МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Національний технічний університет України

"Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Програмування. Частина 1. Основи програмування

Лабораторна робота №6

Тема: Реалізація алгоритмів з використанням одновимірних

масивів. Обробка одновимірних масивів

Варіант №11

Виконав: Коваль Б.А.

студент групи ІС-31

Перевірив:

Драган М.С.

Київ-2023

МЕТА :

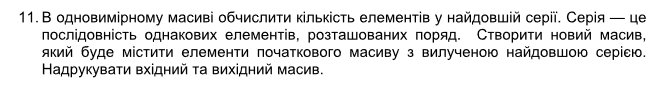
1. Ознайомитися з особливостями типу масиву;

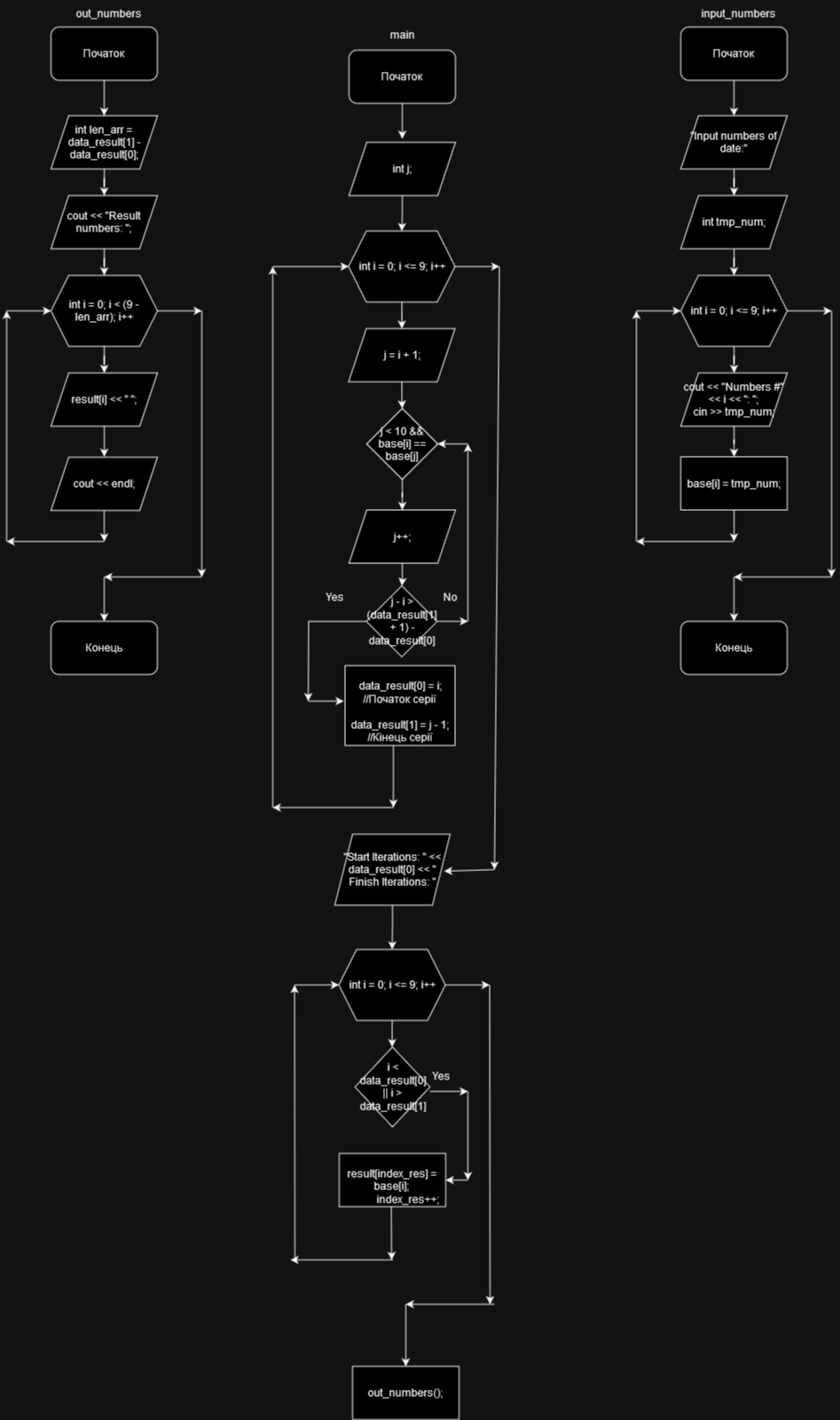
2. Опанувати технологію застосування масивів даних;

