

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»

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

ХАІ.301. 174. 319. 12 ЛР

Виконав студент гр. 319

\_\_\_\_\_  
(підпис, дата) Віталій НОВИКОВ  
(П.І.Б.)

Перевірів

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

## 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 (див. лаб.роб.№6, завд.2), ім'я файлу і необхідні змінні ввести з консолі. Вивести результати у файл «array\_out\_n.txt».

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 (лаб.роб.№7, завд.1), ім'я файлу і необхідні змінні ввести з консолі. Дописати результати в той же файл.

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.

## ВИКОНАННЯ РОБОТИ

Завдання 1. Видалити з масиву всі елементи, що зустрічаються менше трьох разів, і вивести розмір отриманого масиву і його вміст.

Вирішення задачі: task1().

Вхідні дані:

- Текстовий файл «array\_in\_12.txt».

Вихідні дані:

- Текстовий файл «array\_out\_12.txt».

Алгоритм вирішення:

- Отримуємо ім'я файлу з вхідним одновимірним масивом;
- Отримуємо ім'я файлу для виведення масиву після обробки;
- Отримуємо масив з файлу;
- Перевіряємо елементи масиву та видаляємо всі елементи які трапляються менше трьох разів;
- Виводимо отриманий масив в вихідний файл.

Завдання 2. Пошук елемента, який є максимальним у своєму рядку і мінімальним у своєму стовпці.

Вирішення задачі: task2().

Вхідні дані:

- Текстовий файл «matr\_in\_.txt».

Вихідні дані:

- Текстовий файл «matr\_out\_12.txt».

Алгоритм вирішення:

- Отримуємо ім'я файлу з вхідним двовимірним масивом;
- Отримуємо ім'я файлу для виведення масиву;
- Отримуємо масив з файлу;
- Шукаємо елемент який є максимальним у своєму рядку і мінімальним у своєму стовпці;
- Якщо такий елемент не знайдено, то в консоль виводимо «0»;
- Якщо такий елемент знайдено, то виводимо його в консоль;
- Виводимо масив в вихідний файл.

Завдання 3. Сортування методом вставки для масиву символів.

Вирішення задачі: task3().

Вхідні дані:

- Текстовий файл «sort\_in\_12.txt».

Вихідні дані:

- arr[n]: масив відсортованих елементів(тип int).

Алгоритм вирішення для функцій зчитування масиву з файлу представлений у графічному вигляді в додатку Б, рис. Б.1(стор.9).

Алгоритм вирішення для сортування представлений у графічному вигляді в додатку Б, рис. Б.2(стор.10).

Лістинг коду вирішення задач наведено в додатку А (стор. 4).

Екран роботи програми показаний в додатку Б, рис. Б.3(стор.11).

Вміст текстових файлів представлено в додатку Б, рис. Б.4(стор.12).

## ВИСНОВКИ

У ході виконання лабораторної роботи була розроблена програма на C++, яка складається з двох задач. Загалом, робота поглибила знання з програмування, покращила навички алгоритмічного мислення та навчила ефективно взаємодіяти з користувачем.

## ДОДАТОК А

### Лістинг коду програми

#### main.cpp

```
#include <iostream>
#include "my_array.h"
#include "my_matrix.h"
#include "my_sort.h"
using namespace std;

int main() {
    int choice;
    do {
        cout << "Меню:\n";
        cout << "1. Завдання 1: Обробка одновимірного масиву\n";
        cout << "2. Завдання 2: Обробка матриці\n";
        cout << "3. Завдання 3: Сортювання масиву\n";
        cout << "0. Вихід\n";
        cout << "Виберіть опцію: ";
        cin >> choice;

        switch (choice) {
            case 1:
                task_1();
                break;
            case 2:
                task_2();
                break;
            case 3:
                task_3();
                break;
            case 0:
                cout << "Вихід\n";
                break;
            default:
                cout << "Невірний вибір, спробуйте ще раз\n";
        }
    } while (choice != 0);
    return 0;
}
```

#### my\_array.h

```
#ifndef MY_ARRAY_H
#define MY_ARRAY_H

#include <string>
using namespace std;

// Прототипи функцій
void input_arr_from_file(const string& filename, int arr[], int &n); // Зчитування з файлу
void output_arr_to_file(const string& filename, const int arr[], int n); // Запис у файл
void remove_elements_less_than_three(int arr[], int &n); // Видалення елементів, що зустрічаються менше 3 разів
void task_1(); // Завдання 1

#endif
```

