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 процесів управління»

Тема: "Реалізація алгоритмів сортування та робота з файлами на мові С ++"

ХАІ.301.175.318.15 ЛР

Виконав студент гр. № 318

Мальківська В. А

(підпис, дата) (П.І.Б.)

Перевірив к.т.н., доцент

( вчена ступінь, вчене звання)

  Гавриленко О. В

(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.** За допомогою текстового редактору створити текстовий файл «array\_in\_n.txt» з елементами вихідного масиву (n - Array72). У

програмі на С++ зчитати і перетворити цей масив відповідно до свого варіанту завдання (див. лаб.роб.№6, завд.2), ім'я файлу і необхідні змінні ввести з консолі. Вивести результати у файл «array\_out\_n.txt».

**Array72.** Дано масив A розміру N і цілі числа K і L (1 ≤ K <L ≤ N). Переставити в зворотному порядку елементи масиву, розташовані між елементами AK і AL, включаючи ці елементи.

**Завдання 2.** За допомогою текстового редактору створити текстовий файл «matr\_in\_n.txt» з елементами вихідного двовимірного масиву (n - Matrix37). У програмі зчитати і обробити матрицю відповідно до свого

варіанту завдання (лаб.роб.№7, завд.1), ім'я файлу і необхідні змінні ввести з

консолі. Дописати результати в той же файл. Перетворити матрицю відповідно до свого варіанту завдання, розмір матриці і його елементи ввести з консолі. Вивести результати у консоль.

**Matrix37.** Дана цілочисельна матриця розміру M × N, елементи якої можуть набувати значень від 0 до 100. Різні стовпці матриці назвемо схожими, якщо збігаються безлічі чисел, що зустрічаються в цих стовпцях. Знайти кількість стовпців, схожих на останній стовпець даної матриці.

**Завдання 3.** Вивчити метод сортування відповідно до свого варіанту (див.табл. 1), проаналізувати його складність і продемонструвати на прикладі з 7-ми елементів (n - Version10). Реалізувати у вигляді окремої функції алгоритм сортування елементів масиву. Зчитування і виведення відсортованого масиву організувати на файлах.

**Version10.** Шейкерна, зменшення, символьний.

**Завдання 4.** Введення, виведення, обробку масивів реалізувати окремими функціями з параметрами. Структурувати проєкт програми для виконання завдань 1-3 наступним чином:

main.cpp //основна функція і три функції для 3х завдань

array\_utils.h //заголовки функцій для роботи з

//одновимірними масивами (завдання1,3)

array\_utils.cpp //визначення функцій для роботи з

//одновимірними масивами (завдання1,3)

matrix\_utils.h //заголовки функцій для роботи з

//двовимірними масивами (завдання2)

matrix\_utils.cpp //визначення функцій для роботи з

//двовимірними масивами (завдання2)

ВИКОНАННЯ РОБОТИ

**Завдання 1.**

Вирішення задачі (Array72).

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

1. Ім’я: n;

Опис: кількість елементів масиву;

Тип: ціле число;

Обмеження: n > 0;

2. Ім’я: mas;

Опис: елементи масиву;

Тип: дійсні числа;

Обмеження: довжина масиву дорівнює n;

3. Ім’я: K, L;

Опис: індекси перестановки;

Тип: цілі числа;

Обмеження: 1 ≤ K < L ≤ n;

Вихідні дані (ім’я, опис, тип, обмеження):

Ім’я: array\_out.txt;

Опис: файл, що містить оновлений масив після перестановки;

Тип: текстовий файл;

Алгоритм:

1. Зчитати розмір масиву n;

2. Створити динамічний масив mas з n елементів типу float;

3. Заповнити масив введеними користувачем значеннями;

4. Зчитати значення індексів K і L;

5. Виконати перестановку частини масиву з K по L у зворотному порядку, використовуючи reverse;

6. Записати оновлений масив у файл array\_out.txt.;

7. Вивести повідомлення про успішне виконання завдання;

8. Звільнити динамічну пам'ять;

Лістинг коду вирішення завдання 1 наведено в дод. А (стор. 10-11)

Екран роботи програми показаний на рис. Б.1 (дод. Б, стор.14)

Діаграма активності для завдання Array72 наведено на рис. Б.4 (дод. Б, стор.16)

**Завдання 2.**

Вирішення задачі (Matrix37).

