

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.301.173.310.1 ЛР

Виконав студент гр. 310

Микола АНДРЮШКІН
(підпис, дата) (П.І.Б.)

Перевірів

 к.т.н., доц. Олена ГАВРИЛЕНКО
(підпис, дата) (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_1.txt» з елементами вихідного масиву. У програмі на C++ зчитати і перетворити цей масив відповідно до свого варіанту завдання. Дан цілочисельний масив розміру N. Потроїти в ньому входження всіх непарних чисел., 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_1.txt» з елементами вихідного двовимірного масиву. У програмі зчитати і обробити матрицю відповідно до свого варіанту завдання. Дана цілочисельна матриця розміру $M \times N$. Знайти елемент, який є максимальним у своєму рядку і мінімальним в своєму стовпці. Якщо такий елемент відсутній, то вивести 0. Ім'я файлу і необхідні змінні ввести з консолі. Дописати результати в той же файл.

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

Завдання 4. Введення, виведення, обробку масивів реалізувати окремими функціями з параметрами. Структурувати проєкт програми для виконання завдань 1-3 на декілька підфайлів.

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Array 111

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

array_in_1.txt — файл з цілочисельним масивом, int, нема обмежень у розмірі матриці.

Вихідні дані (ім'я, опис, тип):

array_in_1.txt — файл з цілочисельним масивом та дописаним у кінці результатом, int, нема обмежень у розмірі матриці.

Алгоритм вирішення taskArray111() показано на рис. 1, readArrayFromFile() на рис. 2, expandOddNumbers() на рис. 3, writeResultsToFile() на рис. 4.

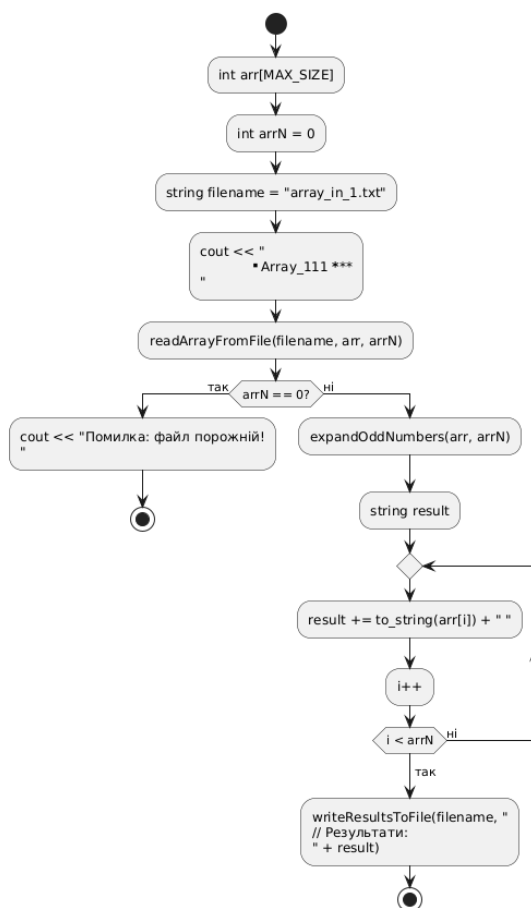
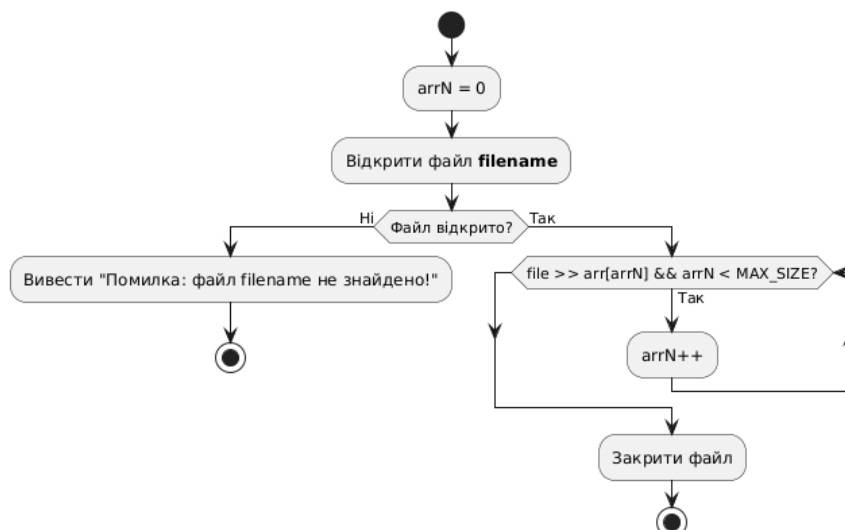
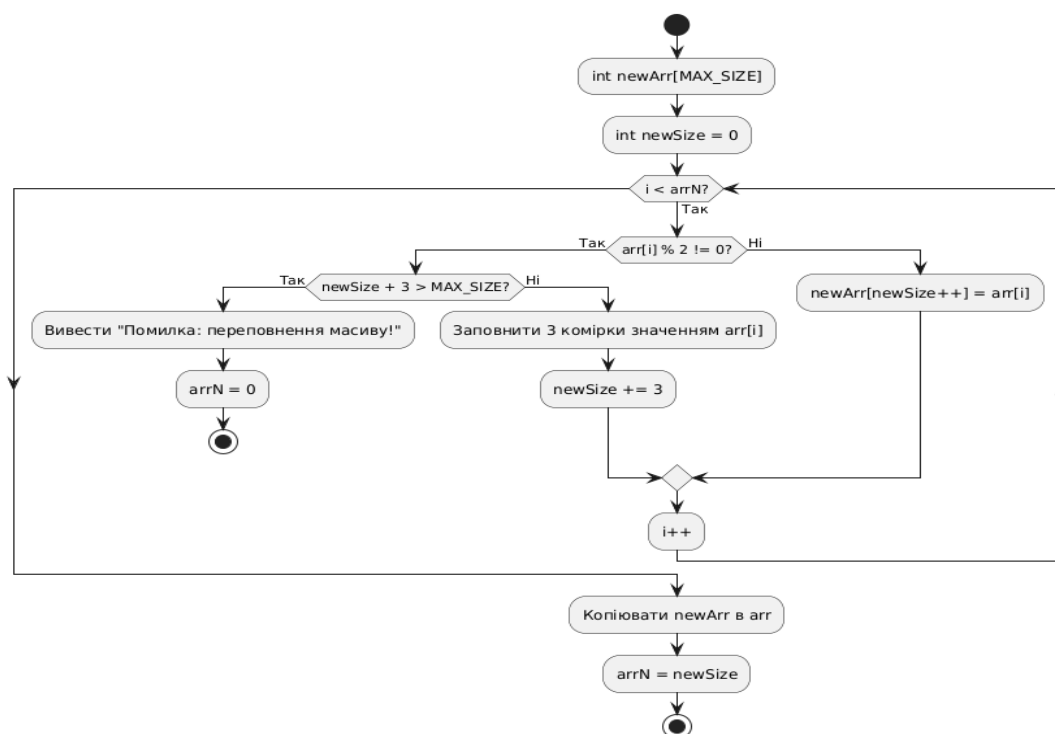


