

ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №7

за курсом “Програмування”

студентки групи ПА-23-1

Мазур Віолети

кафедра комп'ютерних технологій, ДНУ

2023/2024

1. Постановка задачі

Складіть програму, яка здійснює обробку вхідних даних за допомогою одновимірних масивів. Для вводу даних і виводу результатів програма використовує текстові файли за допомогою перенаправлення у командному рядку, ввід з клавіатури також можливий. Кількість елементів масивів задається на початку файлу (вважаємо, що користувач всі дані вводить вірно). Програма повинна містити не менше 3 функцій. Одну із підзадач вирішити за допомогою вказівників при обробці масиву. Глобальні змінні не використовуємо. Розмірність одновимірних масивів при демонстрації виконання завдання повинна бути не меншою 15.

Варіант 11. Задайте два масиви $A(n)$ і $B(m)$. Виконайте такі завдання:

- визначте масив $C(n+m)$ з елементів масивів A і B , спочатку парних у порядку спадання, а потім непарних у порядку зростання;
- знайдіть добуток елементів масиву A , розташованих між максимальними і мінімальними за модулем елементами;
- упорядкуйте елементи масивів за спаданням модулів елементів.

2. Опис розв'язку + блок-схема

Програма призначена для обробки одновимірних масивів і виконує наступні завдання:

- 1) зчитування двох масивів A та B з текстового файлу. Перший рядок файлу вказує на кількість елементів у масиві A , за яким слідують самі елементи. Аналогічно другий рядок для масиву B . Функція `readArray` використовується для зчитування елементів масиву з файлу;

2) об'єднання масивів A та B в один масив C, з подальшим сортуванням парних елементів у порядку спадання, а непарних - у порядку зростання. Функція `mergeAndSortArrays` реалізує цю логіку, використовуючи додаткові вектори для парних і непарних елементів та стандартні алгоритми сортування;

3) знаходження добутку елементів масиву A, розташованих між максимальним і мінімальним за модулем елементами. Функція `productBetweenExtremes` спочатку знаходить індекси екстремальних елементів, а потім обчислює добуток елементів між ними;

4) сортування елементів масивів A та B за спаданням модулів елементів. Функція `sortArrayByMagnitude` використовує алгоритм сортування з власним критерієм порівняння, який базується на абсолютних значеннях елементів.

Результати роботи програми записуються у вихідний файл `output.txt`, де представлені об'єднаний і відсортований масив C, добуток елементів масиву A та відсортовані масиви A та B за модулем. Також є можливість вводу даних в консолі і вивід результатів в консолі.

Програма забезпечує гнучкість у введенні даних, дозволяючи користувачеві вводити дані як з клавіатури, так і через перенаправлення вхідного файлу в командному рядку. Однак, для демонстрації завдання, розмірність масивів повинна бути не меншою за 15.