#### my\_array.cpp

```
#include <iostream>
#include <fstream>
#include <map>
#include "my_array.h"
```

```

using namespace std;
const int N = 50; // Максимальний розмір масиву

// Зчитування масиву з файлу
void input_arr_from_file(const string& filename, int arr[], int &n) {
    ifstream input_file(filename);
    if (!input_file) {
        cout << "Не вдалося відкрити файл для зчитування.\n";
        return;
    }
    input_file >> n;
    for (int i = 0; i < n; i++) {
        input_file >> arr[i];
    }
    input_file.close();
}

// Запис масиву у файл
void output_arr_to_file(const string& filename, const int arr[], int n) {
    ofstream output_file(filename);
    if (!output_file) {
        cout << "Не вдалося відкрити файл для запису.\n";
        return;
    }
    output_file << "Розмір масиву: " << n << "\n";
    for (int i = 0; i < n; i++) {
        output_file << arr[i] << " ";
    }
    output_file.close();
}

// Видалення елементів, що зустрічаються менше 3 разів
void remove_elements_less_than_three(int arr[], int &n) {
    map<int, int> freq;

    // Підрахунок частоти кожного елемента
    for (int i = 0; i < n; i++) {
        freq[arr[i]]++;
    }

    // Формування нового масиву
    int new_arr[N], new_size = 0;
    for (int i = 0; i < n; i++) {
        if (freq[arr[i]] >= 3) {
            new_arr[new_size++] = arr[i];
        }
    }

    // Оновлення вихідного масиву
    for (int i = 0; i < new_size; i++) {
        arr[i] = new_arr[i];
    }
    n = new_size;
}

// Завдання 1
void task_1() {
    int arr[N], n;
    string filename_in, filename_out;

    // Введення імен файлів
    cout << "Введіть ім'я вхідного файлу: ";
    cin >> filename_in;
    cout << "Введіть ім'я вихідного файлу: ";
    cin >> filename_out;

    // Зчитування масиву з файлу

```

```

input_arr_from_file(filename_in, arr, n);

// Обробка масиву: видалення елементів, що зустрічаються менше 3 разів
remove_elements_less_than_three(arr, n);

// Виведення результату в файл
output_arr_to_file(filename_out, arr, n);
}

```

### my\_matrix.h

```

#ifndef MY_MATRIX_H
#define MY_MATRIX_H

#include <string>
using namespace std;

// Прототипи функцій
void get_matrix_from_file(const string& filename, int matr[20][20], int &rows,
int &cols); // Зчитування з файлу
void output_matrix_to_file(const string& filename, const int matr[20][20], int
rows, int cols); // Запис у файл
int find_special_element(const int matr[20][20], int rows, int cols); // Пошук
спеціального елемента
void task_2(); // Завдання 2

#endif

```

### my\_matrix.cpp

```

#include <iostream>
#include <fstream>
#include "my_matrix.h"
using namespace std;

const int M = 20, N = 20; // Максимальні розміри матриці

// Зчитування матриці з файлу
void get_matrix_from_file(const string& filename, int matr[M][N], int &rows, int
&cols) {
    ifstream input_file(filename);
    if (!input_file) {
        cout << "Не вдалося відкрити файл для зчитування матриці.\n";
        return;
    }
    input_file >> rows >> cols;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            input_file >> matr[i][j];
        }
    }
    input_file.close();
}

// Запис матриці у файл
void output_matrix_to_file(const string& filename, const int matr[M][N], int
rows, int cols) {
    ofstream output_file(filename);
    if (!output_file) {
        cout << "Не вдалося відкрити файл для запису матриці.\n";
        return;
    }
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            output_file << matr[i][j] << " ";
        }
        output_file << endl;
    }
}