Рисунок 1 – taskArray111()

Рисунок 2 – `readArrayFromFile()`Рисунок 3 – `expandOddNumbers()`

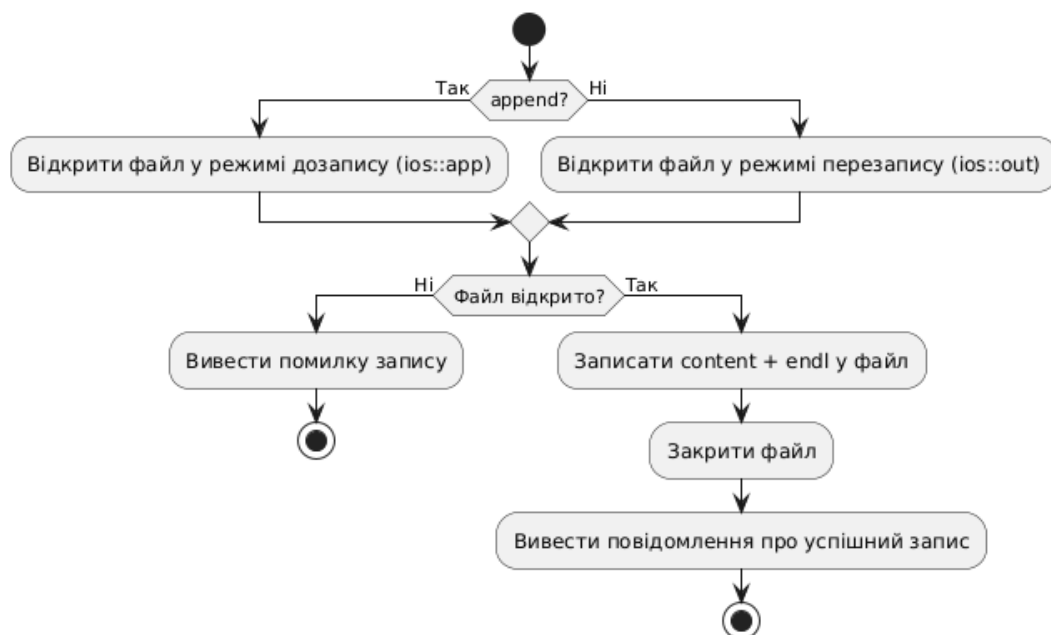


Рисунок 4 – writeResultsToFile()

Лістинг коду вирішення задачі розділ і номер задач(і) наведено в дод. А (стор. 13).

Екран роботи програми показаний на рис. Б.1.

Завдання 2.

Вирішення задачі taskMatrix 46

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

matr_in_1.txt — файл з цілочисельною матрицею, int, нема обмежень у розмірі матриці.

список імен змінних з описом, типами даних, та діапазонами допустимих значень

Вихідні дані (ім'я, опис, тип):

matr_in_1.txt — файл з цілочисельною матрицею та дописаним у кінці результатом, int, нема обмежень у розмірі матриці.

