

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  
++»

ХАІ.301.електроенергетика,електромеханіка і електротехніка  
319а,№23ЛР

Виконав студент гр. 319а

(підпис, дата)

Дмитро Хара  
(П.І.Б.)

Перевірив

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

## МЕТА РОБОТИ

масивів на мові

реалізувати оголошення,

одновимірних і двовимірних

Вивчити теоретичний матеріал по алгоритмам обробки

C++, а також бібліотеки для роботи з файлами і

введення з файлу, обробку і виведення в файл

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 -23). У

програмі на C++ перетворити масив Дан цілочисельний масив розміру N. Видалити з масиву всі непарні числа і

вивести розмір отриманого масиву і його вміст. , ім'я файлу і необхідні змінні ввести з консолі. Вивести результати у файл «array\_out\_n.txt».

Для вирішення цього завдання на мові C++ потрібно реалізувати наступні кроки:

Зчитати масив з файлу "array\_in\_n.txt".

Пройтися по всім елементам масиву та видалити непарні числа.

Записати отриманий масив у файл "array\_out\_n.txt".

Вивести розмір та вміст масиву у консоль.

Ось приклад коду на C++, який виконує ці кроки:

```
include <iostream>
include <fstream>
include <vector>
```

```
using namespace
```

```
int readArrayFromFile const
```

```
int
```

```

        inputFile
    if
        "Unable to open input file "

    exit 1

    int
    while
        push_back

        close
    return

void writeArrayToFile const
    const    int

        outputFile
    if
        "Unable to open output file "

    exit 1

    for int
        " "

        close

void printArray const    int
    "Size of the resulting array: "

    "Content of the resulting array:

    "

    for int
        " "
```

```

        void removeOddNumbers(int *arr, int n)
        {
            int i = 0;
            for (int j = 0; j < n; j++)
            {
                if (arr[j] % 2 == 0)
                {
                    arr[i] = arr[j];
                    i++;
                }
            }
            n = i;
        }

    int main()
    {
        int n;

        cout << "Enter the input filename: ";
        string filename;
        getline(cin, filename);

        int *arr = new int[n];

        readArrayFromFile(filename, arr, n);

        removeOddNumbers(arr, n);

        cout << "Enter the output filename: ";
        string outputFilename;
        getline(cin, outputFilename);

        writeArrayToFile(outputFilename, arr, n);

        printArray(arr, n);

        return 0;
    }

```

Цей код зчитує масив з файлу, видаляє непарні числа, записує отриманий масив у файл "array\_out\_n.txt" та виводить розмір та вміст масиву у консоль.

у

програмі на C++ Завдання 2. За допомогою текстового редактору створити текстовий файл

«matr\_in\_n.txt» з елементами вихідного двовимірного масиву (n -23). У програмі обробити матрицю Matrix43. Дана матриця розміру  $M \times N$ . Знайти кількість її стовпців, елементи яких

впорядковані за спаданням. , ім'я файлу і необхідні змінні ввести з консолі.

Дописати результати в той же файл.

Для реалізації цього завдання у програмі на C++ потрібно зчитати матрицю з файлу "matr\_in\_n.txt", обробити її для знаходження кількості стовпців, елементи яких впорядковані за спаданням, а потім записати результат у той же файл.

Ось заготовка коду, яка реалізує це завдання:

```
include <iostream>
```

```
include <fstream>
```

```
include <vector>
```

```
using namespace
```

```
int readMatrixFromFile const int int
```

```
inputFile
```

```

if
    "Unable to open input file "

exit 1

    int    matrix    vector int

for int    0

for int    0


    close

return

```

```

int countColumnsSortedDescending(const int

    int    0

    int    size

    int    0 size

    for int    0

    bool    true

```

```
for int 1
```

```
if 1
```

```
    false
```

```
break
```

```
if
```

```
return
```

```
int main
```

```
int
```

```
    "Enter the input filename: "
```

```
int readMatrixFromFile

countColumnsSortedDescending

"Number of columns sorted in descending order: "

outputFile

if
    "Unable to open output file "

exit 1

"Number of columns sorted in descending order: "

close

return 0
```



У цьому коді ми зчитуємо матрицю з файлу, обробляємо її для знаходження кількості стовпців, елементи яких впорядковані за спаданням, і виводимо результат у консоль. Потім ми дописуємо цей результат у той самий файл, щоб зберегти його.