```

```

    }
    output_file.close();
}

// Пошук спеціального елемента
int find_special_element(const int matr[M][N], int rows, int cols) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            bool isMaxInRow = true, isMinInCol = true;

            // Перевіряємо, чи є поточний елемент максимальним у своєму рядку
            for (int k = 0; k < cols; k++)
                if (matr[i][k] > matr[i][j]) isMaxInRow = false;

            // Перевіряємо, чи є поточний елемент мінімальним у своєму стовпці
            for (int k = 0; k < rows; k++)
                if (matr[k][j] < matr[i][j]) isMinInCol = false;

            // Якщо обидві умови виконуються, повертаємо цей елемент
            if (isMaxInRow && isMinInCol) return matr[i][j];
        }
    }
    return 0;    // Якщо такого елемента немає, повертаємо 0
}

// Завдання 2
void task_2() {
    int matr[M][N], rows, cols;
    string filename_in, filename_out;

    // Введення імен файлів
    cout << "Введіть ім'я вхідного файлу для матриці: ";
    cin >> filename_in;
    cout << "Введіть ім'я вихідного файлу для матриці: ";
    cin >> filename_out;

    // Зчитування матриці з файлу
    get_matrix_from_file(filename_in, matr, rows, cols);

    // Пошук спеціального елемента
    int special_element = find_special_element(matr, rows, cols);

    // Запис результату в файл
    output_matrix_to_file(filename_out, matr, rows, cols);
    cout << "Спеціальний елемент: " << (special_element ?
to_string(special_element) : "0") << endl;
}

```

### my\_sort.h

```

#ifndef MY_SORT_H
#define MY_SORT_H

#include <string>
using namespace std;
// Прототипи функцій
void insertionSort(char arr[], int n); // Функція сортування методом вставки
bool readArrayFromFile(const string& filename, char arr[], int& n); //
Зчитування масиву з файлу
void printArray(const char arr[], int n); // Виведення масиву

void task_3(); // Завдання 3: Виконання сортування

#endif

```

### my\_sort.cpp

```

#include <iostream>
#include <fstream>
#include "my_sort.h"
using namespace std;
// Сортування методом вставки для масиву символів
void insertionSort(char arr[], int n) {
    for (int i = 1; i < n; ++i) {
        char key = arr[i];
        int j = i - 1;

        // Переміщуємо елементи масиву, що більші за key, на одну позицію вправо
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            --j;
        }
        arr[j + 1] = key;
    }
}
// Зчитування масиву з файлу
bool readArrayFromFile(const string& filename, char arr[], int& n) {
    ifstream inputFile(filename);
    if (!inputFile) {
        cerr << "Не вдалося відкрити файл для зчитування!" << endl;
        return false;
    }

    n = 0;
    while (inputFile >> arr[n]) {
        ++n;
    }

    inputFile.close();
    return true;
}
// Виведення масиву
void printArray(const char arr[], int n) {
    for (int i = 0; i < n; ++i) {
        cout << arr[i] << " ";
    }
    cout << endl;
}

// Завдання 3: Сортування масиву
void task_3() {
    const int MAX_SIZE = 7; // Максимальна кількість елементів
    char arr[MAX_SIZE];
    int n;

    string filename;
    cout << "Введіть ім'я файлу для зчитування масиву: ";
    cin >> filename;

    // Зчитуємо масив з файлу
    if (!readArrayFromFile(filename, arr, n)) {
        return;
    }

    cout << "Масив до сортування: ";
    printArray(arr, n);

    // Сортуємо масив методом вставки
    insertionSort(arr, n);

    cout << "Відсортований масив: ";
    printArray(arr, n);
}