список імен змінних з описом, типами даних, або відповідних текстових сповіщень (див. Приклад в LPN^{№1})

Алгоритм вирішення taskMatrix46() показано на рис. 5, readMatrixFromFile() на рис. 6, findSaddlePoints() на рис. 7, writeResultsToFile() на рис. 4.

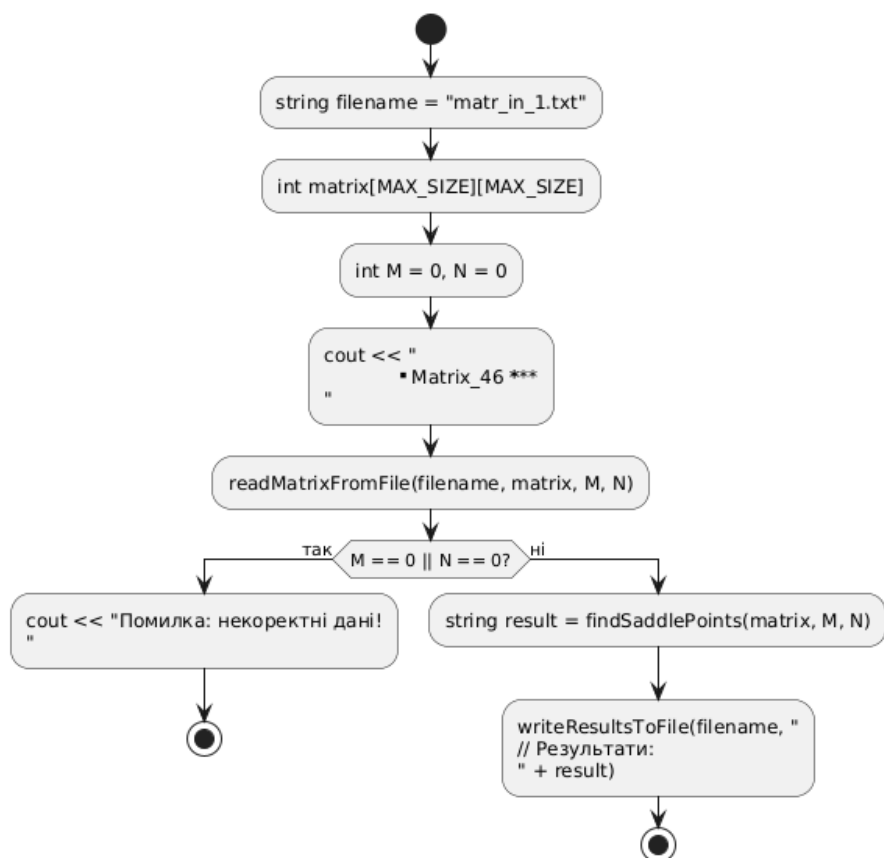


Рисунок 5 – taskMatrix46()

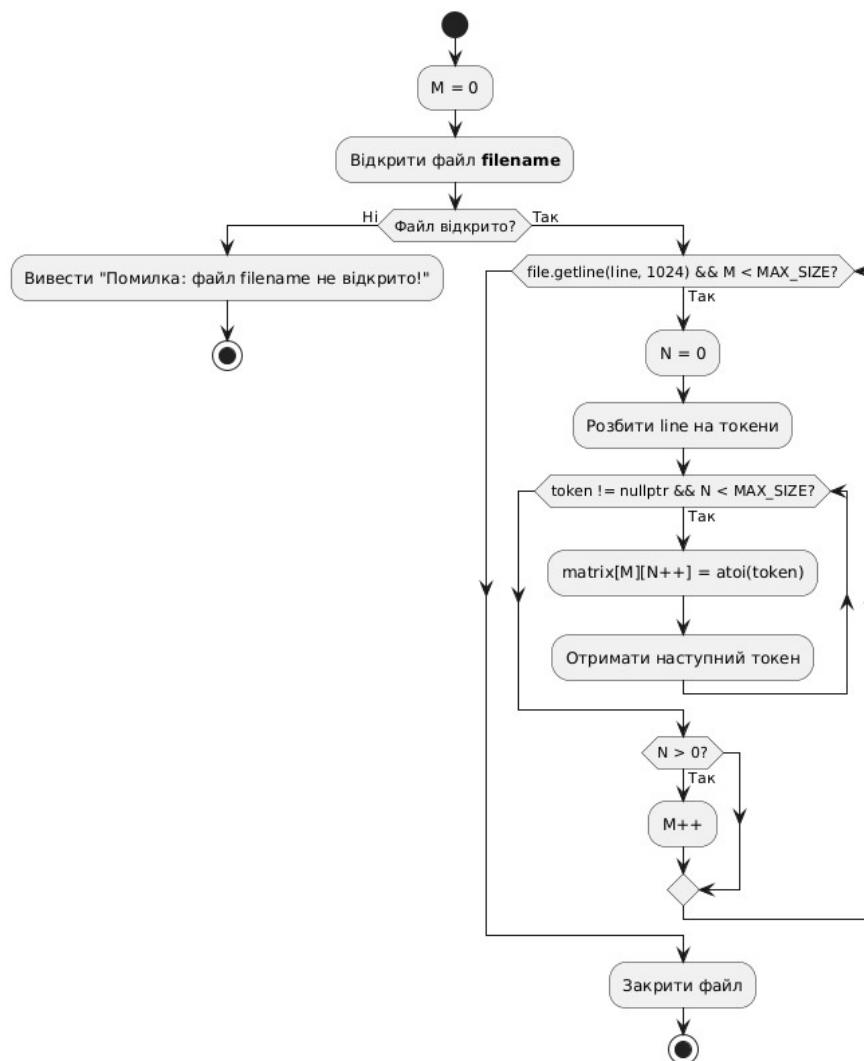


Рисунок 6 – readMatrixFromFile()

