

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 С
++»

ХАІ.301.електроенергетика,електромеханіка і електротехніка
319а,№17ЛР

Виконав студент гр. 319а

Владислав Мудрік
(підпис, дата) (П.І.Б.)

Перевірів

_____ 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 -17). У

програмі на C++ перетворити масив відповідно Дан цілочисельний масив розміру N. Видалити з масиву всі елементи, що

зустрічаються менше трьох разів, 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, спочатку потрібно зчитати масив з файлу "array_in_n.txt", виконати операцію видалення елементів, які зустрічаються менше трьох разів, а потім записати результуючий масив у файл "array_out_n.txt".

Ось код на мові C++, який вирішує цю задачу:

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

```
using namespace
```

```
int readArrayFromFile const
```

```
int
inputFile
```

```

        if
            "Unable to open input file "

        exit 1

    int
    while
        push_back

        close

    return

const      int

        outputFile

    if
        "Unable to open output file "

    exit 1

    for int

        " "

        close

        int removeInfrequentElements const

    int

        int int

    for int

        int

    for int
    if
        3

        push_back

    return

```

```

int main

    "Enter the input filename: "

    int

readArrayFromFile

    int

removeInfrequentElements

    size

    "Size of the resulting array: "

    "Content of the resulting array:
"
    for int
        " "

    "Enter the output filename: "

writeArrayToFile

    "Results written to "

return 0

```

Цей код зчитує масив з файлу, виконує операцію видалення елементів, які зустрічаються менше трьох разів, і записує результат у файл "array_out_n.txt".

У програмі на C++Завдання 2. За допомогою текстового редактору створити текстовий файл «matr_in_n.txt» з елементами вихідного двовимірного масиву (n -17). У програмі обробити матрицю Дана цілочисельна матриця розміру $M \times N$, 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. , ім'я файлу і необхідні змінні ввести з консолі. Для вирішення цього завдання, спочатку потрібно зчитати матрицю з файлу "matr_in_n.txt", виконати обробку матриці та знайти кількість стовпців, які є схожими на останній стовпець. Після цього результат буде виведений у консоль.

Ось код на мові C++, який вирішує це завдання:

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

using namespace

int
int int
inputFile
if
"Unable to open input file "

exit 1
```

```
int matrix;

vector<int>

for (int i = 0; i < n; i++)
    for (int j = 0; j < m; j++)
        cin >> matrix[i][j];

// Close file
close(fd);

return 0;
}

int countSimilarColumns(const vector<vector<int>>& matrix) {
    int n = matrix.size(); // Number of rows
    int m = matrix[0].size(); // Number of columns

    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            cout << matrix[i][j] << " ";

    cout << endl;

    for (int i = 0; i < m; i++)
        cout << matrix[0][i] << " ";

    return 0;
}

int main() {
    string filename;

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

    if (filename.empty())
        return -1;

    ifstream fin(filename.c_str());
    if (!fin.is_open())
        return -1;

    int n, m;
```

```

int
readMatrixFromFile

int
countSimilarColumns

"Number of columns similar to the
last column: "

return 0

```

Цей код зчитує матрицю з файлу, обробляє її та знаходить кількість стовпців, які є схожими на останній стовпець. Результат виводиться у консоль.

у

програмі на C++ Завдання 3. Вивчити метод сортування , 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.

Метод сортування Порядок Тип елементів

Двійкові вставки Зростання Дійсний

Двійкове вставне сортування (Binary Insertion Sort) є варіацією звичайного сортування вставками, яка використовує бінарний пошук для знаходження правильної позиції для вставки на кожному кроці.

Складність цього методу в середньому та в найгіршому випадках є $O(n^2)$, але в кращому випадку - $O(n \log n)$.

Ось код на мові C++, який демонструє реалізацію двійкового вставного сортування та функції зчитування масиву з файлу і виведення відсортованого масиву у консоль:

```
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 binaryInsertionSort(int
```

```
    int size
```

```
    for (int i = 1
```

```
        int
```

```
        int j = 0; while (j < i - 1
```

```
            while
```

```
                int k = j; while (k > 0 && arr[k] > arr[k - 1]
```

```
                    if
```

```
        1  
    else
```

```
        1
```

```
    for int 1  
        1
```

```
int main
```

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

```
int readArrayFromFile
```

```
    "Original array: "
```

```
    printArray
```

```
    binaryInsertionSort
```

```
    "Sorted array: "
```

```
    printArray
```

```
    return 0
```

У цьому коді реалізовано алгоритм двійкового вставного сортування для сортування масиву. Функції зчитування масиву з файлу і виведення відсортованого масиву у консоль також включені.

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 реалізації меню в командному вікні для вибору одного з трьох завдань, а також виклику відповідних функцій зчитування, обробки та виведення масивів, можна скористатися циклом `do-while`. Ось приклад організації такого меню:

```
include <iostream>
```

```
using namespace
```

```
void processTask1
```

```
"Task 1 is processed"
```

```
void processTask2
```

```
"Task 2 is processed"
```

```
void processTask3
```

```
"Task 3 is processed"
```

```
int main
```

```
int
```

```
do
```

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

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

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

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

```
"4. Exit"
```

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

```
switch
```

```
case 1
```

```
processTask1
```

```
break
```

```
case 2
```

```
processTask2
```

```
break
```

```
case 3
```

```
processTask3

break

case 4

    "Exiting the program."

break

default

    "Invalid choice. Please enter a valid option."

while 4

return 0
```

У цьому коді цикл `do-while` використовується для відображення меню в командному вікні. Користувач може обрати одне з трьох завдань або вийти з програми. Після вибору опції викликається відповідна функція обробки завдання.

ВИСНОВКИ

Під час виконання завдань у програмі на C++ було вивчено та відпрацьовано методи зчитування, обробки та виведення масивів, а також реалізовано алгоритми сортування та обробки матриць. Організація меню в командному вікні дозволила зручно вибирати потрібне завдання. Отримано навички роботи з файлами, роботи з функціями та використання різних алгоритмів для розв'язання задач.

