

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання лабораторних та практичних робіт блоку № 5

На тему: «Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли.

Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.»

з дисципліни: «Основи програмування»

до:

ВНС Лабораторної Роботи № 6

ВНС Лабораторної Роботи № 8

ВНС Лабораторної Роботи № 9

Алготестер Лабораторної Роботи №4

Алготестер Лабораторної Роботи №6

Практичних Робіт до блоку №5

Виконав:

Студент групи ШІ-11

Федоришин Микола Володимирович

Львів 2024

Тема роботи:

Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли.
Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.

Мета роботи:

Опанувати вивчений матеріал на практиці, а саме попрацювати з файлами як у мові C та C++, вдосконалити роботу зі стрічками в C++.

Теоретичні відомості:

- Тема №1: Вступ до Роботи з Файлами.
- Тема №2: Символи і Рядкові Змінні.
- Тема №3: Текстові Файли.
- Тема №4: Бінарні Файли.
- Тема №5: Стандартна бібліотека та робота з файлами.
- Тема №6: Створення й використання бібліотек.

1) Індивідуальний план опрацювання теорії:

- Тема №1: Вступ до роботи з файлами:
 - Джерела інформації:
 - Статті.
<https://www.youtube.com/watch?v=FeNqHytI0fA>
 - Що опрацьовано:
 - Основні операції з файлами: відкриття, читання, запис, закриття
 - Робота з файловими дескрипторами
 - C-style читання з файлу та запис до файлу
 - Перевірка стану файлу: перевірка помилок, кінець файлу
 - Базові приклади читання та запису в файл
- Запланований час на вивчення 2 години.
Витрачений час 2 години.
- Тема №2: Символи і рядкові змінні:
 - Джерела інформації:
 - Статті.
<https://www.youtube.com/watch?v=1FkTJYm-T34&t=80s>
 - Що опрацьовано:
 - Робота з char та string: основні операції і методи
 - Стрічкові літерали та екранування символів
 - Конкатенація, порівняння та пошук у рядках
- Запланований час на вивчення 2 години.
Витрачений час 2 години.
- Тема №3: Текстові файли:
 - Джерела інформації:
 - Статті.
<https://www.youtube.com/watch?v=SSNJ7alki-E&t=3834s>
 - Що опрацьовано:
 - Особливості читання та запису текстових файлів
 - Обробка рядків з файлу: getline, ignore, peek
 - Форматування тексту при записі: setw, setfill, setprecision
 - Парсинг текстових файлів: розділення на слова, аналіз структури
 - Обробка помилок при роботі з файлами
- Запланований час на вивчення 2 години.
Витрачений час 2 години.

- Тема №4:Бінарні файли:
 - Джерела інформації:
 - Статті.
https://www.youtube.com/watch?v=_h5eHf65lgs
 - Що опрацьовано
 - Вступ до бінарних файлів: відмінності від текстових, приклади (великі дані, ігрові ресурси, зображення)
 - Читання та запис бінарних даних
 - Робота з позиціонуванням у файлі: seekg, seekp
 - Серіалізація об'єктів у бінарний формат
- Запланований час на вивчення 2 години.
Витрачений час 2 години.
- Тема № 5:Стандартна бібліотека та робота з файлами:
 - Джерела інформації:
 - Статті.
<https://www.youtube.com/watch?v=SSNJ7alki-E&t=3932s>
 - Що опрацьовано
 - Огляд стандартної бібліотеки для роботи з файлами
 - Потоки вводу/виводу: ifstream, ofstream, fstream
 - Обробка помилок при роботі з файлами
- Запланований час на вивчення 2 години.
Витрачений час 2 години.
- Тема №6:Створення й використання бібліотек:
 - Джерела інформації:
 - Статті.
<https://www.youtube.com/watch?v=mnwDpO4zqLA&t=433s>
 - Що опрацьовано
 - Вступ до створення власних бібліотек у C++
 - Правила розбиття коду на header-и(.h) та source(.cpp) файли
 - Статичні проти динамічних бібліотек: переваги та використання
 - Інтерфейси бібліотек: створення, документування, версіонування
 - Використання сторонніх бібліотек у проектах
- Запланований час на вивчення 2 години.
Витрачений час 2 години.