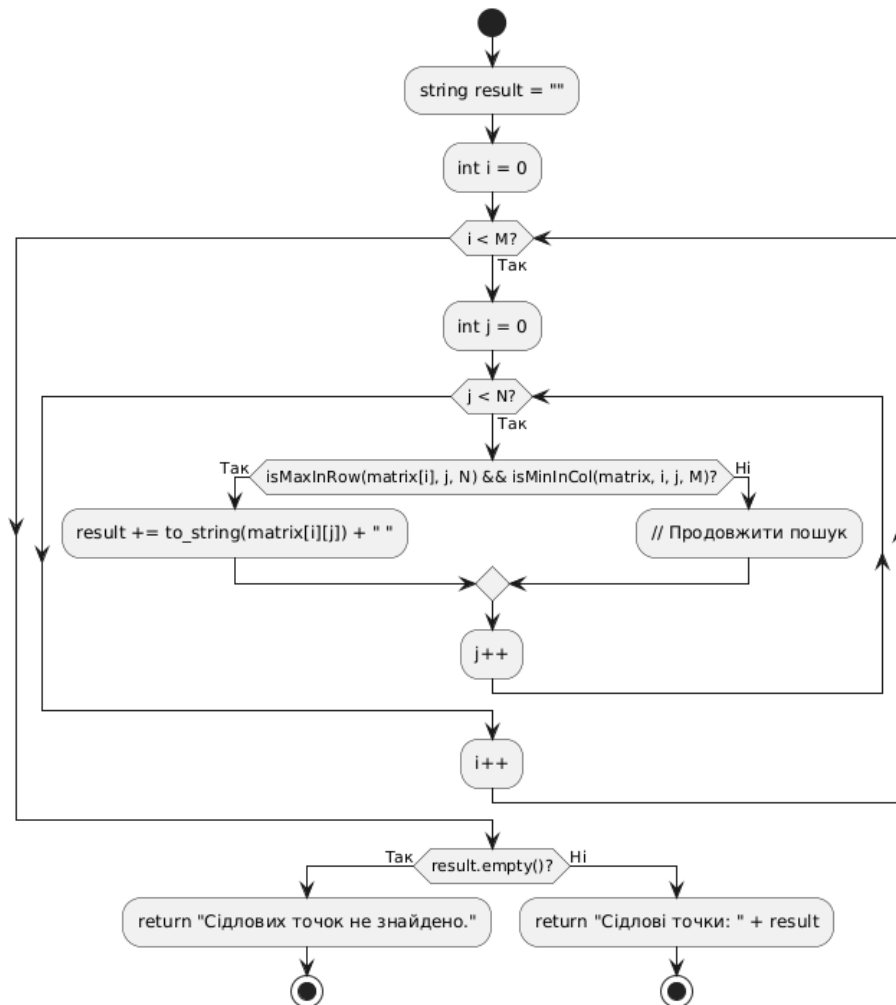


Рисунок 7 – findSaddlePoints()

Лістинг коду вирішення задачі розділ і номер задач(i) наведено в дод. А (стор. 13).

Екран роботи програми показаний на рис. Б.2.

Завдання 3.

Вирішення задачі taskSort.

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

letters_1.txt — файл з масивом типу char, нема обмежень у розмірі матриці.

список імен змінних з описом, типами даних, та діапазонами допустимих значень

Вихідні дані (ім'я, опис, тип):

letters_1.txt — файл з масивом типу char та дописаним у кінці результатом, нема обмежень у розмірі матриці.

список імен змінних з описом, типами даних, або відповідних текстових сповіщень (див. Приклад в LPN№1)

Алгоритм вирішення taskSort() показано на рис. 8, readCharsFromFile() на рис. 9, insertionSort() на рис. 10, writeResultsToFile() на рис. 4.