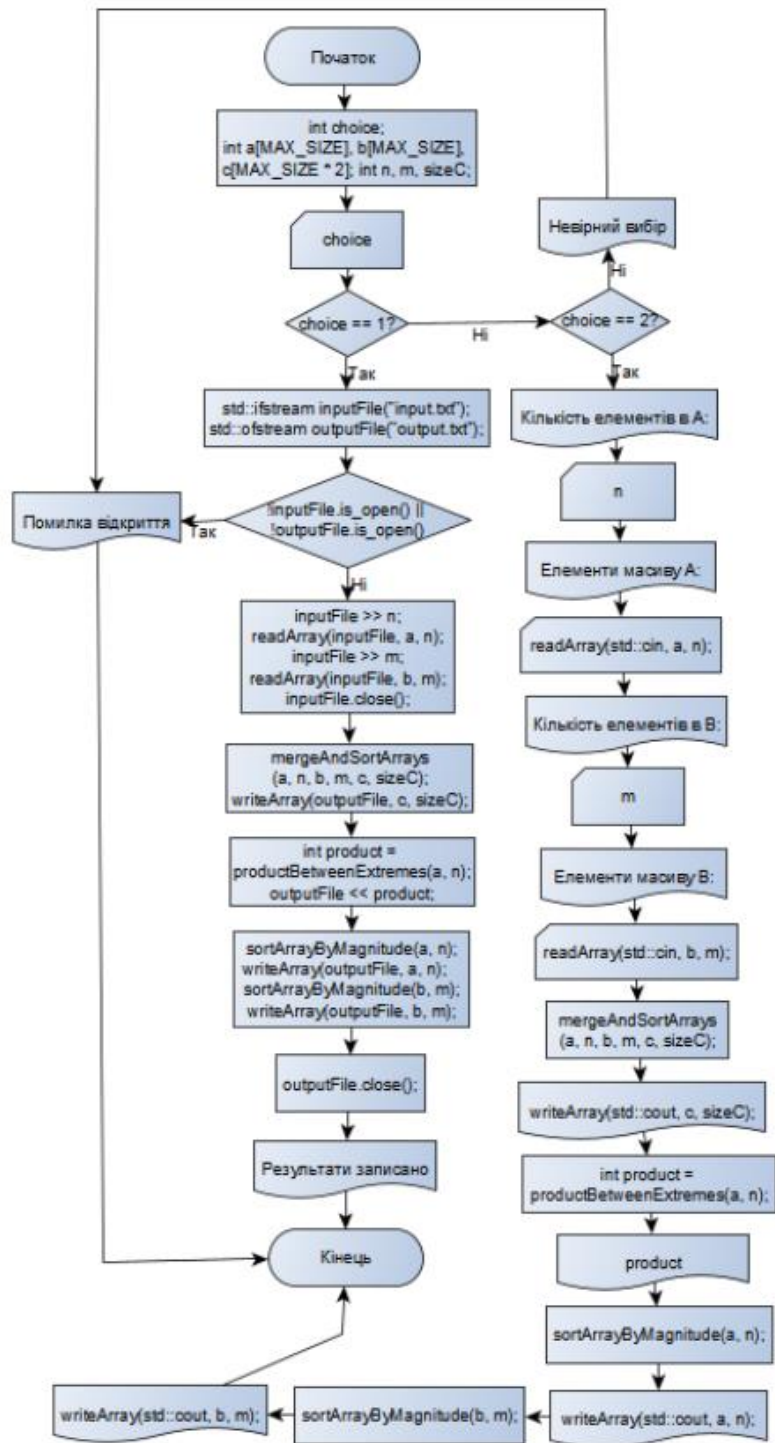


Рисунок 1 – Блок-схема роботи програми

3. Вихідний текст програми розв'язку задачі (основні фрагменти з коментарями)

```
#include <iostream>
```

```
#include <fstream>
```

```
#include <algorithm>
```

```
#include <cmath>
```

```
#include <locale>
```

```
const int MAX_SIZE = 30; // Максимальний розмір масиву
```

```
// Функція для читання масиву з потоку вводу
```

```
void readArray(std::istream& input, int* array, int size) {
```

```
    for (int i = 0; i < size; ++i) {
```

```
        input >> *(array + i);
```

```
    }
```

```
}
```

```
// Функція для запису масиву в потік виводу
```

```
void writeArray(std::ostream& output, const int array[], int size) {
```

```
    for (int i = 0; i < size; ++i) {
```

```
        output << array[i] << " ";
```

```
    }
```

```
    output << std::endl;
```

```
}
```

```
// Функція для об'єднання та сортування двох масивів
```

```
void mergeAndSortArrays(const int a[], int sizeA, const int b[], int sizeB, int  
merged[], int& sizeMerged) {
```

```
    int even[MAX_SIZE], odd[MAX_SIZE];
```

```
    int sizeEven = 0, sizeOdd = 0;
```

