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 С ++»

ХАІ.301.175.318.13 ЛР

Виконав студент гр. № 319а

Михайло Тютюнник

(підпис, дата) (П.І.Б.)

Перевірив к.т.н., доцент

( вчена ступінь, вчене звання)

Олена ГАВРИЛЕНКО (підпис,дата) (П.І.Б.)

2025

Мета роботи

Вивчити теоретичний матеріал по алгоритмам обробки масивів на мові С++, 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: Реверсування масиву дійсних чисел, зчитаного з файлу array\_in\_28.txt, та запис результату у файл array\_out\_28.txt.

Завдання 2: Обчислення сум елементів кожного рядка матриці з файлу matr\_in\_28.txt та допис результатів у цей же файл.

Завдання 3: Сортування символьного масиву з файлу array\_in\_char\_28.txt методом вибору за спаданням та запис результату у файл array\_out\_char\_28.txt.

Виконання роботи

Завдання 1: Реверсування масиву

Вхідні дані:  
Дано текстовий файл array\_in\_28.txt, який містить:

* У першому рядку — ціле число n (розмір масиву).
* У другому рядку — n дійсних чисел, розділених пробілами.

Вихідні дані:  
Очікується створити текстовий файл array\_out\_28.txt, який містить:

* У першому рядку — розмір масиву n.
* У другому рядку — ті самі елементи, але у зворотному порядку.

Алгоритм:

1. Відкрити вхідний файл для читання.
2. Зчитати перший рядок — розмір масиву n.
3. Зчитати другий рядок — елементи масиву.
4. Створити вихідний файл для запису.
5. Записати у перший рядок вихідного файлу розмір масиву n.
6. Записати у другий рядок елементи масиву у зворотному порядку.
7. Закрити обидва файли.
8. Вивести повідомлення про успішне завершення.

Лістинг коду до завдання 1 наведено в дод. А (стор. 4).

Діаграма активності програми наведена в дод. Б(стор. 13, рис.1)

Завдання 2: Суми рядків матриці

Вхідні дані:  
Дано текстовий файл matr\_in\_28.txt, який містить:

* У першому рядку — два цілих числа m (кількість рядків) та n (кількість стовпців).
* У наступних m рядках — по n дійсних чисел, розділених пробілами (елементи матриці).

Вихідні дані:  
Очікується дописати у кінець того ж файлу matr\_in\_28.txt:

* Підпис "Row sums:".
* Для кожного рядка матриці — його номер та суму елементів у форматі "Row X: Y.Y".

Алгоритм:

1. Відкрити вхідний файл для читання.
2. Зчитати перший рядок — розміри матриці m та n.
3. Зчитати наступні m рядків — елементи матриці.
4. Відкрити файл у режимі дописування.
5. Дописати у файл заголовок "Row sums:".
6. Для кожного рядка матриці:  
   a. Обчислити суму елементів.  
   b. Дописати результат у файл.
7. Закрити файл.
8. Вивести повідомлення про успішне завершення.

Лістинг коду до завдання 2 наведено в дод. А (стор. 4).

Завдання 3: Сортування символів

Вхідні дані:  
Дано текстовий файл array\_in\_char\_28.txt, який містить:

* У першому рядку — ціле число n (розмір масиву).
* У другому рядку — n символів, розділених пробілами.

Вихідні дані:  
Очікується створити текстовий файл array\_out\_char\_28.txt, який містить:

* У першому рядку — розмір масиву n.
* У другому рядку — символи, відсортовані за спаданням.

Алгоритм:

1. Відкрити вхідний файл для читання.
2. Зчитати перший рядок — розмір масиву n.
3. Зчитати другий рядок — елементи масиву.
4. Відсортувати масив методом вибору за спаданням:  
   Для кожного елемента з індексом i від 0 до n-2:  
   Знайти індекс maxIdx найбільшого елемента у несортованій частині.  
   Поміняти місцями елементи з індексами i та maxIdx.
5. Відкрити вихідний файл для запису.
6. Записати у перший рядок розмір масиву n.
7. Записати у другий рядок відсортовані елементи.
8. Закрити файли.
9. Вивести повідомлення про успішне завершення.

Лістинг коду до завдання 3 наведено в дод. А (стор. 4).

Діаграма активності програми наведена в дод. Б(стор. 13, рис.2)

Висновки

У ході виконання лабораторної роботи було успішно реалізовано три ключові алгоритми обробки даних, що демонструють практичне застосування роботи з файлами та масивами у мові С++. Виконана робота підтвердила теоретичні знання з алгоритмізації та надала цінний практичний досвід роботи з файлами та масивами у мові програмування С++. Отримані результати можуть слугувати основою для розробки більш складних програм з обробки даних.