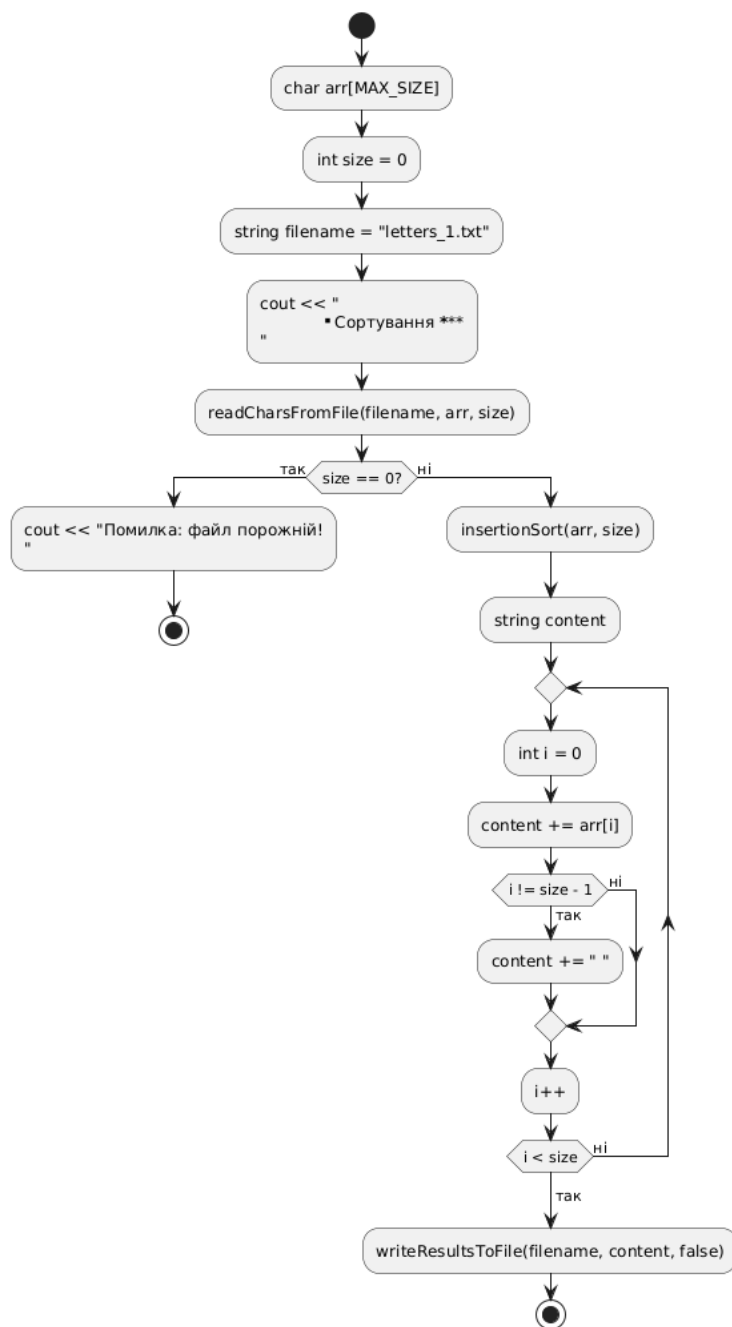
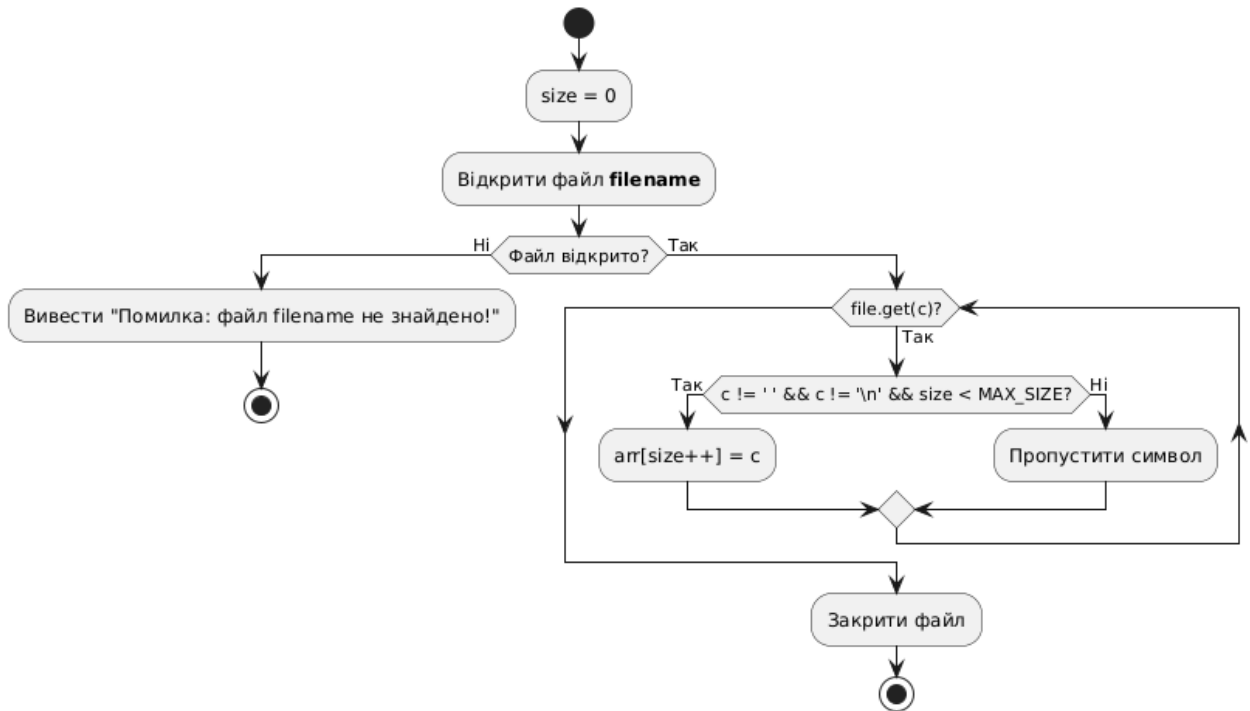
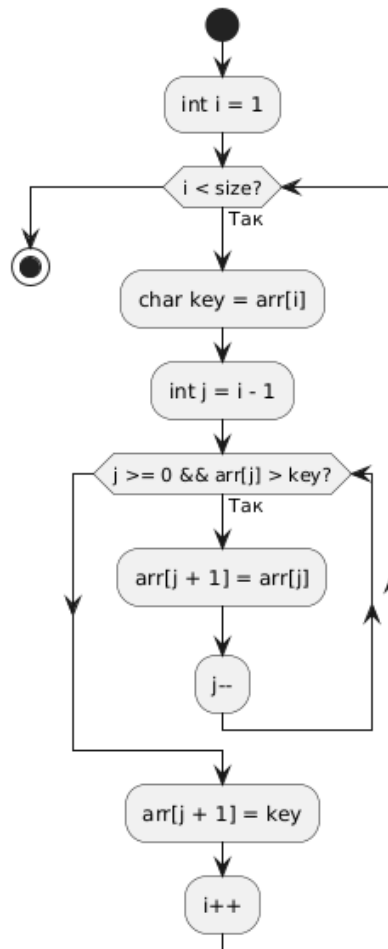