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

1. Ім’я: M;

Опис: кількість рядків матриці;

Тип: ціле число;

Обмеження: M > 0;

2. Ім’я: N;

Опис: кількість стовпців матриці;

Тип: ціле число;

Обмеження: N > 0;

3. Ім’я: matrix;

Опис: цілочисельна матриця розміром M×N;

Тип: цілі числа;

Обмеження: розмір матриці повинен відповідати M × N;

Вихідні дані (ім’я, опис, тип, обмеження):

Ім’я: matr\_out.txt;

Опис: файл, що містить кількість однакових пар стовпців у матриці;

Тип: текстовий файл;

Алгоритм:

1. Зчитати значення M та N;

2. Виділити пам’ять під матрицю розміром M×N;

3. Заповнити матрицю значеннями, введеними користувачем;

4. Ініціалізувати лічильник count = 0;

5. Перебрати всі пари стовпців (j, k) у матриці;

6. Для кожної пари перевірити, чи всі елементи у відповідних рядках однакові;

7. Якщо стовпці однакові – збільшити count;

8. Записати результат у файл matr\_out.txt.;

9. Вивести повідомлення про завершення;

10. Звільнити виділену пам’ять;

Лістинг коду вирішення завдання 2 наведено в дод. А (стор. 11-12)

Екран роботи програми показаний на рис. Б.2 (дод. Б, стор. 14)

Діаграма активності для завдання Matrix37 наведено на рис. Б.5 (дод. Б, стор.17)

**Завдання 3.**

Вирішення задачі (Version10).

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

Ім’я: text;

Опис: рядок, введений користувачем;

Тип: рядковий;

Вихідні дані (ім’я, опис, тип, обмеження):

Ім’я: text\_out.txt;

Опис: файл, що містить реверсований рядок;

Тип: текстовий файл;

Алгоритм:

1. Вивести запрошення до введення рядка;

2. Зчитати увесь рядок за допомогою getline(cin, text);

3. Використати функцію reverse() для перевертання рядка;

4. Відкрити файл text\_out.txt для запису;

5. Записати реверсований рядок у файл;

6. Закрити файл;

7. Вивести повідомлення про завершення операції;

Лістинг коду вирішення завдання Version10 наведено в дод. А (стор. 15)

Екран роботи програми показаний на рис. Б.3 (дод. Б, стор.15)

Діаграма активності для завдання Version10 наведено на рис. Б.3 (дод. Б, стор.18)

**Завдання 4.**

Вирішення задачі (Меню).

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

1. Ім’я: масив;

Опис: вхідний масив цілих чисел, у якому потрібно виконати перестановку частини елементів;

Тип: масив цілих чисел;

2. Ім’я: матриця;

Опис: вхідна матриця цілих чисел, де потрібно знайти стовпці, що є схожими;

Тип: двовимірний масив цілих чисел;

3. Ім’я: рядок;

Опис: вхідний рядок символів, який потрібно відсортувати;

Тип: рядок символів;

Вихідні дані (ім’я, опис, тип, обмеження):

1. Ім’я: масив після перестановки;

Опис: після виконання перестановки частини масиву вивести змінений масив;

Тип: цілі числа;

2. Ім’я: стовпці;

Опис: вивести всі пари схожих стовпців у матриці;

Тип: список пар індексів стовпців;

3. Ім’я: рядок після сортування;

Опис: вивести відсортований рядок;

Тип: рядок символів;

Алгоритм:

Завдання 1: перестановка частини масиву;

* отримати від користувача масив і два індекси для перестановки;
* переставити елементи між вказаними індексами;
* вивести змінений масив;

Завдання 2: пошук схожих стовпців у матриці;

* ініціалізувати масив стовпців матриці;
* для кожної пари стовпців порівняти всі елементи у кожному рядку;
* якщо всі елементи однакові, вивести індекси стовпців як схожі;

Завдання 3: сортування символів;

* отримати строку від користувача;
* сортувати символи в рядках за допомогою стандартних алгоритмів сортування;
* вивести відсортовану строку;

Лістинг коду вирішення завдання 3 наведено в дод. А (стор. 12-13)

Діаграма активності для завдання 3 наведено на рис. Б.7 (дод. Б, стор.19)

ВИСНОВКИ

Під час вивчення роботи з масивами в C++ освоєно оголошення, введення, обробку та виведення на екран. Реалізація програм у Visual Studio допомогла краще зрозуміти, як це працює на практиці. Навчання роботі з файлами за допомогою бібліотеки fstream дозволило закріпити знання і уникнути помилок.