у

програмі на C++ Завдання 3. Вивчити метод сортування , проаналізувати його складність і продемонструвати на прикладі з 7-ми елементів

Метод сортування Порядок Тип елементів

Вибір Зростання Дійсний . Реалізувати у вигляді окремої функції

алгоритм сортування елементів масиву. Також окремими функціями

реалізувати зчитування масиву з текстового файлу і виведення відсортованого

масиву в консоль.

Для реалізації цього завдання у програмі на C++ потрібно спочатку реалізувати алгоритм сортування методом вибору (Selection Sort), потім написати функції для зчитування масиву з файлу та виведення відсортованого масиву в консоль.

Метод сортування вибором полягає у виборі найменшого (або найбільшого) елемента з несортованої частини масиву і обміну його з першим елементом несортованої частини. Цей процес повторюється до повного сортування масиву.

Складність алгоритму сортування вибором у середньому та в найгіршому випадку становить  $O(n^2)$ , де  $n$  - кількість елементів у масиві.

Ось приклад коду для реалізації цього завдання:

```
include <iostream>

include <fstream>

include <vector>

using namespace

    int  readArrayFromFile const

    int

    inputFile

if

    "Unable to open input file "

exit 1

int

while

    push_back
```

```
        close
```

```
    return
```

```
void printArray(const int
```

```
    for (int
```

```
        " "
```

```
void selectionSort(int
```

```
    int size
```

```
    for (int i = 0; i < size - 1; i++)
```

```
        int
```

```
        for (int j = i + 1; j < size; j++)
```

```
            if
```

```
if
```

```
swap
```

```
int main
```

```
"Enter the input filename: "
```

```
int readArrayFromFile
```

```
"Original array: "
```

```
printArray
```

```
selectionSort
```

```
"Sorted array: "  
  
printArray  
  
return 0
```

У цьому коді ми реалізували алгоритм сортування методом вибору, а також функції для зчитування масиву з файлу та виведення відсортованого масиву у консоль.

Завдання 4. Для багаторазового виконання будь-якого з трьох зазначених

вище завдань на вибір розробити алгоритм організації меню в командному

вікні. Введення, виведення, обробку масивів реалізувати окремими функціями з

параметрами.

Для багаторазового виконання будь-якого з трьох зазначених вище завдань на вибір у командному вікні потрібно розробити меню з варіантами вибору кожного завдання, які будуть виконуватися за допомогою відповідних функцій.

Ось приклад організації такого меню:

```
include <iostream>
```

```
using namespace
```

```
void task1
```

```
    "Executing task 1..."
```

```
void task2
```

```
    "Executing task 2..."
```

```
void task3
```

```
    "Executing task 3..."
```

```
int main
```

```
int
```

```
do
```

```
    "===== Menu ====="
```

```
    "1. Task 1"
```

```
    "2. Task 2"
```

```
    "3. Task 3"
```

```
    "4. Exit"
```

```
    "===== "
```

```
    "Enter your choice: "
```

```
switch
```

```
case 1
```

```
    task1
```

```
    break
```

```
case 2
```

```
    task2
```

```
    break
```

```
case 3
```

```
    task3
```

```
    break
```

```
case 4
```

```
        "Exiting the program."

break

default

        "Invalid choice. Please enter a valid option."

while 4

return 0
```

У цьому коді використовується цикл `do-while` для виведення меню та вибору користувачем одного з трьох завдань або виходу з програми. Кожне завдання реалізоване у власній функції, яку викликає відповідна опція меню.

## ВИСНОВКИ

Під час виконання завдань у програмі на C++ було вивчено та відпрацьовано різноманітні концепції програмування, такі як зчитування та запис до файлів, обробка масивів та матриць, реалізація функцій та алгоритмів сортування. Завдяки практичній роботі з кодом програми отримано навички роботи з командним вікном, використання різних структур даних та алгоритмів для вирішення завдань. Хоча виникали труднощі з початку, вони були успішно подолані завдяки пошуковій роботі та взаємодії з різними ресурсами для програмістів.