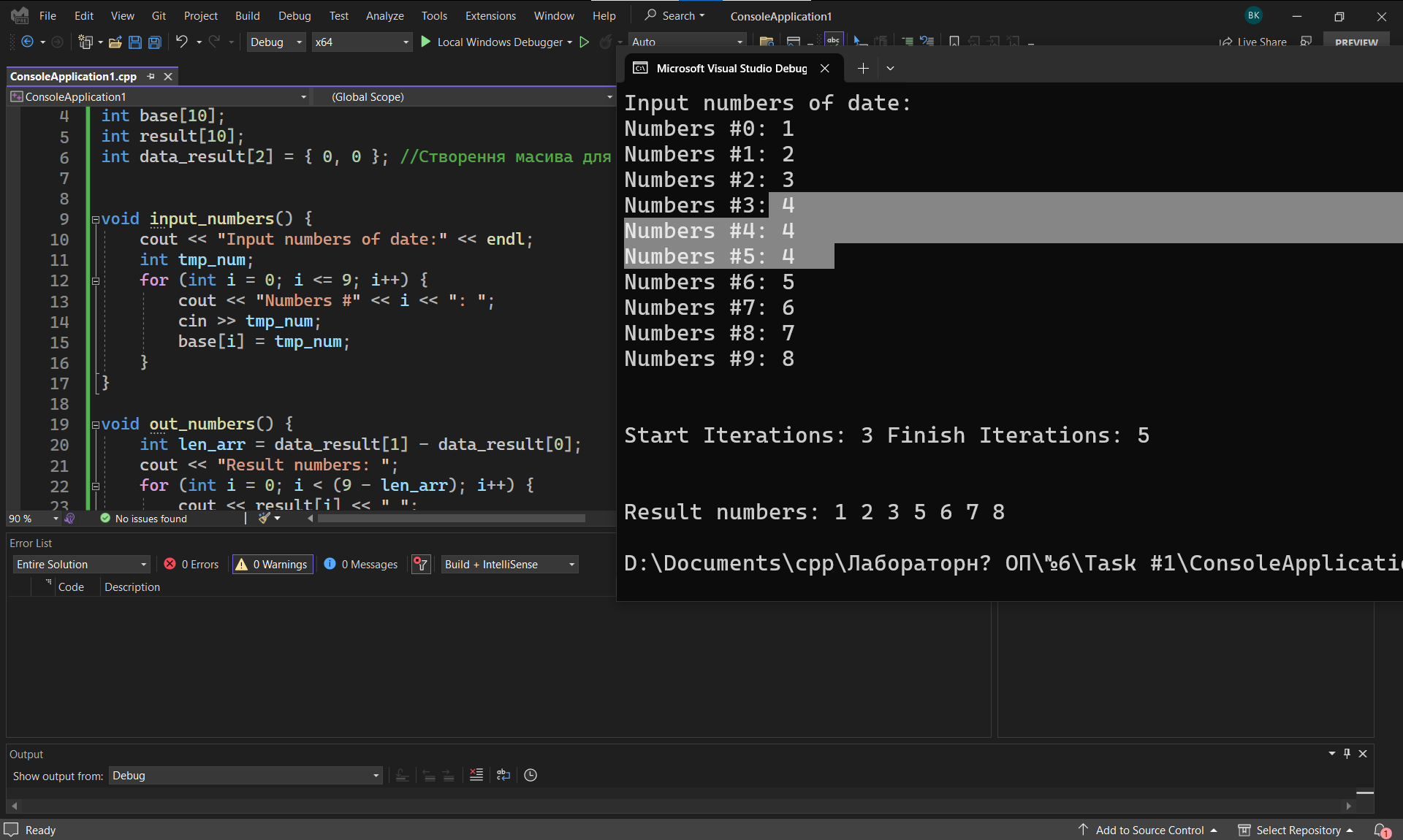
3. Навчитися розробляти алгоритми та програми із застосуванням одновимірних