ДОДАТОК А

Лістинг до коду

#include <iostream> // Для введення/виведення даних

#include <fstream> // Для роботи з файлами

#include <algorithm> // Для алгоритмів (reverse, sort)

using namespace std; // Дозволяє використовувати стандартну бібліотеку без префікса std::

/\*

\* Для того, аби перевірити правильність написання завдань, натисніть 4 у головному меню.

\*/

// Декларація функцій

void executeTask1(); // Функція для перестановки частини масиву

void executeTask2(); // Функція для пошуку схожих стовпців у матриці

void executeTask3(); // Функція для сортування символів та обернення тексту

void showMenu(); // Функція для відображення меню програми

// ========================== Завдання 1: Обробка масиву ==========================

void executeTask1() {

int n; // Оголошення змінної для розміру масиву

cout << "Завдання 1: Перестановка частини масиву.\n"; // Виведення заголовку завдання

cout << "Введіть кількість елементів масиву: "; // Запит розміру масиву

cin >> n; // Введення розміру масиву

float\* mas = new float[n]; // Динамічний масив для зберігання елементів

cout << "Введіть елементи масиву: "; // Запит на введення елементів

for (int i = 0; i < n; i++) { // Цикл для заповнення масиву

cin >> mas[i]; // Введення елементів масиву

}

int K, L; // Змінні для індексів перестановки

cout << "Введіть K та L для перестановки (1 ≤ K < L ≤ " << n << "): "; // Запит індексів

cin >> K >> L; // Введення K і L

// Перестановка частини масиву у зворотному порядку

reverse(mas + K - 1, mas + L);

// Запис результату у файл

ofstream outputFile("array\_out.txt");

for (int i = 0; i < n; i++) { // Запис елементів у файл

outputFile << mas[i] << " ";

}

outputFile << endl; // Додаємо перехід на новий рядок у файлі

outputFile.close(); // Закриваємо файл

cout << "Завдання 1 виконано! Результат записано у файл array\_out.txt.\n"; // Повідомлення про завершення

delete[] mas; // Звільнення пам'яті

}

// ========================== Завдання 2: Пошук схожих стовпців у матриці ==========================

void executeTask2() {

int M, N; // Оголошення змінних для розмірів матриці

cout << "Завдання 2: Пошук схожих стовпців у матриці.\n"; // Виведення заголовку завдання

cout << "Введіть кількість рядків та стовпців матриці: "; // Запит на введення розмірів

cin >> M >> N; // Введення розмірів матриці

int\*\* matrix = new int\*[M]; // Створення масиву вказівників на рядки матриці

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

matrix[i] = new int[N]; // Виділення пам'яті для кожного рядка

}

cout << "Введіть елементи матриці:\n"; // Запит на введення елементів

for (int i = 0; i < M; i++) { // Цикл по рядках

for (int j = 0; j < N; j++) { // Цикл по стовпцях

cin >> matrix[i][j]; // Введення елементів матриці

}

}

int count = 0; // Лічильник схожих стовпців

for (int j = 0; j < N - 1; j++) { // Перебір усіх стовпців

for (int k = j + 1; k < N; k++) { // Перебір стовпців для порівняння

bool isEqual = true;

for (int i = 0; i < M; i++) { // Перевірка, чи рівні стовпці

if (matrix[i][j] != matrix[i][k]) {

isEqual = false;

break;

}

}

if (isEqual) count++; // Якщо стовпці рівні — збільшуємо лічильник

}

}

// Запис результату у файл

ofstream outputFile("matr\_out.txt");

outputFile << "Кількість схожих стовпців: " << count << endl;

outputFile.close(); // Закриваємо файл

cout << "Завдання 2 виконано! Результат записано у файл matr\_out.txt.\n"; // Повідомлення про завершення

// Звільнення пам'яті

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

delete[] matrix[i];

}

delete[] matrix;