// Розділення елементів на парні та непарні

```
for (int i = 0; i < sizeA; ++i) {  
    if (a[i] % 2 == 0) {  
        even[sizeEven++] = a[i];  
    }  
    else {  
        odd[sizeOdd++] = a[i];  
    }  
}
```

```
for (int i = 0; i < sizeB; ++i) {  
    if (b[i] % 2 == 0) {  
        even[sizeEven++] = b[i];  
    }  
    else {  
        odd[sizeOdd++] = b[i];  
    }  
}
```

// Сортювання парних та непарних елементів

```
std::sort(even, even + sizeEven, std::greater<int>());  
std::sort(odd, odd + sizeOdd);
```

// Об'єднання відсортованих масивів

```
sizeMerged = 0;  
for (int i = 0; i < sizeEven; ++i) {  
    merged[sizeMerged++] = even[i];  
}
```

```

    }
    for (int i = 0; i < sizeOdd; ++i) {
        merged[sizeMerged++] = odd[i];
    }
}

```

// Функція для знаходження добутку елементів між максимальним та мінімальним за модулем

```

int productBetweenExtremes(const int array[], int size) {
    int minIndex = 0, maxIndex = 0;

    for (int i = 1; i < size; ++i) {
        if (std::abs(array[i]) < std::abs(array[minIndex])) {
            minIndex = i;
        }
        if (std::abs(array[i]) > std::abs(array[maxIndex])) {
            maxIndex = i;
        }
    }

    if (minIndex > maxIndex) {
        std::swap(minIndex, maxIndex);
    }

    int product = 1;
    for (int i = minIndex + 1; i < maxIndex; ++i) {
        product *= array[i];
    }
}

```

```

    return product;
}

// Функція для сортування масиву за спаданням модулів елементів
void sortArrayByMagnitude(int array[], int size) {
    std::sort(array, array + size, [](int a, int b) {
        return std::abs(a) > std::abs(b);
    });
}

int main() {
    setlocale(LC_CTYPE, "ukr");

    int choice;

    std::cout << "Виберіть спосіб вводу даних:\n";
    std::cout << "1. Читати з файлу\n";
    std::cout << "2. Вводити з клавіатури\n";
    std::cin >> choice;

    int a[MAX_SIZE], b[MAX_SIZE], c[MAX_SIZE * 2];
    int n, m, sizeC;

    if (choice == 1) {
        std::ifstream inputFile("C:/Users/Slim/Desktop/input.txt");
        std::ofstream outputFile(" C:/Users/Slim/Desktop/output.txt ");
    }

```

```

    if (!inputFile.is_open() || !outputFile.is_open()) {
        std::cerr << "Помилка відкриття файлу вводу або виводу." <<
std::endl;
        return 1;
    }
    inputFile >> n;
    readArray(inputFile, a, n);
    inputFile >> m;
    readArray(inputFile, b, m);
    inputFile.close();

    mergeAndSortArrays(a, n, b, m, c, sizeC);
    writeArray(outputFile, c, sizeC);

    int product = productBetweenExtremes(a, n);
    outputFile << "Добуток між екстремальними елементами в A: " <<
product << std::endl;

    sortArrayByMagnitude(a, n);
    writeArray(outputFile, a, n);

    sortArrayByMagnitude(b, m);
    writeArray(outputFile, b, m);

    outputFile.close();
    std::cout << "Результати записано у файл output.txt" << std::endl;
}
else if (choice == 2) {