```



## ДОДАТОК Б

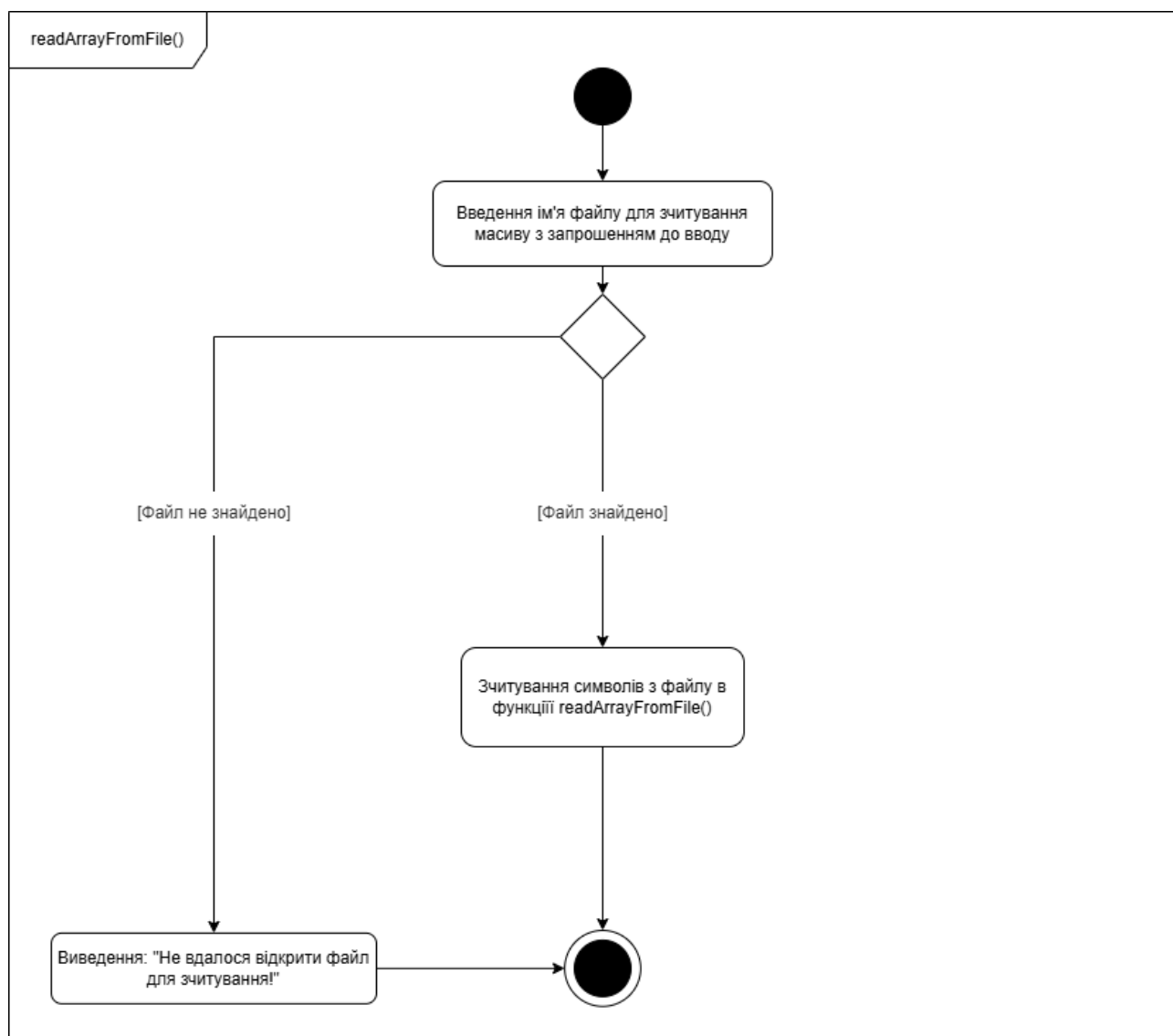


Рисунок Б.1 – діаграма активності зчитування масиву з файлу.