масивів

ЗАВДАННЯ №1







#include <iostream>

using namespace std;

int base[10];

int result[10];

int data\_result[2] = { 0, 0 }; //Створення масива для зберігання поч. кінця серії

void input\_numbers() {

cout << "Input numbers of date:" << endl;

int tmp\_num;

for (int i = 0; i <= 9; i++) {

cout << "Numbers #" << i << ": ";

cin >> tmp\_num;

base[i] = tmp\_num;

}

}

void out\_numbers() {

int len\_arr = data\_result[1] - data\_result[0];

cout << "Result numbers: ";

for (int i = 0; i < (9 - len\_arr); i++) {

cout << result[i] << " ";

}

cout << endl;

}

int main()

{

input\_numbers();

int j;

for (int i = 0; i <= 9; i++) {

j = i + 1;

while (j < 10 && base[i] == base[j])

{

j++;

if (j - i > (data\_result[1] + 1) - data\_result[0]) {

data\_result[0] = i; //Початок серії

data\_result[1] = j - 1; //Кінець серії

}

}

}

cout << endl << endl << "Start Iterations: " << data\_result[0] << " Finish Iterations: " << data\_result[1] << endl << endl << endl;

int index\_res = 0;

for (int i = 0; i <= 9; i++) {

if (i < data\_result[0] || i > data\_result[1]) {

result[index\_res] = base[i];

index\_res++;