```



```

std::cout << "Введіть кількість елементів у масиві А: ";
std::cin >> n;
std::cout << "Введіть елементи масиву А: ";
readArray(std::cin, a, n);
std::cout << "Введіть кількість елементів у масиві В: ";
std::cin >> m;
std::cout << "Введіть елементи масиву В: ";
readArray(std::cin, b, m);

mergeAndSortArrays(a, n, b, m, c, sizeC);
std::cout << "Об'єднаний та відсортований масив С: ";
writeArray(std::cout, c, sizeC);

int product = productBetweenExtremes(a, n);
std::cout << "Добуток між екстремальними елементами в А: " <<
product << std::endl;

sortArrayByMagnitude(a, n);
std::cout << "Масив А, відсортований за модулем: ";
writeArray(std::cout, a, n);

sortArrayByMagnitude(b, m);
std::cout << "Масив В, відсортований за модулем: ";
writeArray(std::cout, b, m);
}
else {
std::cerr << "Невірний вибір. Завершення програми." << std::endl;
return 1;
}

```

```
}  
    return 0;  
}
```

4. Опис інтерфейсу програми (керівництво користувача)

1) підготовка вхідних даних:

- створіть текстовий файл (наприклад, input.txt) для введення даних;
- у першому рядку файлу вкажіть кількість елементів першого масиву A;
- у наступних рядках введіть елементи масиву A, розділяючи їх пробілами або переносами рядків;
- після завершення масиву A, у новому рядку вкажіть кількість елементів другого масиву B;
- у наступних рядках введіть елементи масиву B, аналогічно масиву A;

2) запуск програми:

- запустіть виконуваний файл програми у командному рядку або через інтегроване середовище розробки (IDE);
- якщо програма розроблена для роботи з файлами, переконайтеся, що вхідний файл (input.txt) знаходиться у відповідній директорії або вкажіть шлях до файлу при запуску програми;

3) перегляд результатів:

- після виконання програми перегляньте вихідний файл (наприклад, output.txt) для ознайомлення з результатами обробки даних;
- у файлі будуть представлені;
- об'єднаний і відсортований масив C;
- добуток елементів масиву A, розташованих між максимальним і мінімальним за модулем елементами;
- відсортовані масиви A та B за спаданням модулів елементів;

4) додаткові можливості:

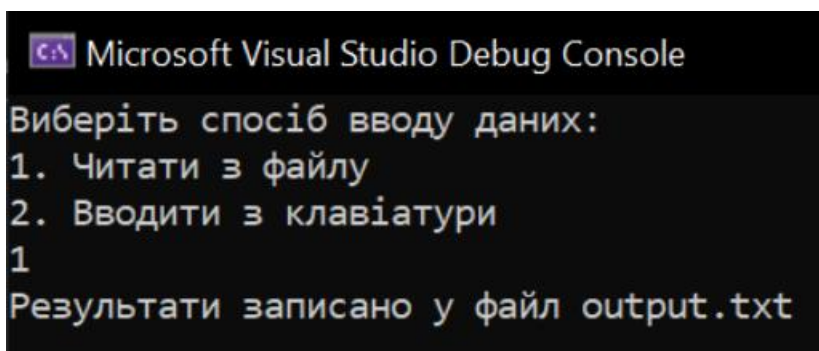
- програма також підтримує введення даних безпосередньо з клавіатури;

- у разі необхідності, можна модифікувати програму для зміни формату введення або виведення даних, а також для додавання додаткових функцій обробки масивів.

Під час використання програми варто переконатися, що ви дотримуетесь формату вхідних даних і правильно вказуєте шляхи до файлів при роботі з програмою.

