

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів
Кафедра систем управління літальних апаратів

Лабораторна робота № 8

з дисципліни «Алгоритмізація та програмування»
на тему "Реалізація алгоритмів сортування та робота з файлами на мові C ++"

XAI.301. 174. 319. 21 ЛР

Виконав студент гр. _____319_____

_____ Володимир Сисоєв
(підпис, дата) (П.І.Б.)

Перевірів

_____ к.т.н., доц. Олена ГАВРИЛЕНКО
(підпис, дата) (П.І.Б.)

2023

МЕТА РОБОТИ

Вивчити теоретичний матеріал по алгоритмам обробки масивів на мові C++, а також бібліотеки для роботи з файлами і реалізувати оголошення, введення з файлу, обробку і виведення в файл одновимірних і двовимірних масивів на мові C ++ в середовищі Visual Studio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. За допомогою текстового редактору створити текстовий файл «array_in_n.txt» з елементами вихідного масиву (n - номер варіанта). У програмі на C++ перетворити масив відповідно до свого варіанту завдання, ім'я файлу і необхідні змінні ввести з консолі. Вивести результати у файл «array_out_n.txt».

Array 75. Дан масив розміру N. Переставити в зворотному порядку елементи масиву, розташовані між його мінімальним і максимальним елементами, включаючи мінімальний і максимальний елементи.

Завдання 2. За допомогою текстового редактору створити текстовий файл «matr_in_n.txt» з елементами вихідного двовимірного масиву (n - номер варіанта). У програмі обробити матрицю відповідно до свого варіанту завдання, ім'я файлу і необхідні змінні ввести з консолі. Дописати результати в той же файл.

Matrix 43. Дана матриця розміру $M \times N$. Знайти кількість її стовпців, елементи яких впорядковані за спаданням.

Завдання 3. Вивчити метод сортування відповідно до свого варіанту проаналізувати його складність і продемонструвати на прикладі з 7-ми Елементів. Реалізувати у вигляді окремої функції алгоритм сортування елементів масиву. Також окремими функціями реалізувати зчитування масиву з текстового файлу і виведення відсортованого масиву в консоль.

24	Бульбашкова	Зменшення	Цілий
----	-------------	-----------	-------

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Array 75

Вхідні дані (ім'я, опис, тип, обмеження):

- 1) Кількість елементів масиву N
- 2) Створення текстового файлу

Вихідні дані (ім'я, опис, тип):

- 1) Переставлені елементи масиву в зворотному порядку
- 2) Запис результату у файл

Алгоритм вирішення показано на рис. 1

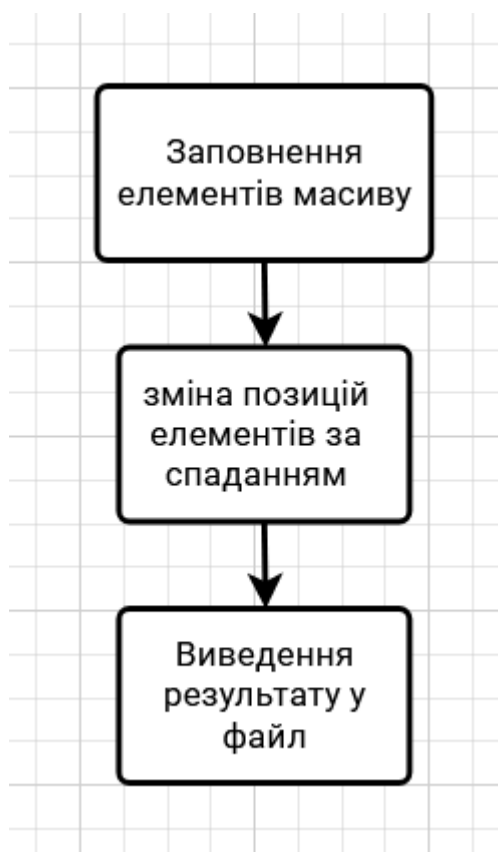


Рисунок 1 – Array 75

Лістинг коду вирішення задачі Array 75 наведено в дод. А (стор. 7-8).

Екран роботи програми показаний на рис. Б.1.

Завдання 2.