}

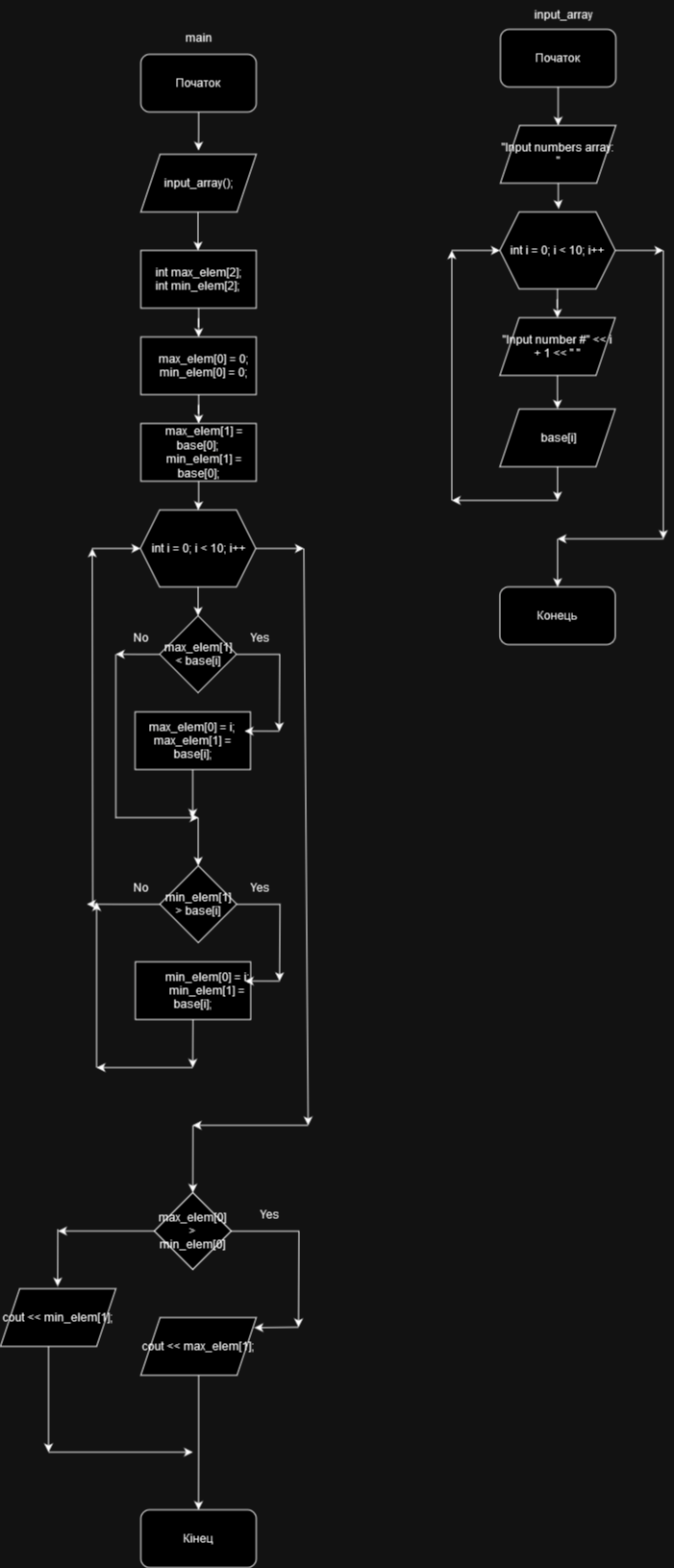
}

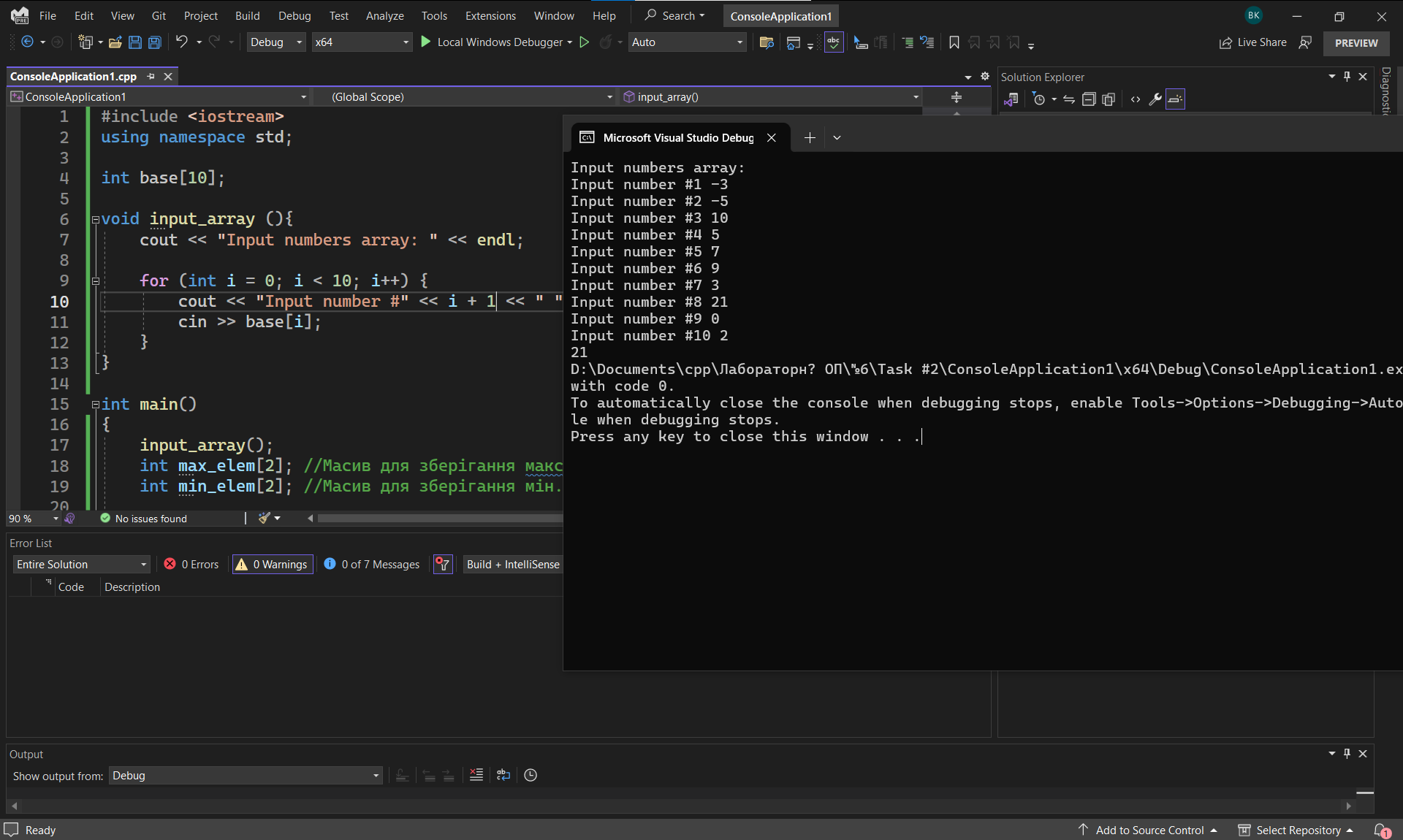
out\_numbers();

