

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.301.174.319.23ЛР

Виконав студент гр. 319

Шаньгин А.А.
(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_87.txt» з елементами вихідного масиву. У програмі на C++ зчитати і перетворити цей масив відповідно до свого варіанту завдання, ім'я файлу і необхідні змінні ввести з консолі. Вивести результати у файл «array_out_87.txt».

Array87	Дан масив розміру N, всі елементи якого, крім першого, впорядковані за зростанням. Зробити масив упорядкованим, перемістивши перший елемент на нову позицію.
----------------	--

Завдання 2. За допомогою текстового редактору створити текстовий файл «matr_in_35.txt» з елементами вихідного двовимірного масиву. У програмі зчитати і обробити матрицю відповідно до свого варіанту завдання, ім'я файлу і необхідні змінні ввести з консолі. Дописати результати в той же файл.

Matrix35	Дана цілочисельна матриця розміру $M \times N$. Знайти номер першого з її стовпців, що містять тільки непарні числа. Якщо таких стовпців немає, то вивести 0.
-----------------	--

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

12	Двійкові вставки	Зменшення	Цілий
-----------	------------------	-----------	-------

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 Array87

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

arr – масив цілих;

Вихідні дані (ім'я, опис, тип):

arr – перетворений масив цілих елементів

Алгоритм вирішення показано на рис. 1

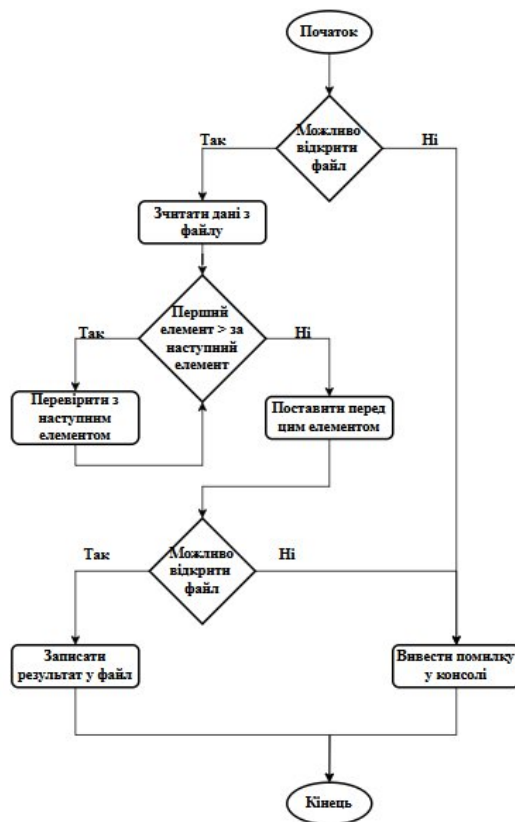


Рисунок 1 – Назва рисунку

Завдання 2.

Вирішення задачі Matrix35

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

matrix – двомірний масив цілих елементів;

Вихідні дані (ім'я, опис, тип):