}

// ========================== Завдання 3: Сортування символів ==========================

void executeTask3() {

string text;

cout << "Введіть символи для сортування: ";

getline(cin, text); // Зчитування всього рядка введеного користувачем

reverse(text.begin(), text.end()); // Сортування рядка, змінюючи порядок символів на зворотний

// Відкриття файлу для запису

ofstream textFile("text\_out.txt"); // Створення або відкриття файлу text\_out.txt для запису

textFile << text << endl; // Запис сорторованого тексту у файл

textFile.close(); // Закриття файлу після запису, щоб уникнути витоку ресурсів

cout << "Текст сорторовано та записано у файл text\_out.txt.\n"; // Виведення повідомлення про успішне виконання

}

// ========================== Головне меню програми ==========================

void showMenu() {

cout << "\n===== МЕНЮ =====\n";

cout << "1. Завдання 1: Перестановка частини масиву\n";

cout << "2. Завдання 2: Пошук схожих стовпців у матриці\n";

cout << "3. Завдання 3: Сортування символів\n";

cout << "4. Вихід\n";

cout << "Оберіть завдання (1-4): ";

}

int main() {

int choice;

while (true) {

showMenu();

cin >> choice;

switch (choice) {

case 1: executeTask1(); break;

case 2: executeTask2(); break;

case 3: executeTask3(); break;

case 4: cout << "Вихід з програми...\n"; return 0;

default: cout << "Невірний вибір! Введіть число від 1 до 4.\n"; break;