Вирішення задачі Matrix 43

Вхідні дані (ім'я, опис, тип, обмеження):

1) кількість рядків row, int

1) кількість стовпців col, int

2) Створення текстового файлу

Вихідні дані (ім'я, опис, тип):

1) кількість стовпців впорядкованих за спаданням count

2) Запис результату у файл

Алгоритм вирішення показано на рис. 2

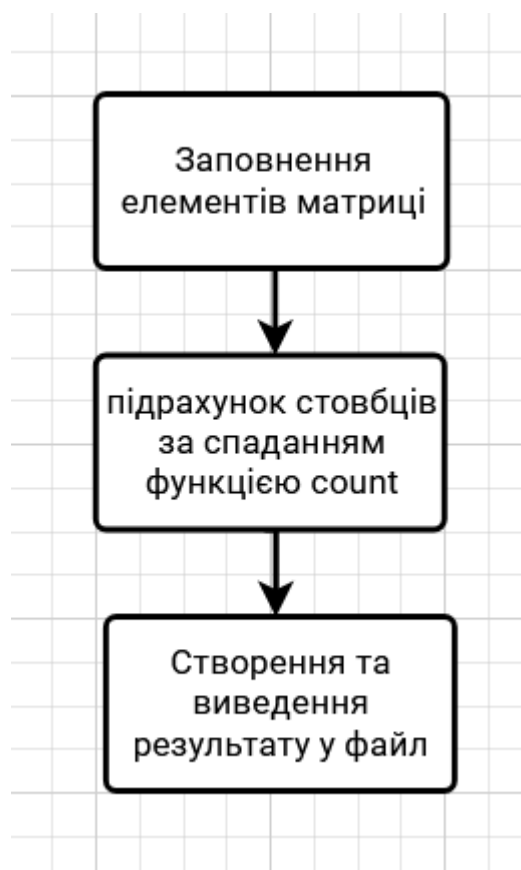


Рисунок 2 – Matrix 43

Лістинг коду вирішення задачі Matrix 43 наведено в дод. А (стор. 7-8).

Екран роботи програми показаний на рис. Б.2

Завдання 3.

Вирішення задачі Sort 24.

Вхідні дані (ім'я, опис, тип, обмеження):

1) numbers – масив цілих чисел який потрібно впорядкувати , int ,

Вихідні дані (ім'я, опис, тип):

1) numbers – відсортований масив за спаданням , int

Алгоритм вирішення показано на рис. 3



Рисунок 3 – Sort 24

Лістинг коду вирішення задачі Sort 24 наведено в дод. А (стор. 7-8).

Екран роботи програми показаний на рис. Б.3.

ВИСНОВКИ

Було вивчено теоретичний матеріал по алгоритмам обробки масивів на мові C++, а також бібліотеки для роботи з файлами і реалізувати оголошення, введення з файлу, обробку і виведення в файл одновимірних і двовимірних масивів на мові C ++ в середовищі Visual Studio.

ДОДАТОК А

Лістинг коду програми

```

void Array75() {
    int N;
    cout << "enter size of array: ";
    cin >> N;
    vector<int> array(N);
    cout << "enter elements of array:\n";
    for (int i = 0; i < N; ++i) {
        cin >> array[i];
    }
    // find min and max element of array
    auto min_element_it = min_element(array.begin(), array.end());
    auto max_element_it = max_element(array.begin(), array.end());
    int min_index = distance(array.begin(), min_element_it);
    int max_index = distance(array.begin(), max_element_it);
    // identify the range of elements to be rearranged
    int start_index = min(min_index, max_index);
    int end_index = max(min_index, max_index);
    // rearrange elements in reverse order
    reverse(array.begin() + start_index, array.begin() + end_index + 1);
    // output results in file
    ofstream outfile("array_out_75.txt");
    if (outfile.is_open()) {
        for (int i = 0; i < N; ++i) {
            outfile << array[i] << " ";
        }
        outfile.close();
        cout << "results saved in file: array_out_75.txt\n";
    }
    else {
        cerr << "error\n";
    }
}

void Matrix43() {
    int M, N;
    std::cout << "Enter row (M): ";
    std::cin >> M;
    std::cout << "Enter col (N): ";
    std::cin >> N;

    std::vector<std::vector<int>> matrix(M, std::vector<int>(N));
    std::cout << "Enter element of matrix:\n";
    for (int i = 0; i < M; ++i) {
        for (int j = 0; j < N; ++j) {
            std::cin >> matrix[i][j];
        }
    }

    // find columns by decreasing
    int count = 0;
    for (int j = 0; j < N; ++j) {
        bool sorted = true;
        for (int i = 1; i < M; ++i) {
            if (matrix[i][j] > matrix[i - 1][j]) {
                sorted = false;
                break;
            }
        }
    }
}