Рисунок 8 – taskSort()

Рисунок 9 – `readCharsFromFile()`Рисунок 10 – `insertionSort()`

Лістинг коду вирішення задачі розділ і номер задач(і) наведено в дод. А (стор. 13).

Екран роботи програми показаний на рис. Б.3.

ВИСНОВКИ

Закріплені на практиці алгоритми сортування та їх реалізація у коді. Відпрацьовано в коді програми робота з масивами та матрицями. Отримано навички з реалізації алгоритмів сортування та розбиття програми на декілька файлів.

ДОДАТОК А

Лістинг коду програми

```

main.cpp:
#include "data_io.h"
#include "processing.h"
#include <iostream>
using namespace std;

void taskMatrix46();
void taskArray111();
void taskSort();
void displayMenu();

int main() {
    int choice;
    while (true) {
        displayMenu();
        cin >> choice;
        switch (choice) {
            case 1: taskMatrix46(); break;
            case 2: taskArray111(); break;
            case 3: taskSort(); break;
            case 4: cout << "Вихід...\n"; return 0;
            default: cout << "Невірний вибір!\n";
        }
    }
    return 0;
}

void taskMatrix46() {
    cout << "\n***** Matrix_46 *****\n";
    int matrix[MAX_SIZE][MAX_SIZE], M = 0, N = 0;
    string filename = "matr_in_1.txt";
    readMatrixFromFile(filename, matrix, M, N);

    if (M == 0 || N == 0) {
        cout << "Помилка: некоректні дані!\n";
        return;
    }

    string result = findSaddlePoints(matrix, M, N);
    writeResultsToFile(filename, "\n// Результати:\n" + result);
}

```

```

void taskArray111() {
    cout << "\n***** Array_111 *****\n";
    int arr[MAX_SIZE], arrN = 0;
    string filename = "array_in_1.txt";
    readArrayFromFile(filename, arr, arrN);

    if (arrN == 0) {
        cout << "Помилка: файл порожній!\n";
        return;
    }

    expandOddNumbers(arr, arrN);
    string result;
    for (int i = 0; i < arrN; i++) result += to_string(arr[i]) + " ";
    writeResultsToFile(filename, "\n// Результати:\n" + result);
}

void taskSort() {
    cout << "\n***** Сортювання *****\n";
    char arr[MAX_SIZE];
    int size = 0;
    string filename = "letters_1.txt";

    readCharsFromFile(filename, arr, size);
    if (size == 0) {
        cout << "Помилка: файл порожній!\n";
        return;
    }

    insertionSort(arr, size);

    string content;
    for (int i = 0; i < size; i++) {
        content += arr[i];
        if (i != size - 1) content += " ";
    }

    writeResultsToFile(filename, content, false);
}

void displayMenu() {
    cout << "\nМеню:\n"
        << "1. Обробити матрицю (Matrix_46)\n"
        << "2. Обробити масив (Array_111)\n"
        << "3. Відсортувати символи\n"
        << "4. Вийти\n"
        << "Ваш вибір: ";
}