}

}

}

ДОДАТОК Б

Скрін-шоти вікна виконання програми

На рис. Б.1 показаний приклад виконання завдання Array72

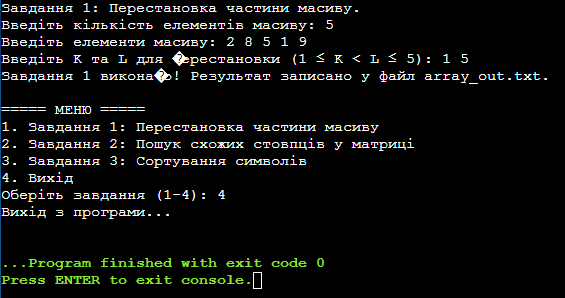


Рисунок Б.1 — Екран виконання програми для вирішення завдання Array72

На рис. Б.2 показаний приклад виконання завдання Matrix37

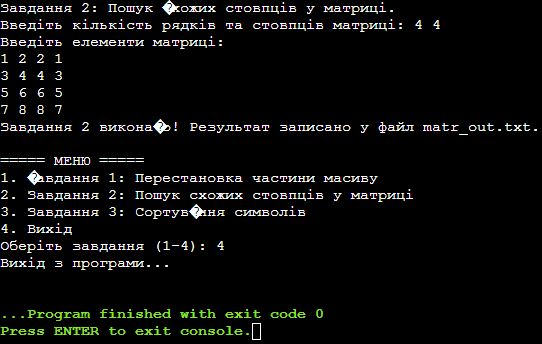


Рисунок Б.2 — Екран виконання програми для вирішення завдання Matrix37

На рис. Б.3 показаний приклад виконання завдання Version10

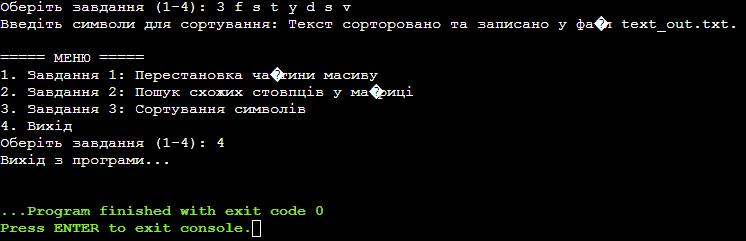


Рисунок Б.3 — Екран виконання програми для вирішення завдання Version10

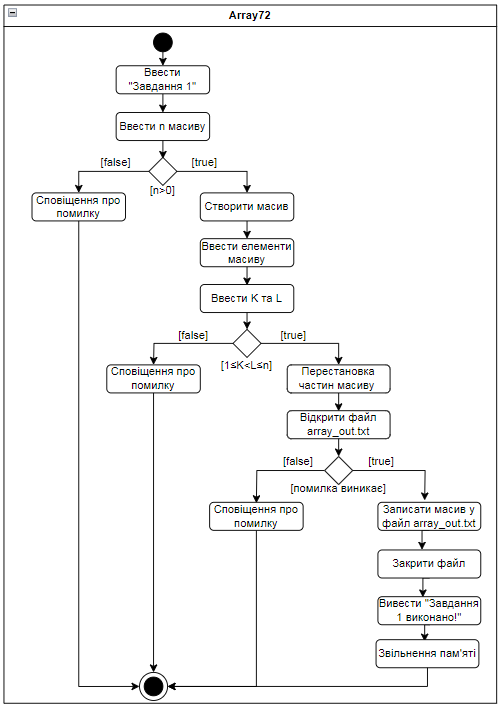


Рисунок Б.4 — Діаграма активності для завдання Array72

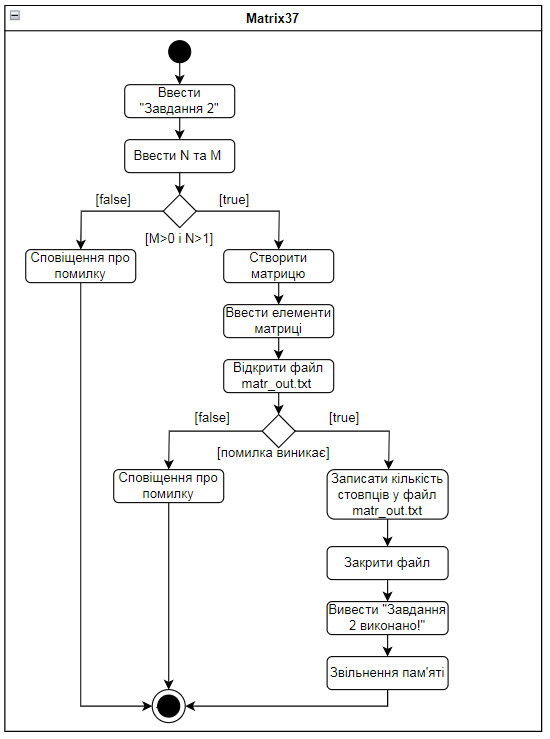


Рисунок Б.5 — Діаграма активності для завдання Matrix76

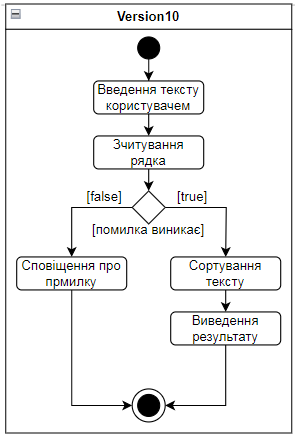


Рисунок Б.6 — Діаграма активності для завдання Version10

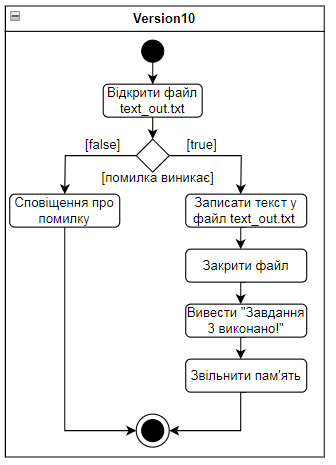


Рисунок Б.6 — Діаграма активності для завдання Version10

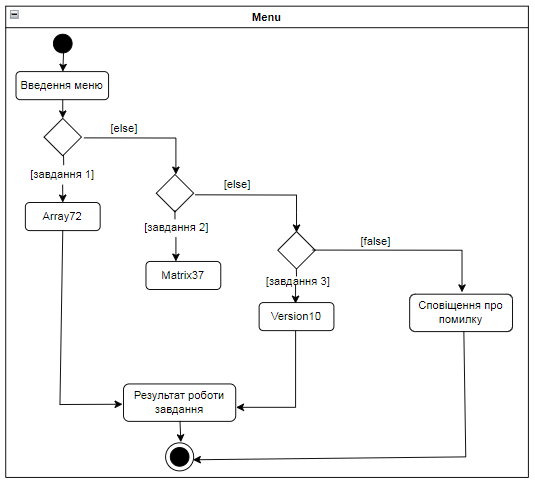


Рисунок Б.7 — Діаграма активності для завдання 4