Також користувався Chat GPT який давав відповіді на конкретні питання по коду.

Виконання роботи:

1. Опрацювання завдання до програм.

Завдання №1

VNS LAB 6 – TASK 1 (VARIANT 14)

Задано рядок, що складається із символів. Символи поєднуються в слова.

Слова одне від одного відокремлюються одним або декількома пробілами.

Наприкінці тексту ставиться крапка. Текст містить не більше 255 символів.

Виконати ввід рядка, використовуючи функцію gets(s) і здійснити обробку рядка у відповідності зі своїм варіантом.

Перетворити рядок таким чином, щоб у ньому залишилися тільки слова, що містять букви й цифри, інші слова знищити.

Завдання №2

VNS LAB 8 – TASK 1 (VARIANT 14)

Сформувати двійковий файл із елементів, заданої у варіанті структури, роздрукувати його вміст, виконати знищення й додавання елементів у відповідності зі своїм варіантом, використовуючи для пошуку елементів що знищуються чи додаються, функцію. Формування, друк, додавання й знищення елементів оформити у вигляді функцій. Передбачити повідомлення про помилки при відкритті файлу й виконанні операцій вводу/виводу.

Структура "Стадіон":

- назва;
- адреса;
- місткість;
- види спорту.

Знищити елемент із заданою назвою, додати 2 елементи після елемента із зазначеним номером.

Завдання №3

VNS LAB 9 – TASK 1 (VARIANT 14)

Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього Інформацію

- 1) Скопіювати з файлу F1 у файл F2 всі рядки, що не містять букву «А» і розташовані між рядками з номерами N1 й N2.
- 2) Визначити номер того рядка, у якому найбільше голосних букв, файлу F2.

Завдання №4

ALGOTESTER LAB 4 (VARIANT 1)

Вам дано 2 цілих чисел масиви, розміром N та M.

Ваше завдання вивести:

1. Різницю N-M
2. Різницю M-N

3. Їх перетин
4. Їх об'єднання
5. Їх симетричну різницю

Вхідні дані

У першому рядку ціле число N - розмір масиву 1

У другому рядку N цілих чисел - елементи масиву 1

У третьому рядку ціле число M - розмір масиву 2

У четвертому рядку M цілих чисел - елементи масиву 2

Вихідні дані

Вивести результат виконання 5 вищезазначених операцій у форматі:

У першому рядку ціле число N - розмір множини

У наступному рядку N цілих чисел - посортована у порядку зростання множина.

Пам'ятайте, ви маєте написати 2 варіанти розв'язку, один з використанням засобів STL (`std::set_intersection`, `std::set_symmetric_difference`, `std::set_difference`, `std::set_union`), інший зі своєю реалізацією. Своє сортування можна не писати.

Завдання №5

ALGOTESTER LAB 4 (VARIANT 1)

Версія коду без `std::set_intersection`, `std::set_symmetric_difference`, `std::set_difference`, `std::set_union`

Завдання №6

ALGOTESTER LAB 6 (VARIANT)

Вам дано N слів та число K.

Ваше завдання перерахувати букви в словах, які зустрічаються в тексті більше-рівне ніж K разів (саме слово, не буква!).

Великі та маленькі букви вважаються однаковими, виводити необхідно малі, посортовані від останньої до першої у алфавіті. Букву потрібно виводити лише один раз.

У випадку якщо таких букв немає - вивести "Empty!".

Вхідні дані

Цілі числа N та K - загальна кількість слів та мінімальна кількість слів щоб враховувати букви цього слова в результаті.

N стрічок s

Вихідні дані

У першому рядку ціле число M - кількість унікальних букв

У другому рядку унікальні букви через пробіли

Завдання №7

CLASS PRACTICE WORK

Задача №1 – Запис текстової стрічки у файл із заданим ім'ям

Реалізувати функцію створення файлу і запису в нього даних:

```
enum FileOpResult { Success, Failure, ... };  
FileOpResult write_to_file(char *name, char *content);
```

Умови задачі:

- створити файл із заданим ім'ям; якщо файл існує – перезаписати його вміст
- написати код стійкий до різних варіантів вхідних параметрів
- name – ім'я, може не включати шлях
- записати у файл вміст стрічки content, прочитати content із стандартного вводу
- повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, запису даних, чи закриття файла.