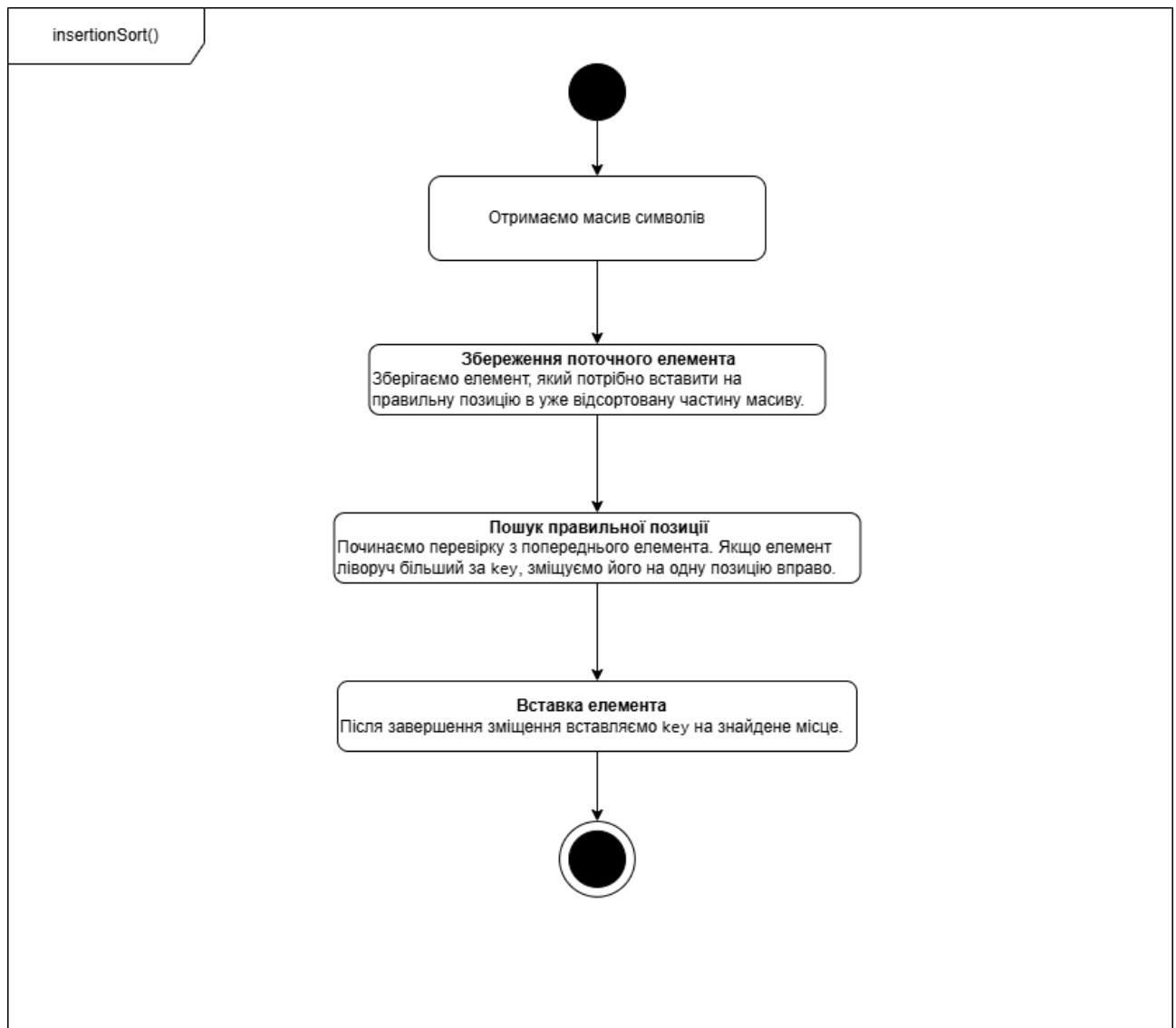


Рисунок Б.2 – Діаграма активності функції insertionSort().

```
Меню:
1. Завдання 1: Обробка одновимірного масиву
2. Завдання 2: Обробка матриці
3. Завдання 3: Сортування масиву
0. Вихід
Виберіть опцію: 1
Введіть ім'я вхідного файлу: array_in_12.txt
Введіть ім'я вихідного файлу: array_out_12.txt
Меню:
1. Завдання 1: Обробка одновимірного масиву
2. Завдання 2: Обробка матриці
3. Завдання 3: Сортування масиву
0. Вихід
Виберіть опцію: 2
Введіть ім'я вхідного файлу для матриці: matr_in_12.txt
Введіть ім'я вихідного файлу для матриці: matr_out_12.txt
Спеціальний елемент: 6
Меню:
1. Завдання 1: Обробка одновимірного масиву
2. Завдання 2: Обробка матриці
3. Завдання 3: Сортування масиву
0. Вихід
Виберіть опцію: 3
Введіть ім'я файлу для зчитування масиву: sort_in_12.txt
Масив до сортування: a d b c e g f
Відсортований масив: a b c d e f g
Меню:
1. Завдання 1: Обробка одновимірного масиву
2. Завдання 2: Обробка матриці
3. Завдання 3: Сортування масиву
0. Вихід
Виберіть опцію: 0
Вихід

...Program finished with exit code 0
Press ENTER to exit console.
```

Рисунок Б.3 – Екран роботи програми.

The image displays three screenshots of a code editor, each showing the contents of a different text file in a project. The project files are listed in the top bar: main.cpp, array\_in\_12.txt, array\_out\_12.txt, matr\_in\_12.txt, matr\_out\_12.txt, my\_array.cpp, and my\_array.h.

**First Screenshot (array\_in\_12.txt):**

```

1 5
2 1 1 1 4 5

```

**Second Screenshot (array\_out\_12.txt):**

```

1 Розмір масиву: 3
2 1 1 1

```

**Third Screenshot (matr\_out\_12.txt):**

```

1 1 4 6 2 1
2 2 3 54 3 4
3 3 2 92 6 4
4

```

Below the grid of numbers, there is a row of letters: **1 a d b c e g f**, where the letter 'a' is highlighted with a green vertical line.

Рисунок Б.4 – Вміст текстових файлів.