Додаток А

Лістинг коду програми

array\_in\_28.txt

7

7.7 6.6 5.5 4.4 3.3 2.2 1.1

array\_in\_char\_28.txt

7

g e b k a z d

array\_utils.cpp

#include "matrix\_utils.h"

#include <fstream> // Для роботи з файлами

#include <vector> // Для використання динамічних масивів

#include <iostream>

// Функція для зчитування матриці з файлу і обчислення сум рядків

void processMatrixFromFile(const string& filename) {

ifstream fin(filename); // Відкриваємо файл для читання

ofstream fout(filename, ios::app); // Відкриваємо файл для дописування

if (!fin.is\_open() || !fout.is\_open()) {

cout << "Error opening file." << endl;

return;

}

int m, n;

fin >> m >> n; // Зчитуємо кількість рядків і стовпців

vector<vector<double>> matrix(m, vector<double>(n)); // Створюємо матрицю

// Зчитуємо елементи матриці

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

for (int j = 0; j < n; j++) {

fin >> matrix[i][j];

}

}

fout << "\nRow sums:" << endl; // Заголовок для виводу сум

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

double sum = 0;

for (int j = 0; j < n; j++) {

sum += matrix[i][j]; // Обчислюємо суму елементів рядка

}

fout << "Row " << i + 1 << ": " << sum << endl; // Записуємо суму

}

fin.close(); // Закриваємо вхідний файл

fout.close(); // Закриваємо вихідний файл

}

array\_utils.h

#ifndef ARRAY\_UTILS\_H

#define ARRAY\_UTILS\_H

#include <string>

using namespace std;

// Прототип функції для реверсу масиву чисел

void reverseArrayFromFile(const string& inputFilename, const string& outputFilename);

// Прототип функції для сортування масиву символів

void sortArrayFromFile(const string& inputFilename, const string& outputFilename);

#endif // ARRAY\_UTILS\_H

main.cpp

#include <iostream> // Підключаємо бібліотеку для вводу/виводу

#include <string> // Підключаємо бібліотеку для роботи з рядками

#include "array\_utils.h" // Підключаємо заголовочний файл для масивів

#include "matrix\_utils.h" // Підключаємо заголовочний файл для матриць

using namespace std; // Використовуємо стандартний простір імен

int main() {

string filename; // Змінна для збереження імені файлу

int task; // Змінна для вибору завдання

// Виводимо підказку та зчитуємо номер завдання

cout << "Enter task number (1 - reverse array, 2 - row sums, 3 - sort characters): ";

cin >> task;

cin.ignore(); // Очищуємо буфер вводу

// В залежності від номера завдання виконуємо відповідну дію

switch (task) {

case 1:

// Завдання 1: реверсування масиву чисел

cout << "Enter filename for input (array\_in\_28.txt): ";

getline(cin, filename); // Зчитуємо ім'я файлу з клавіатури

reverseArrayFromFile(filename, "array\_out\_28.txt"); // Викликаємо функцію реверсу

break;

case 2:

// Завдання 2: обчислення сум рядків матриці

cout << "Enter filename for input (matr\_in\_28.txt): ";

getline(cin, filename);

processMatrixFromFile(filename); // Обробляємо матрицю

break;

case 3:

// Завдання 3: сортування символів методом вибору

cout << "Enter filename for input (array\_in\_char\_28.txt): ";

getline(cin, filename);

sortArrayFromFile(filename, "array\_out\_char\_28.txt"); // Сортуємо символи

break;

default:

// Якщо ввели неправильний номер завдання

cout << "Wrong task number!" << endl;

}

return 0; // Завершення програми

}

matr\_in\_28.txt

3 4

1.5 2.5 -3.0 4.0

0.0 5.0 1.0 -2.0

3.3 -1.3 2.7 0.0

matrix\_utils.cpp

#include "array\_utils.h"

#include <fstream> // Для роботи з файлами

#include <vector> // Для використання динамічних масивів

#include <algorithm> // Для функції swap

#include <iostream>

// Функція реверсу масиву дійсних чисел з файлу

void reverseArrayFromFile(const string& inputFilename, const string& outputFilename) {

ifstream fin(inputFilename); // Відкриваємо файл для читання

ofstream fout(outputFilename); // Відкриваємо файл для запису

if (!fin.is\_open() || !fout.is\_open()) {

cout << "Error opening file." << endl;

return;

}

int n;

fin >> n; // Зчитуємо розмір масиву

vector<double> arr(n); // Створюємо вектор розміру n

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

fin >> arr[i]; // Зчитуємо елементи масиву

}

fout << n << endl; // Записуємо кількість елементів у файл

for (int i = n - 1; i >= 0; i--) {

fout << arr[i] << " "; // Записуємо елементи у зворотному порядку

}

fout << endl; // Перенос рядка

fin.close(); // Закриваємо вхідний файл

fout.close(); // Закриваємо вихідний файл

}

// Функція сортування символів за спаданням методом вибору

void sortArrayFromFile(const string& inputFilename, const string& outputFilename) {

ifstream fin(inputFilename); // Відкриваємо файл для читання

ofstream fout(outputFilename); // Відкриваємо файл для запису

if (!fin.is\_open() || !fout.is\_open()) {

cout << "Error opening file." << endl;

return;