```

```

data_io.cpp
#include "data_io.h"
#include <fstream>
#include <iostream>
#include <cstring>
using namespace std;

void readArrayFromFile(const string& filename, int arr[], int& arrN) {
    arrN = 0;
    ifstream file(filename);
    if (!file.is_open()) {
        cout << "Помилка: файл " << filename << " не знайдено!\n";
        return;
    }
    while (file >> arr[arrN] && arrN < MAX_SIZE) arrN++;
    file.close();
}

void readMatrixFromFile(const string& filename, int matrix[][MAX_SIZE], int& M,
int& N) {
    M = 0;
    char line[1024];
    ifstream file(filename);
    if (!file.is_open()) {
        cout << "Помилка: файл " << filename << " не відкрито!\n";
        return;
    }
    while (file.getline(line, 1024) && M < MAX_SIZE) {
        N = 0;
        char* token = strtok(line, " ");
        while (token != nullptr && N < MAX_SIZE) {
            matrix[M][N++] = atoi(token);
            token = strtok(nullptr, " ");
        }
        if (N > 0) M++;
    }
    file.close();
}

void writeResultsToFile(const string& filename, const string& content, bool
append) {
    ofstream file;
    file.open(filename, append ? ios::app : ios::out);
    if (!file.is_open()) {
        cout << "Помилка запису у файл: " << filename << endl;
        return;
    }
    file << content << endl;
    file.close();
    cout << "Результати дописано у файл: " << filename << endl;
}

```

```

}

void readCharsFromFile(const string& filename, char arr[], int& size) {
    size = 0;
    ifstream file(filename);
    if (!file.is_open()) {
        cout << "Помилка: файл " << filename << " не знайдено!\n";
        return;
    }
    char c;
    while (file.get(c)) {
        if (c != ' ' && c != '\n' && size < MAX_SIZE) {
            arr[size++] = c;
        }
    }
    file.close();
}

```

```

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

```

```

void readArrayFromFile(const string& filename, int arr[], int& arrN) {
    arrN = 0;
    ifstream file(filename);
    if (!file.is_open()) {
        cout << "Помилка: файл " << filename << " не знайдено!\n";
        return;
    }
    while (file >> arr[arrN] && arrN < MAX_SIZE) arrN++;
    file.close();
}

```

```

void readMatrixFromFile(const string& filename, int matrix[][MAX_SIZE], int& M,
int& N) {
    M = 0;
    char line[1024];
    ifstream file(filename);
    if (!file.is_open()) {
        cout << "Помилка: файл " << filename << " не відкрито!\n";
        return;
    }
    while (file.getline(line, 1024) && M < MAX_SIZE) {
        N = 0;
        char* token = strtok(line, " ");
        while (token != nullptr && N < MAX_SIZE) {

```



```

        matrix[M][N++] = atoi(token);
        token = strtok(nullptr, " ");
    }
    if (N > 0) M++;
}
file.close();
}

void writeResultsToFile(const string& filename, const string& content, bool
append) {
    ofstream file;
    file.open(filename, append ? ios::app : ios::out);
    if (!file.is_open()) {
        cout << "Помилка запису у файл: " << filename << endl;
        return;
    }
    file << content << endl;
    file.close();
    cout << "Результати дописано у файл: " << filename << endl;
}

void readCharsFromFile(const string& filename, char arr[], int& size) {
    size = 0;
    ifstream file(filename);
    if (!file.is_open()) {
        cout << "Помилка: файл " << filename << " не знайдено!\n";
        return;
    }
    char c;
    while (file.get(c)) {
        if (c != ' ' && c != '\n' && size < MAX_SIZE) {
            arr[size++] = c;
        }
    }
    file.close();
}

processing.cpp
#include "processing.h"
#include <algorithm>
#include <string>
#include <iostream>
using namespace std;

void expandOddNumbers(int arr[], int& arrN) {
    int newArr[MAX_SIZE];
    int newSize = 0;
    for (int i = 0; i < arrN; i++) {
        if (arr[i] % 2 != 0) {

```

```

        if (newSize + 3 > MAX_SIZE) {
            cout << "Помилка: переповнення масиву!\n";
            arrN = 0;
            return;
        }
        fill_n(newArr + newSize, 3, arr[i]);
        newSize += 3;
    } else {
        newArr[newSize++] = arr[i];
    }
}
copy(newArr, newArr + newSize, arr);
arrN = newSize;
}

bool isMaxInRow(int row[], int colIndex, int cols) {
    for (int j = 0; j < cols; j++) {
        if (row[j] > row[colIndex]) return false;
    }
    return true;
}

bool isMinInCol(int matrix[][MAX_SIZE], int rowIndex, int colIndex, int rows) {
    for (int i = 0; i < rows; i++) {
        if (matrix[i][colIndex] < matrix[rowIndex][colIndex]) return false;
    }
    return true;
}

string findSaddlePoints(int matrix[][MAX_SIZE], int M, int N) {
    string result;
    for (int i = 0; i < M; i++) {
        for (int j = 0; j < N; j++) {
            if (isMaxInRow(matrix[i], j, N) && isMinInCol(matrix, i, j, M)) {
                result += to_string(matrix[i][j]) + " ";
            }
        }
    }
    return result.empty() ? "Сідлових точок не знайдено." : "Сідлові точки: " +
result;
}

void insertionSort(char arr[], int size) {
    for (int i = 1; i < size; ++i) {
        char key = arr[i];
        int j = i - 1;
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j--;
        }
    }
}