5.Опис текстових прикладів

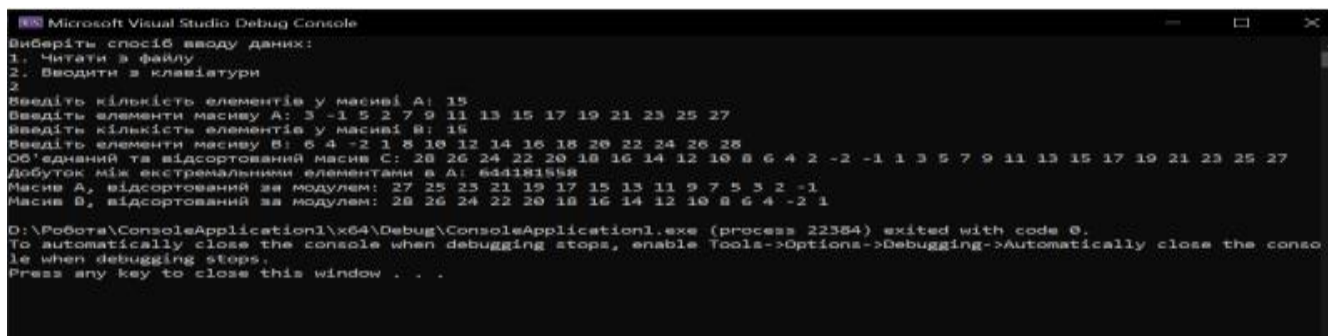
Ввід з файлу та вивід у файл:



```
Microsoft Visual Studio Debug Console
Виберіть спосіб вводу даних:
1. Читати з файлу
2. Вводити з клавіатури
1
Результати записано у файл output.txt
```

Ввід в консолі та вивід в консоль:

1) мінімальний елемент розташований лівіше максимального:



```
Microsoft Visual Studio Debug Console
Виберіть спосіб вводу даних:
1. Читати з файлу
2. Вводити з клавіатури
2
Введіть кількість елементів у масиві A: 15
Введіть елементи масиву A: 3 -1 5 2 7 9 11 13 15 17 19 21 23 25 27
Введіть кількість елементів у масиві B: 15
Введіть елементи масиву B: 6 4 -2 1 8 10 12 14 16 18 20 22 24 26 28
Об'єднаний та відсортований масив C: 28 26 24 22 20 18 16 14 12 10 8 6 4 2 -2 -1 1 3 5 7 9 11 13 15 17 19 21 23 25 27
Добуток між екстремальними елементами в A: 644181558
Масив A, відсортований за модулем: 27 25 23 21 19 17 15 13 11 9 7 5 3 2 -1
Масив B, відсортований за модулем: 28 26 24 22 20 18 16 14 12 10 8 6 4 -2 1

O:\Робота\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe (process 22384) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

2) мінімальний елемент розташований правіше максимального:

```
Microsoft Visual Studio Debug Console
Виберіть спосіб вводу даних:
1. Читати з файлу
2. Вводити з клавіатури
2
Введіть кількість елементів у масиві A: 15
Введіть елементи масиву A: 5 -1 3 2 8 10 12 14 16 18 20 22 24 26 28
Введіть кількість елементів у масиві B: 15
Введіть елементи масиву B: 1 4 -2 6 7 9 11 13 15 17 19 21 23 25 27
Об'єднаний та відсортований масив C: 28 26 24 22 20 18 16 14 12 10 8 6 4 2 -2 -1 1 3 5 7 9 11 13 15 17 19 21 23 25 27
Добуток між екстремальними елементами в A: -1557135360
Масив A, відсортований за модулем: 28 26 24 22 20 18 16 14 12 10 8 5 3 2 -1
Масив B, відсортований за модулем: 27 25 23 21 19 17 15 13 11 9 7 6 4 -2 1

D:\Робота\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe (process 24048) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

3) мінімальний і максимальний елементи збігаються:

```
Microsoft Visual Studio Debug Console
Виберіть спосіб вводу даних:
1. Читати з файлу
2. Вводити з клавіатури
2
Введіть кількість елементів у масиві A: 15
Введіть елементи масиву A: 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
Введіть кількість елементів у масиві B: 15
Введіть елементи масиву B: 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
Об'єднаний та відсортований масив C: 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
Добуток між екстремальними елементами в A: 1
Масив A, відсортований за модулем: 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
Масив B, відсортований за модулем: 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3

D:\Робота\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe (process 20508) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

4) мінімальний і максимальний елементи розташовані поряд:

```
Microsoft Visual Studio Debug Console
Виберіть спосіб вводу даних:
1. Читати з файлу
2. Вводити з клавіатури
2
Введіть кількість елементів у масиві A: 15
Введіть елементи масиву A: 3 -1 2 5 8 10 12 14 16 18 20 22 24 26 28
Введіть кількість елементів у масиві B: 15
Введіть елементи масиву B: 1 4 -2 6 7 9 11 13 15 17 19 21 23 25 27
Об'єднаний та відсортований масив C: 28 26 24 22 20 18 16 14 12 10 8 6 4 2 -2 -1 1 3 5 7 9 11 13 15 17 19 21 23 25 27
Добуток між екстремальними елементами в A: 1699741696
Масив A, відсортований за модулем: 28 26 24 22 20 18 16 14 12 10 8 5 3 2 -1
Масив B, відсортований за модулем: 27 25 23 21 19 17 15 13 11 9 7 6 4 -2 1

D:\Робота\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe (process 20552) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

5) мінімальний і максимальний елементи розташовані на початку і в кінці масиву:

```
Microsoft Visual Studio Debug Console
Виберіть спосіб вводу даних:
1. Читати з файлу
2. Вводити з клавіатури
2
Введіть кількість елементів у масиві A: 15
Введіть елементи масиву A: 3 -1 2 5 8 10 12 14 16 18 20 22 24 26 28
Введіть кількість елементів у масиві B: 15
Введіть елементи масиву B: 1 4 -2 6 7 9 11 13 15 17 19 21 23 25 27
Об'єднаний та відсортований масив C: 28 26 24 22 20 18 16 14 12 10 8 6 4 2 -2 -1 1 3 5 7 9 11 13 15 17 19 21 23 25 27
Добуток між екстремальними елементами в A: 1699741696
Масив A, відсортований за модулем: 28 26 24 22 20 18 16 14 12 10 8 5 3 2 -1
Масив B, відсортований за модулем: 27 25 23 21 19 17 15 13 11 9 7 6 4 -2 1

D:\Робота\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe (process 20552) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

6) у масиві знаходиться один від'ємний елемент:

```
Microsoft Visual Studio Debug Console
Виберіть спосіб вводу даних:
1. Читати з файлу
2. Вводити з клавіатури
2
Введіть кількість елементів у масиві A: 15
Введіть елементи масиву A: 3 2 -1 4 5 6 7 8 9 10 11 12 13 14 15
Введіть кількість елементів у масиві B: 0
Введіть елементи масиву B: Об'єднаний та відсортований масив C: 14 12 10 8 6 4 2 -1 3 5 7 9 11 13 15
Добуток між екстремальними елементами в A: 1644813312
Масив A, відсортований за модулем: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 -1
Масив B, відсортований за модулем:

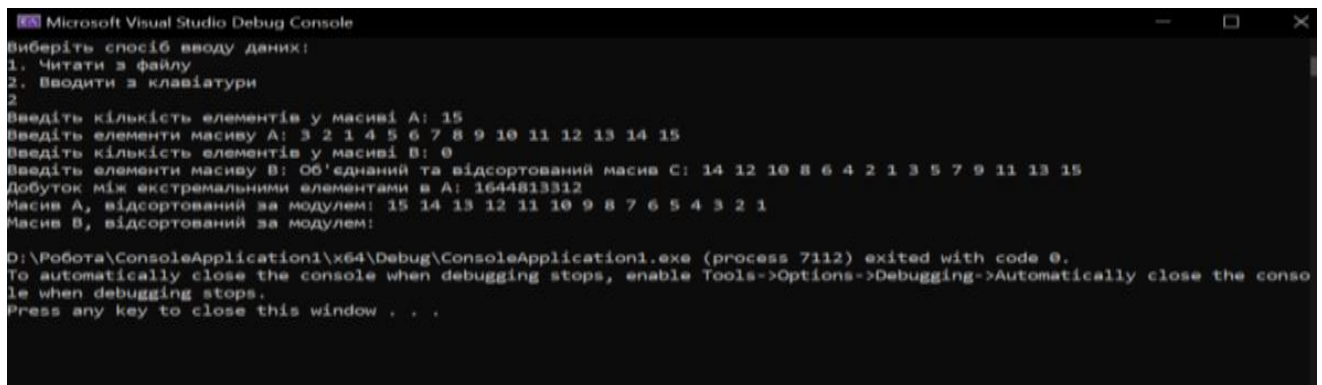
D:\Робота\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe (process 15760) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