```

```

        if (sorted) {
            ++count;
        }
    }
    // output results in file
    std::string filename;
    std::cout << "enter name of file in which the result will be saved: " <<
endl;
    std::cin >> filename;

    std::ofstream outputFile(filename);
    if (outputFile.is_open()) {
        outputFile << "columns sorted by decreasing: " << count << std::endl;
        outputFile.close();
        std::cout << "Results saved in file: " << filename << std::endl;
    }
    else {
        std::cerr << "error!" << std::endl;
    }
}

void swap(int& a, int& b) {
    int temp = a;
    a = b;
    b = temp;
}

// Bubble Sorting in Decreasing Order
void bubbleSort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] < arr[j + 1]) {
                swap(arr[j], arr[j + 1]);
            }
        }
    }
}

void Sort24() {
    const int size = 7;
    int numbers[size];

    // array filling
    cout << "Enter " << size << " whole number:\n";
    for (int i = 0; i < size; i++) {
        cin >> numbers[i];
    }

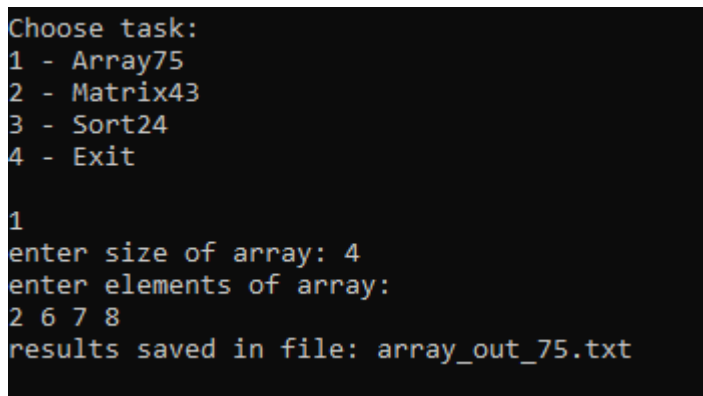
    // sort method
    bubbleSort(numbers, size);

    // output sorted number
    cout << "Sorted numbers in decreasing order:\n";
    for (int i = 0; i < size; i++) {
        cout << numbers[i] << " ";
    }
    cout << endl;
}

```

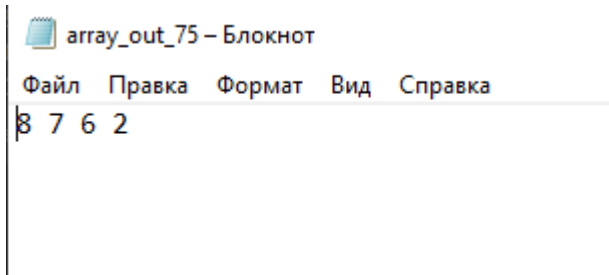

ДОДАТОК Б

Скрін-шоти вікна виконання програми



```
Choose task:
1 - Array75
2 - Matrix43
3 - Sort24
4 - Exit

1
enter size of array: 4
enter elements of array:
2 6 7 8
results saved in file: array_out_75.txt
```



array_out_75 – Блокнот

Файл Правка Формат Вид Справка

8 7 6 2

Рисунок Б.1 – Екран виконання програми для вирішення завдання
Array 75

```
Choose task:
1 - Array75
2 - Matrix43
3 - Sort24
4 - Exit

2
Enter row (M): 2
Enter col (N): 2
Enter element of matrix:
4 8
1 9
enter name of file in which the result will be saved:
matr_out_43.txt
Results saved in file: matr_out_43.txt
```

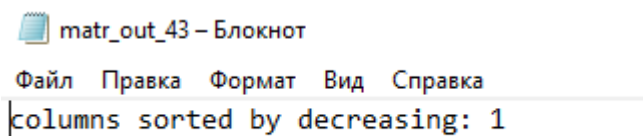


Рисунок Б.2 – Экран виконання програми для вирішення завдання
Matrix 43

```
Choose task:
1 - Array75
2 - Matrix43
3 - Sort24
4 - Exit

3
Enter 7 whole number:
1 3 2 4 7 5 6
Sorted numbers in decreasing order:
7 6 5 4 3 2 1
```

Рисунок Б.3 – Екран виконання програми для вирішення завдання
Sort 24