Задача №2 – Копіювання вмісту файлу у інший файл

Реалізувати функцію створення файлу і запису в нього даних:

```
enum FileOpResult { Success, Failure, ... };  
FileOpResult copy_file(char *file_from, char *file_to);
```

Умови задачі:

- копіювати вміст файлу з ім'ям file_from у файл з ім'ям file_to; написати код стійкий до різних варіантів вхідних параметрів, обробити всі можливі варіанти відсутності одного з файлів
- file_from, file_to – можуть бути повним або відносним шляхом
- повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, читання чи запису даних, закриття файла.

Завдання №8

SELF PRACTICE WORK ALGOTESTER

Щасливий результат

Обмеження: 2 сек., 256 MiB

Після завершення основного туру олімпіади з програмування Зеник отримав невеличкий клаптик паперу, на якому було надруковане число x — кількість балів, що набрав Зеник. Зауважте, що згідно з кращими традиціями олімпіади з програмування, кількість балів Зеника не може бути нульовою чи від'ємною. Помітивши не дуже щасливе обличчя Зеника, Марічка нагадала йому про щасливі цифри. Як ви вже напевно знаєте, щасливими вважають цифри 4 та 7. Марічка запевнила Зеника, що найкращим є не найбільший результат, а той, десятковий запис якого містить найбільше щасливих цифр.

Вам необхідно допомогти юному учаснику олімпіади з програмування та порахувати кількість щасливих цифр у його результаті.

Вхідні дані

У єдиному рядку задано одне ціле число x — результат Зеника.

Вихідні дані

У єдиному рядку виведіть одне ціле число — кількість щасливих цифр у десятковому записі x .

2. Вимоги та планувальна оцінка часу виконання завдань:

Програма №1

- Плановий час на реалізацію 1.5 години.

Програма №2

- Плановий час на реалізацію 3 години.

Програма №3

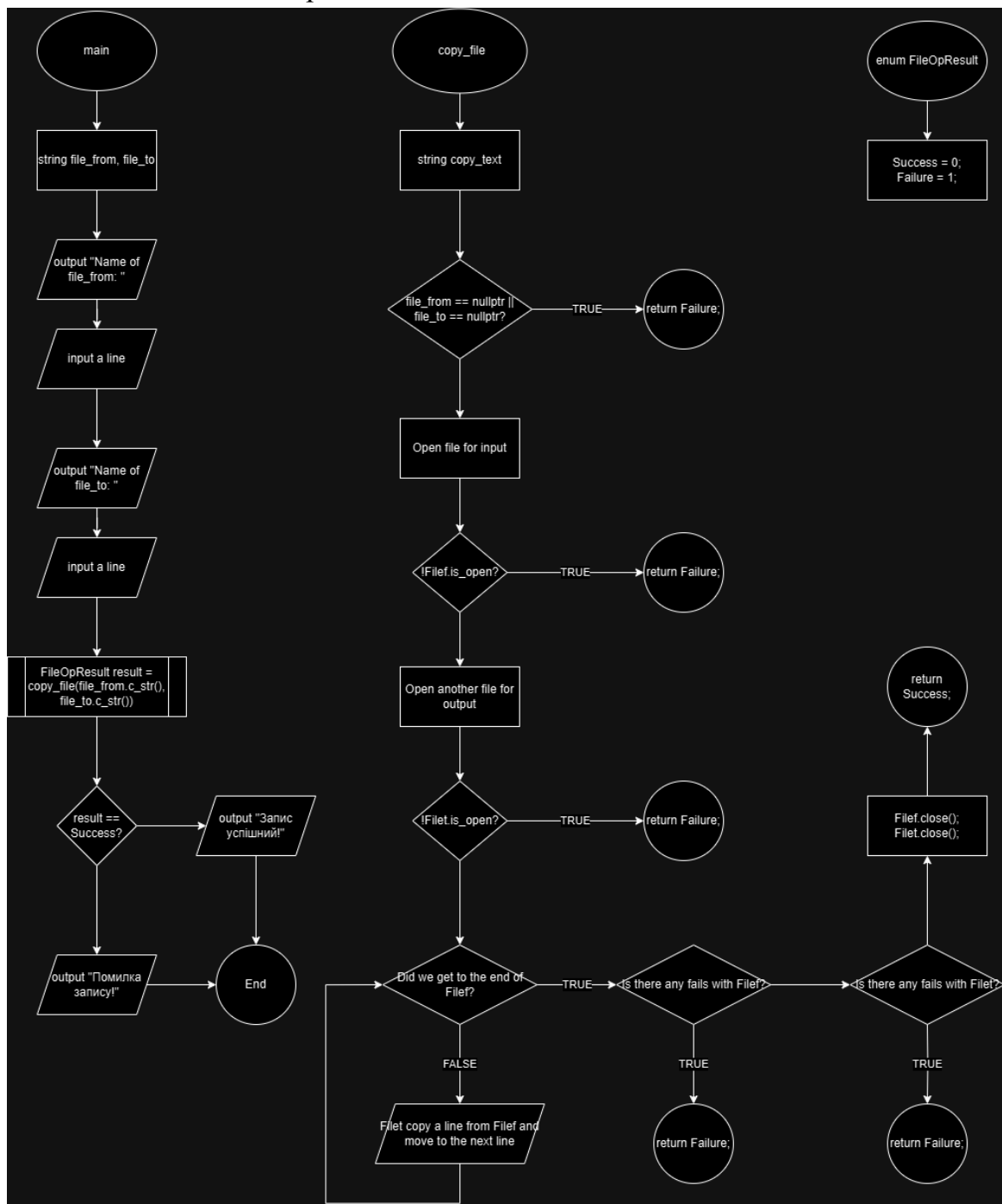
- Плановий час на реалізацію 1.5 години.