```

```

        }
        arr[j + 1] = key;
    }
}

```

processing.h

```

#ifndef PROCESSING_H
#define PROCESSING_H

```

```

#include "data_io.h"
#include <string>    // Додано для типу string
using namespace std; // Додано для string без std::

```

```

// Прототипи функцій
void insertionSort(char arr[], int size);

```

```

bool isMaxInRow(int row[], int colIndex, int cols);
bool isMinInCol(int matrix[][MAX_SIZE], int rowIndex, int colIndex, int rows);
string findSaddlePoints(int matrix[][MAX_SIZE], int M, int N);
void expandOddNumbers(int arr[], int& arrN);

```

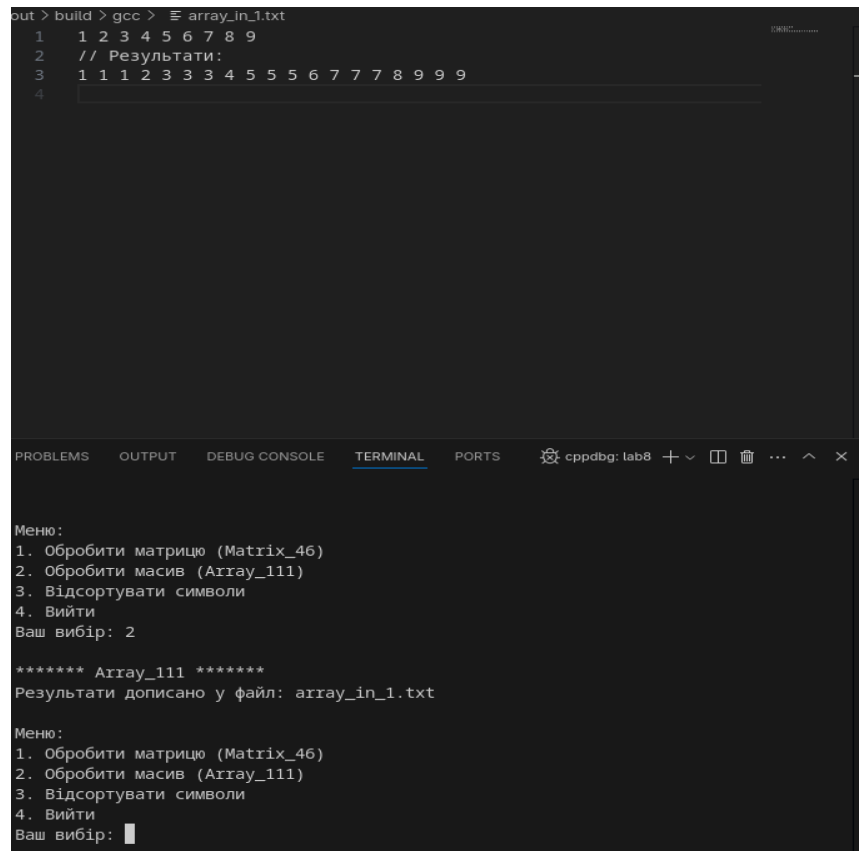
```

#endif

```

ДОДАТОК Б

Скрін-шоти вікна виконання програми



```
out > build > gcc > array_in_1.txt
1 1 2 3 4 5 6 7 8 9
2 // Результати:
3 1 1 1 2 3 3 3 4 5 5 5 6 7 7 7 8 9 9 9
4

ПРОБЛЕМЫ OUTPUT DEBUG CONSOLE TERMINAL PORTS cppdbg: lab8 + - [ ] ... ^ x

Меню:
1. Обробити матрицю (Matrix_46)
2. Обробити масив (Array_111)
3. Відсортувати символи
4. Вийти
Ваш вибір: 2

***** Array_111 *****
Результати дописано у файл: array_in_1.txt

Меню:
1. Обробити матрицю (Matrix_46)
2. Обробити масив (Array_111)
3. Відсортувати символи
4. Вийти
Ваш вибір: 
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання

Array 111

```

out > build > gcc > ≡ matr_in_1.txt
1 1 2 3
2 4 5 6
3 7 8 9
4 // Результати:
5 Сідлові точки: 3
6

МЕНЮ:
1. Обробити матрицю (Matrix_46)
2. Обробити масив (Array_111)
3. Відсортувати символи
4. Вийти
Ваш вибір: 1

***** Matrix_46 *****
Результати дописано у файл: matr_in_1.txt

МЕНЮ:
1. Обробити матрицю (Matrix_46)
2. Обробити масив (Array_111)
3. Відсортувати символи
4. Вийти
Ваш вибір: 

```

Рисунок Б.2 – Екран виконання програми для вирішення завдання
Matrix 46

```

out > build > gcc > ≡ letters_1.txt
1 d f f g g h i k l m n n p r s s v x y
2

МЕНЮ:
1. Обробити матрицю (Matrix_46)
2. Обробити масив (Array_111)
3. Відсортувати символи
4. Вийти
Ваш вибір: 3

***** Сортування *****
Результати дописано у файл: letters_1.txt

МЕНЮ:
1. Обробити матрицю (Matrix_46)
2. Обробити масив (Array_111)
3. Відсортувати символи
4. Вийти
Ваш вибір: 

```

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