}

int n;

fin >> n; // Зчитуємо розмір масиву

vector<char> arr(n); // Створюємо масив символів

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

fin >> arr[i]; // Зчитуємо символи

}

// Реалізація сортування вибором (за спаданням)

for (int i = 0; i < n - 1; i++) {

int maxIdx = i;

for (int j = i + 1; j < n; j++) {

if (arr[j] > arr[maxIdx]) {

maxIdx = j; // Знаходимо індекс максимуму

}

}

swap(arr[i], arr[maxIdx]); // Міняємо місцями

}

fout << n << endl; // Записуємо кількість елементів

for (char c : arr) {

fout << c << " "; // Записуємо відсортовані символи

}

fout << endl;

fin.close(); // Закриваємо вхідний файл

fout.close(); // Закриваємо вихідний файл

}

matrix\_utils.h

#ifndef MATRIX\_UTILS\_H

#define MATRIX\_UTILS\_H

#include <string>

using namespace std;

// Прототип функції обробки матриці (обчислення сум рядків)

void processMatrixFromFile(const string& filename);

#endif // MATRIX\_UTILS\_H

Додаток Б

Діаграми активності

На рисунку 1 зображено діаграму активності до завдання 1 функції

зчитування масиву з файлу

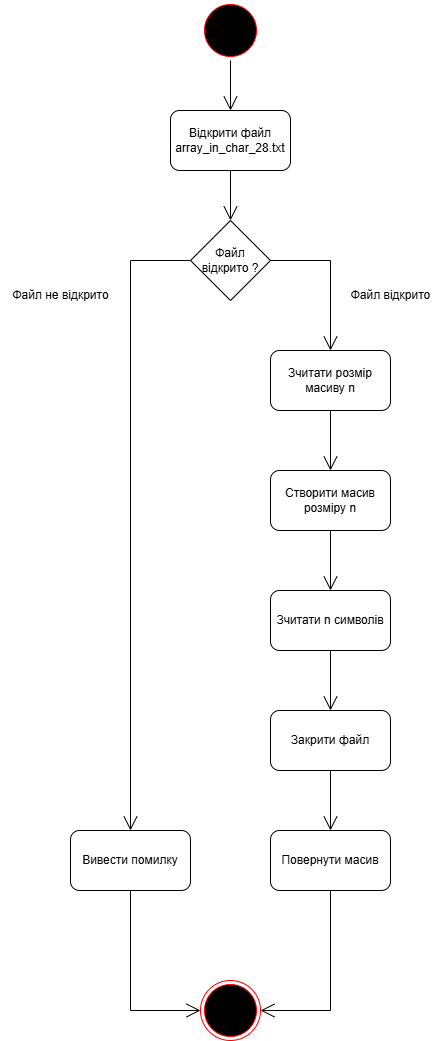


Рисунок 1 діаграма активності до завдання 1 функції зчитування масиву зчитування масиву з файлу

На рисунку 2 зображено діаграму активності до завдання 3 функції сортування

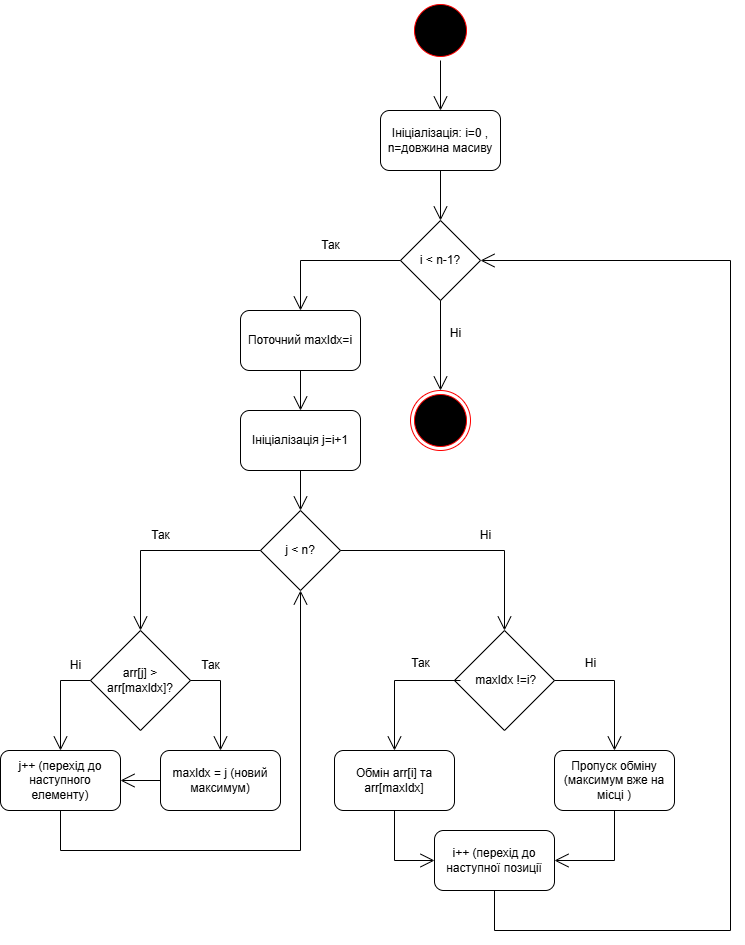


Рисунок 2 діаграма активності до завдання 3 функції сортування