}

ЗАВДАННЯ №2







#include <iostream>

using namespace std;

int base[10];

void input\_array (){

cout << "Input numbers array: " << endl;

for (int i = 0; i < 10; i++) {

cout << "Input number #" << i + 1 << " ";

cin >> base[i];

}

}

int main()

{

input\_array();

int max\_elem[2]; //Масив для зберігання макс. елемн. (індек/елемент)

int min\_elem[2]; //Масив для зберігання мін. елемен. (індек/елемент)

max\_elem[0] = 0;

min\_elem[0] = 0;

max\_elem[1] = base[0];

min\_elem[1] = base[0];

for (int i = 0; i < 10; i++) {

if (max\_elem[1] < base[i]) {

max\_elem[0] = i;

max\_elem[1] = base[i];

}

if (min\_elem[1] > base[i]) {

min\_elem[0] = i;

min\_elem[1] = base[i];

}

}

if (max\_elem[0] > min\_elem[0])

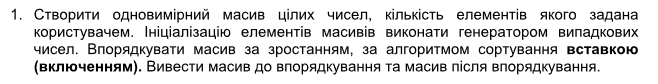
cout << max\_elem[1];

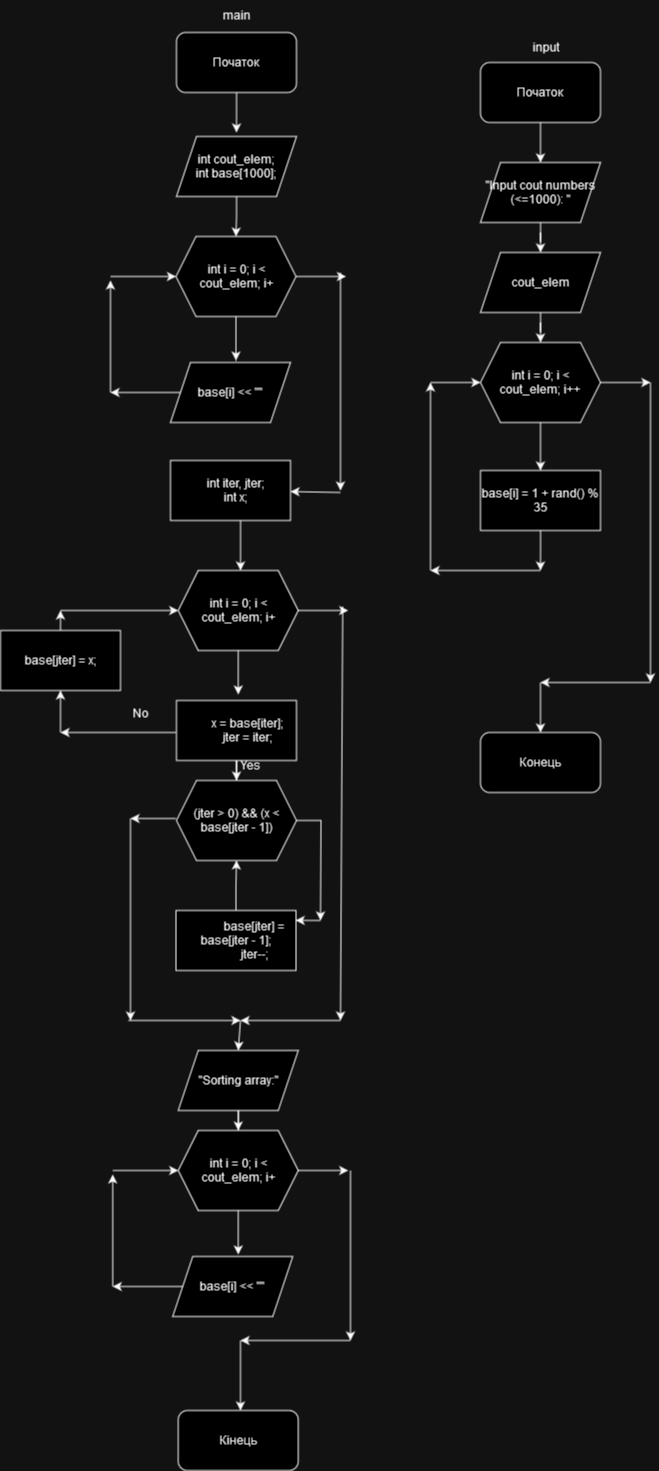
else

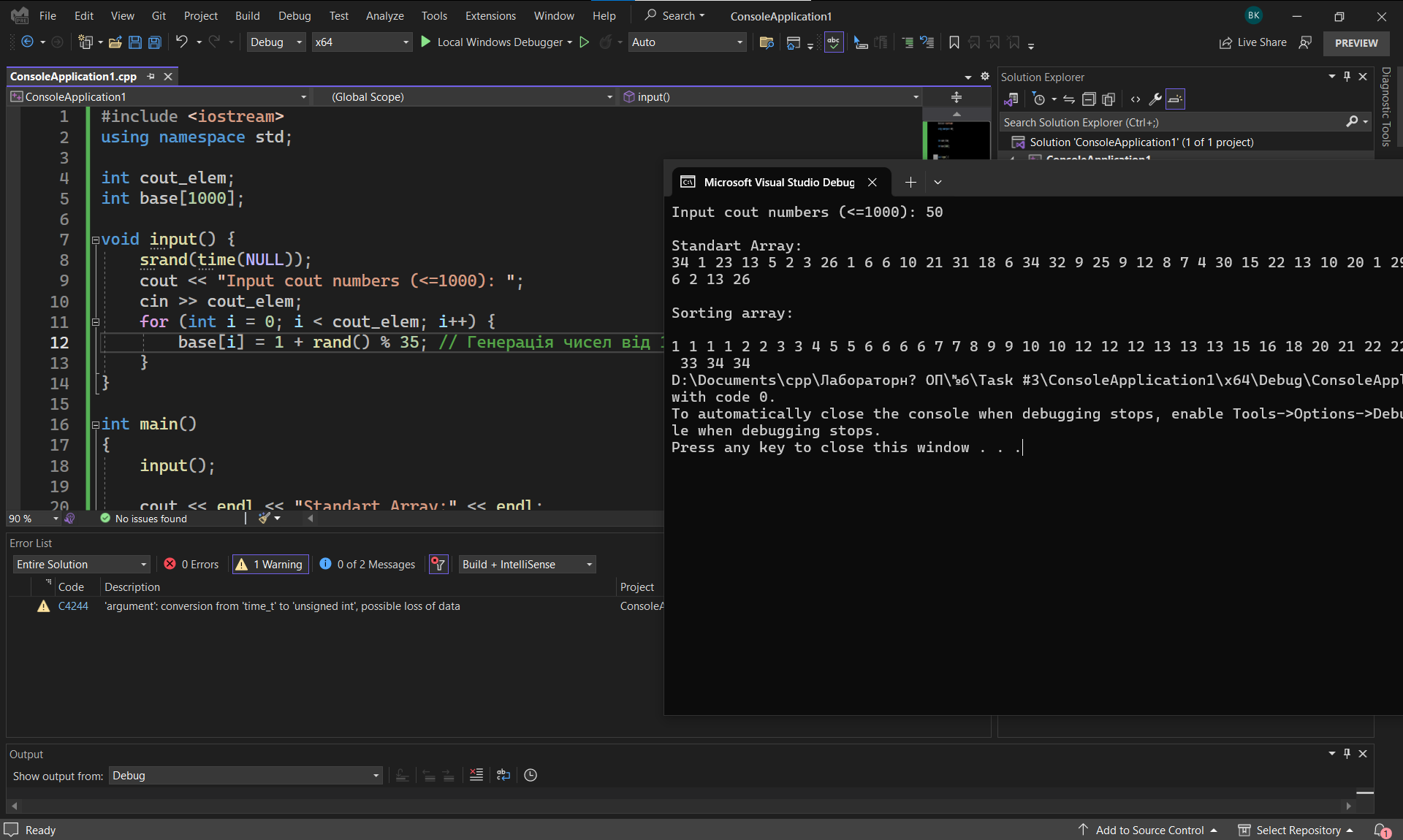
cout << min\_elem[1];

}

ЗАВДАННЯ №3







#include <iostream>

using namespace std;

int cout\_elem;

int base[25];

void input() {

srand(time(NULL));

cout << "Input cout numbers (<=25): ";

cin >> cout\_elem;

for (int i = 0; i < cout\_elem; i++) {

base[i] = 1 + rand() % 35; // Генерація чисел від 1 до 100

}

}

int main()