7) у масиві знаходяться декілька від'ємних елементів:

```
Microsoft Visual Studio Debug Console
Виберіть спосіб вводу даних:
1. Читати з файлу
2. Вводити з клавіатури
2
Введіть кількість елементів у масиві A: 15
Введіть елементи масиву A: -3 2 -1 4 5 -6 7 8 -9 10 11 -12 13 14 15
Введіть кількість елементів у масиві B: 0
Введіть елементи масиву B: Об'єднаний та відсортований масив C: 14 10 8 4 2 -6 -12 -9 -3 -1 5 7 11 13 15
Добуток між екстремальними елементами в A: -1644813312
Масив A, відсортований за модулем: 15 14 13 -12 11 10 -9 8 7 -6 5 4 -3 2 -1
Масив B, відсортований за модулем:

D:\Робота\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe (process 11992) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

8) у масиві немає від'ємних елементів:



```
Microsoft Visual Studio Debug Console
Виберіть спосіб вводу даних:
1. Читати з файлу
2. Вводити з клавіатури
2
Введіть кількість елементів у масиві A: 15
Введіть елементи масиву A: 3 2 1 4 5 6 7 8 9 10 11 12 13 14 15
Введіть кількість елементів у масиві B: 0
Введіть елементи масиву B: Об'єднаний та відсортований масив C: 14 12 10 8 6 4 2 1 3 5 7 9 11 13 15
Добуток між екстремальними елементами в A: 1644813312
Масив A, відсортований за модулем: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
Масив B, відсортований за модулем:
D:\Робота\ConsoleApplication1\x64\Debug\ConsoleApplication1.exe (process 7112) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

6. Аналіз помилок (опис усунення зауважень)

Під час розробки програми виникло декілька помилок, які були усунені. Була помилка у логіці сортування, яка призводила до неправильного порядку елементів у вихідному масиві. Ми виправили умову порівняння у функції сортування для забезпечення правильного порядку елементів. Також спочатку були проблеми з читанням даних з файлу та записом результатів у файл. Ми перевірили правильність шляхів до файлів та виправили формат введення/виведення даних.

7. Висновки

У цій лабораторній роботі було поставлено завдання створити два масиви $A(n)$ і $B(m)$ і виконати над ними різні операції. По-перше, ми визначили масив $C(n+m)$, об'єднавши елементи з масивів A і B . Елементи в масиві C були розташовані на дві частини – спочатку парні елементи в порядку спадання, потім непарні елементи в порядку зростання. Далі ми обчислили добуток елементів масиву A , які були розташовані між максимальним і мінімальним елементами за абсолютною величиною. Це передбачало визначення максимального та мінімального елементів у масиві A , обчислення їхніх абсолютних значень, а потім знаходження добутку елементів, розташованих між цими двома крайніми значеннями. Відсортовано елементи масивів у порядку спадання на основі абсолютних значень елементів. Цей процес вимагав від нас обчислити абсолютні значення кожного елемента, відсортувати їх відповідно, а потім змінити порядок елементів у масивах A і B на основі цього критерію сортування. Загалом, поставлені завдання допомогли більш точно навчитись аналізувати масиви за допомогою різних математичних операцій і методів сортування, покращуючи розуміння маніпулювання масивами та алгоритмічного вирішення проблем.