col_num – номер стовпця з непарних елементів

Алгоритм вирішення показано на рис. 2

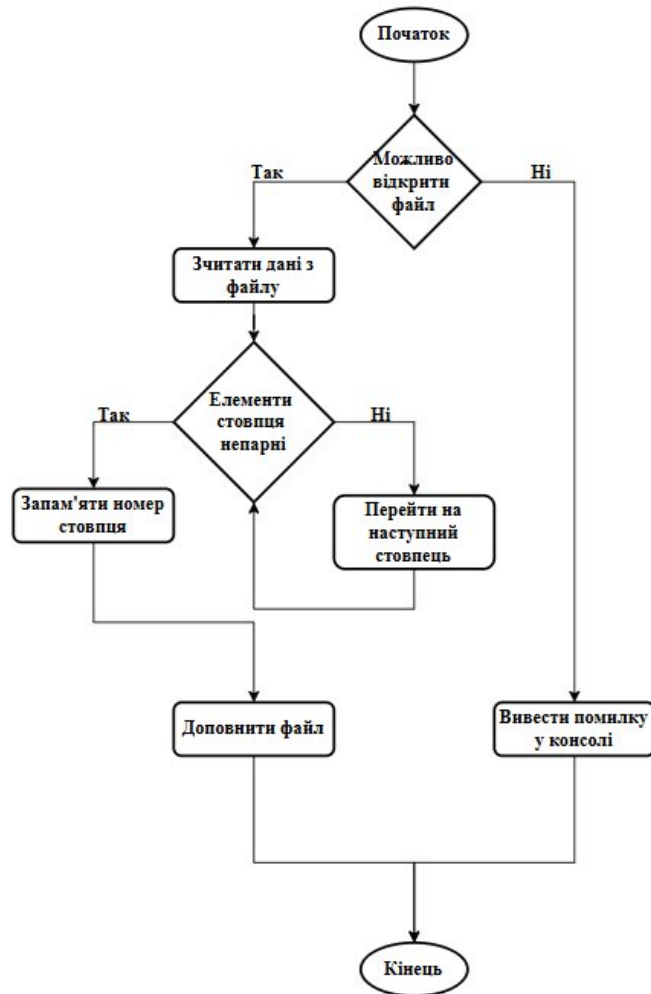


Рисунок 2 – Назва рисунку

Завдання 3.

Розв'язання задачі Sort12

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

arr – масив цілісних;

Вихідні дані (ім'я, опис, тип):

arr – відсортований масив цілих елементів;

Лістинг коду вирішення задачі розділ і номер задач наведено в дод. А (стор. х). Екран роботи програми показаний на рис. Б.х.

ВИСНОВКИ

Вивчено теоретичний матеріал з обробки масивів, бібліотеки для роботи з файлами, методи сортування масивів.

ДОДАТОК А

Лістинг коду програми

```
#include <iostream>
#include "array_utils.h"
#include "matrix_utils.h"
#include "sort_utils.h"

using namespace std;
int main()
{
    int choice;

    do
    {
        cout << "\nSelect task:\n"
              << "1. Array87\n"
              << "2. Matrix35\n"
              << "3. Sort12\n"
              << "0. Exit\n"
              << "Your choice: ";
        cin >> choice;

        switch (choice)
        {
            case 1:
                array_87();
                break;
            case 2:
                matrix_35();
                break;
            case 3:
                sort_12();
                break;
            case 0:
                cout << "Exit the program.\n";
                break;
            default:
                cout << "Wrong choice! Try again.\n";
        }
    } while (choice != 0);

    return 0;
}
```

ДОДАТОК Б

Скріншоти вікна виконання програми

```
Select task:
1. Array87
2. Matrix35
3. Sort12
0. Exit
Your choice: 1
Enter the input file name: array_in_87.txt
Enter the output file name: array_out_87.txt
Original array: 15 10 11 12 13 14
Ordered array: 10 11 12 13 14 15
The result was successfully written to the file array_out_87.txt

Select task:
1. Array87
2. Matrix35
3. Sort12
0. Exit
Your choice: 2
Enter the file name: matrix_in_35.txt
Read matrix:
1 2 3 4 5 6
6 5 7 3 2 1
10 11 17 13 14 15
1 2 3 4 5 6
99 98 97 96 95 94
First column number with only odd numbers: 3
The result has been successfully added to the file matrix_in_35.txt

Select task:
1. Array87
2. Matrix35
3. Sort12
0. Exit
Your choice: 3
Enter the input file name: sort_in_12.txt
Enter the output file name: sort_out_12.txt
Demonstration of insertion sort with binary search:
Initial array: 64 34 25 12 22 11 90
Step 1:
Current element: 34
Position for insertion (found by binary search): 0
Move elements: 64
Array after step 1: 34 64 25 12 22 11 90

Step 2:
Current element: 25
Position for insertion (found by binary search): 0
Move elements: 34 64
Array after step 2: 25 34 64 12 22 11 90

Step 3:
Current element: 12
Position for insertion (found by binary search): 0
Move elements: 25 34 64
Array after step 3: 12 25 34 64 22 11 90

Step 4:
Current element: 22
Position for insertion (found by binary search): 1
Move elements: 25 34 64
Array after step 4: 12 22 25 34 64 11 90

Step 5:
Current element: 11
Position for insertion (found by binary search): 0
Move elements: 12 22 25 34 64
Array after step 5: 11 12 22 25 34 64 90

Step 6:
Current element: 90
Position for insertion (found by binary search): 6
Array after step 6: 11 12 22 25 34 64 90

Sorted array: 11 12 22 25 34 64 90

The result was successfully written to the file sort_out_12.txt

Select task:
1. Array87
2. Matrix35
3. Sort12
0. Exit
Your choice: 0
Exit the program.
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання Array87, Matrix35 і Sort12