{

input();

cout << endl << "Standart Array:" << endl;

for (auto& elem : base) {

cout << elem << " ";

}

int iter, jter;

int x;

for (iter = 1; iter < cout\_elem; iter++)

{

x = base[iter];

jter = iter;

while ((jter > 0) && (x < base[jter - 1]))

{

base[jter] = base[jter - 1];

jter--;

}

base[jter] = x;

}

cout << endl << endl << "Sorting array:" << endl << endl;

for (auto& elem : base) {

cout << elem << " ";

}

}

КОНТРОЛЬНІ ЗАПИТАННЯ:

1. **Масив:** Масив - це структура даних, яка зберігає елементи одного типу у впорядкованому списку. Всі елементи масиву мають однаковий тип даних.

**Тип масиву:** Тип масиву визначається типом його елементів. Наприклад, масив цілих чисел (integers) має тип "цілі числа".

1. **Властивості масивів даних:**
   * Впорядкованість: Елементи масиву розташовані в послідовності.
   * Однотипність: Усі елементи мають однаковий тип даних.
   * Індексація: Доступ до елементів відбувається за допомогою індексів (номерів).
   * Фіксований розмір: Розмір масиву фіксований під час його створення.
2. **Принцип зображення масиву в оперативній пам'яті:**
   * Елементи масиву зберігаються в ряд у відповідності з їх індексами.
   * Залежно від мови програмування і типу даних, може використовуватися різне представлення (наприклад, послідовна або блокова зберігання).
3. **Доступ до елементів одновимірного масиву:**
   * Елементи масиву нумеруються починаючи з 0.
   * Для отримання доступу до елементу використовується індексація: масив[індекс].
4. **Список ініціалізації масиву:**
   * Зазвичай, список ініціалізації повинен мати стільки значень, скільки вказано в оголошенні масиву.
   * Деякі мови програмування дозволяють менший список, в цьому випадку невказані значення можуть мати значення за замовчуванням.
5. **Базові операції обробки одновимірних масивів:**
   * Читання та запис елементів.
   * Пошук елементів.
   * Сортування елементів.
   * Обробка масиву за допомогою циклів (наприклад, for або while).
6. **Методи сортування масивів:**
   * **Сортування обміном (Bubble Sort):** порівнюються сусідні елементи і, якщо вони в неправильному порядку, вони міняються місцями.
   * **Швидке сортування (QuickSort):** використовує рекурсивний підхід та "розбиття" масиву на менші частини.
   * **Злиття (Merge Sort):** розбиває масив на дві половини, сортує їх і потім об'єднує.
7. **Найшвидший та найповільніший алгоритми сортування:**
   * **Найшвидший:** Залежно від умов та реалізації, швидке сортування (QuickSort) часто вважається швидшим.
   * **Найповільніший:** Сортування обміном (Bubble Sort) зазвичай є одним з найповільніших.
8. **Алгоритми пошуку в упорядкованому та неупорядкованому масивах:**
   * **Упорядкований масив:** Бінарний пошук може бути використаний, оскільки масив вже впорядкований.
   * **Неупорядкований масив:** Лінійний пошук, оскільки немає можливості скористатися властивістю впорядкованості для оптимізації.
9. **Повернення масиву з функції:**
   * Масив може бути повернений з функції, передаючи його як результат або через покажчик.
10. **Подвійний покажчик на одновимірний масив:**
    * Подвійний покажчик вказує на адресу покажчика, який вказує на початок одновимірного масиву.
    * Використовується для зручного пересування в масиві та доступу до його елементів.

ВИСНОВОК:

Під час виконання лабораторної роботи з теми "Реалізація алгоритмів з використанням одновимірних масивів. Обробка одновимірних масивів" я досягнув усіх поставлених мет та завдань.

Ознайомлення з особливостями типу масиву надало мені чітке уявлення про структуру та доступ до елементів. Це важливо для поглибленого розуміння програмування, оскільки масиви є ключовою складовою багатьох програм та алгоритмів. Успішно опрацював технологію застосування масивів даних, показавши, як ефективно використовувати масиви для зберігання та обробки великої кількості даних у реальних сценаріях. Особливо задоволений тим, що здобув навички розробки алгоритмів та програм із використанням одновимірних масивів.