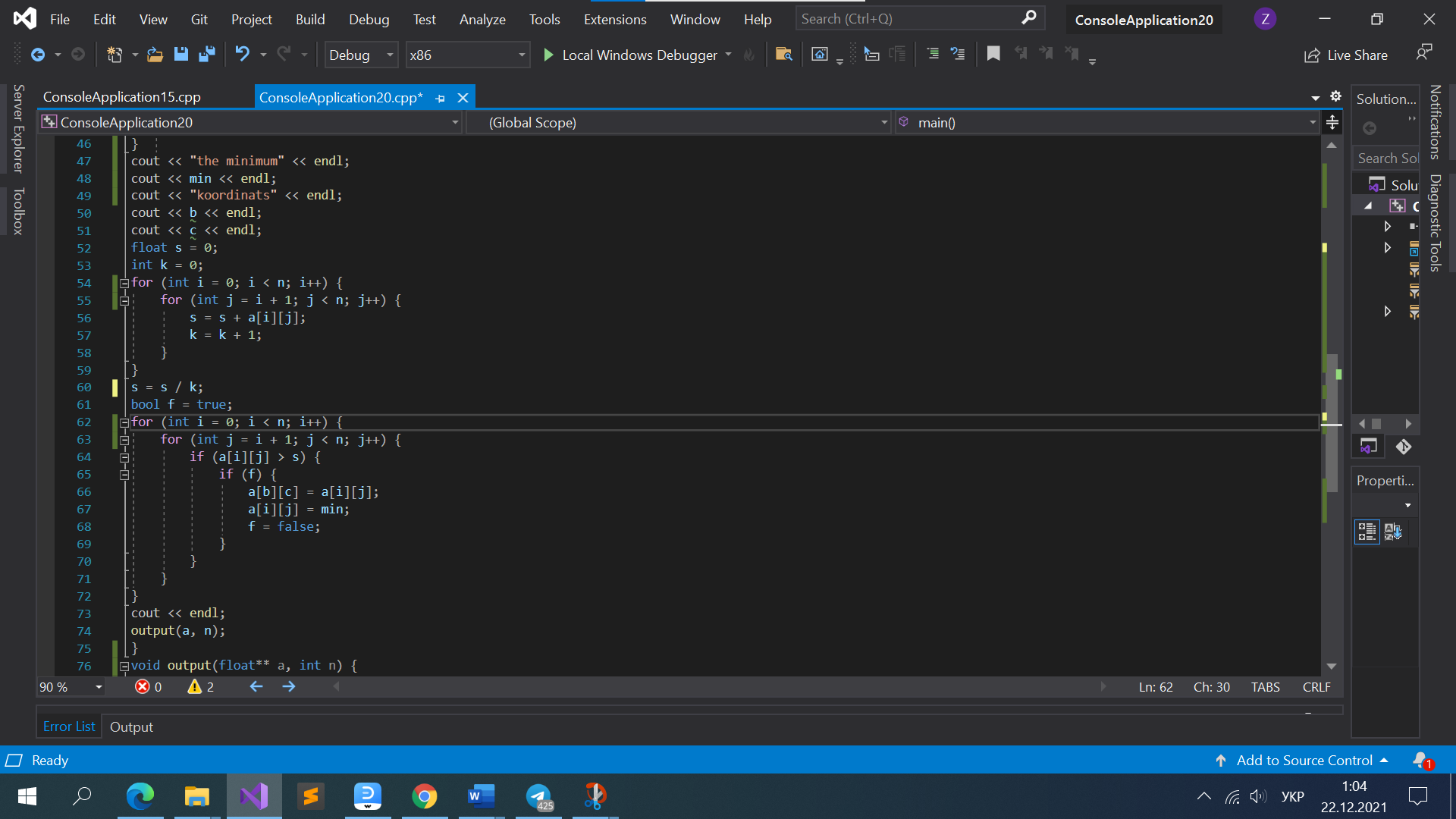
**Крок 5**

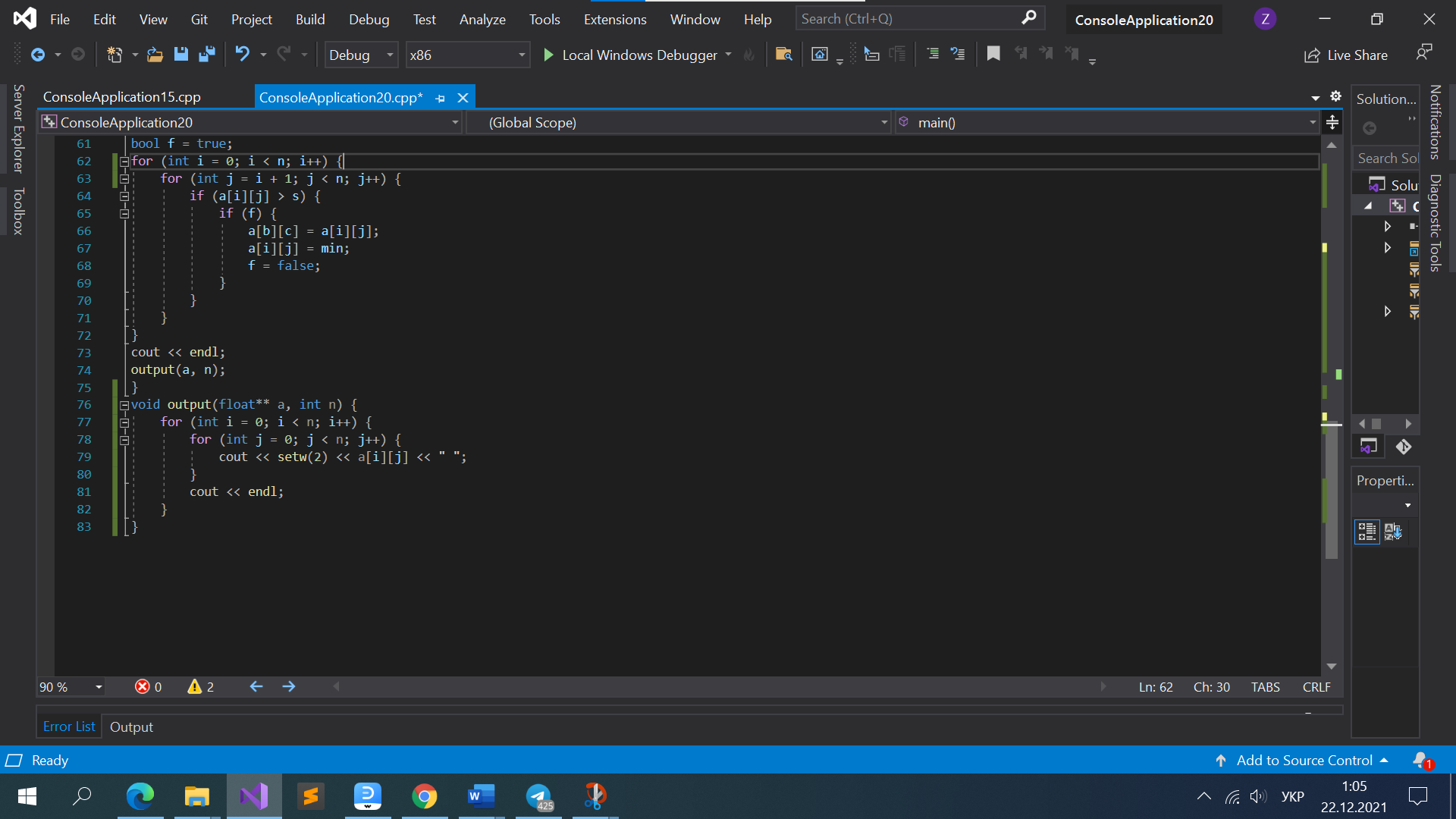


1. **Код програми**

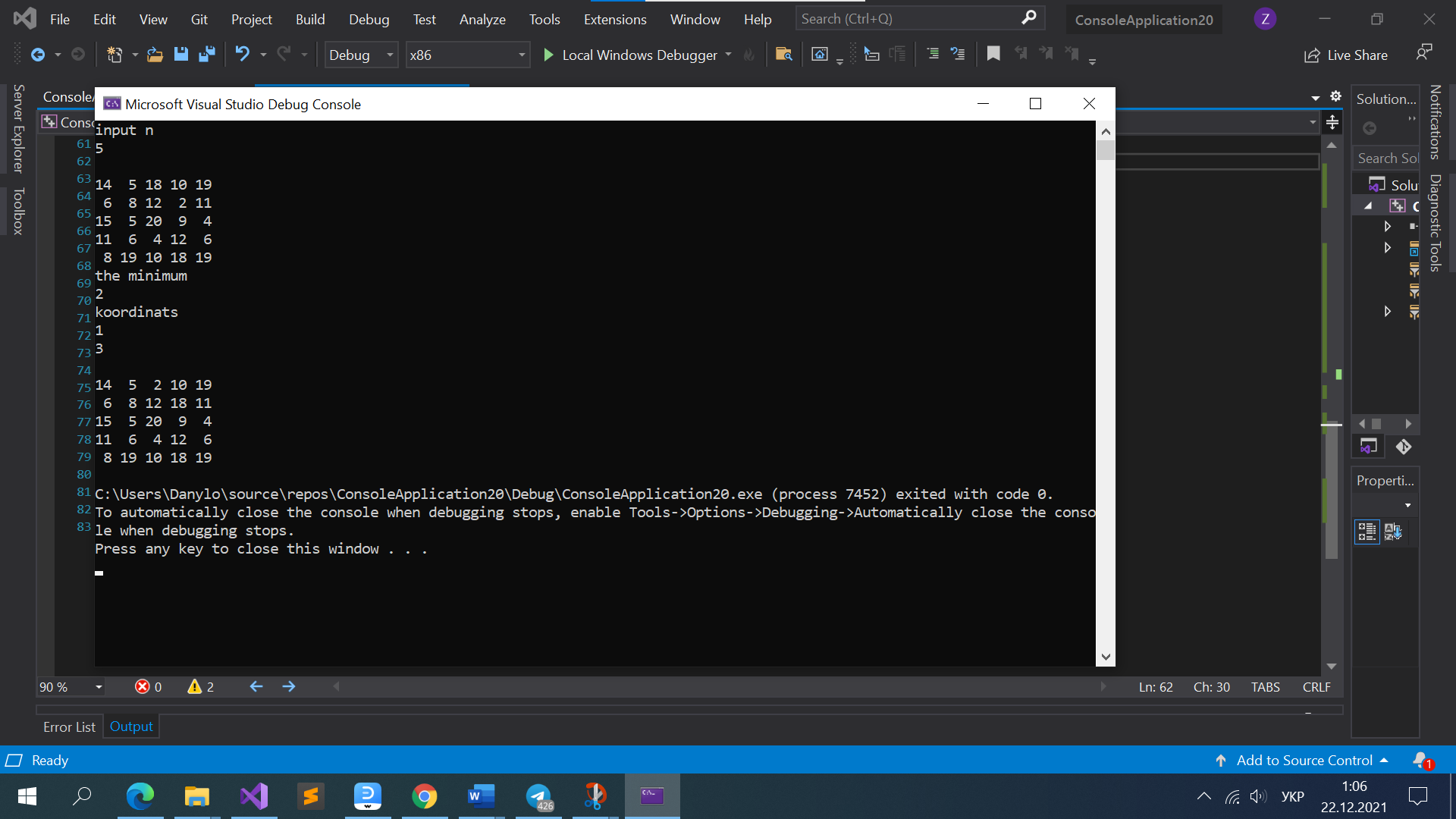








**7. Тестування програми**



1. **Висновок**

Було досліджено алгоритми обходу масивів, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Особливістю моєї роботи було знаходження середнього арифметичного чисел які знаходяться над головною діагоналлю матриці. Ще однією особливістю було знаходження першого мінімального елементу. Також декомпозував задачу на 6 кроків: визначив основні дії, ініціалізував двовимірний масив за допомогою функції rand(), організував обхід матриці по стовпчиках, знайшов перший мінімальний елемент та його місцезнаходження, потім знайшов середнє арифметичне чисел, які знаходяться над головною діагоналлю і обміняв елемент над головною діагоналлю, який більше за середнє арифметичне з мінімальним та вивів двовимірний масив