Програма №4

- Плановий час на реалізацію 2 години.

Програма №5

- Блок – схема.
- Плановий час на реалізацію 4 години.



Програма №6

- Плановий час на реалізацію 5 години.

Програма №7

- Плановий час на реалізацію 4 години.

Програма №8

- Плановий час на реалізацію 30 хвилин.

3. Код програм з посиланням на зовнішні ресурси та фактично затрачений час:

Завдання №1

```
lab_6 > C:\vns_lab_6_task_2_variant_14_mykola_fedoryshyn.cpp > main()
1  #include <iostream>
2  #include <sstream>
3  #include <cctype>
4  #include <vector>
5  using namespace std;
6  bool Valid(const string &word)
7  {
8      bool has_letter = false;
9      bool has_digit = false;
10     for (char c : word)
11     {
12         if (isalpha(c))
13             has_letter = true;
14         if (isdigit(c))
15             has_digit = true;
16         if (has_letter && has_digit)
17             return true;
18     }
19
20     return false;
21 }
22 int main()
23 {
24     string input;
25     cout << "Введіть рядок: ";
26     getline(cin, input);
27
28     istringstream stream(input);
29     string word;
30     vector<string> valid_words;
31
32     while (stream >> word)
33     {
34         if (Valid(word))
35         {
36             valid_words.push_back(word);
37         }
38     }
39
40     if (valid_words.empty())
41     {
42         cout << "Жодного слова не знайдено!" << endl;
43     }
44     else
45     {
46         for (const string &w : valid_words)
47         {
48             cout << w << " ";
49         }
50         cout << endl;
51     }
52
53     return 0;
54 }
55
```

Рисунок 3.1. Код до програми № 1

```
Введіть рядок: the written w4ords in a3 book, magazine, etc.2, not the pictures:1  
w4ords a3 etc.2, pictures:1  
PS C:\Github\ai_programming_playground_2024\ai_11\mykola_fedoryshyn\epic_5>
```

Рисунок 3.2. Приклад виконання програми № 1

Фактично затрачений час 2.5 години.

- **Посилання на файл у пулл реквесті**

Завдання №2

```

lab8 > G:\vns_lab_8_task_2_variant_1_john_black.cpp > ...
1  #include <iostream>
2  #include <fstream>
3  #include <cstdio>
4  #include <cstring>
5
6  using namespace std;
7
8  struct Stadium {
9      char name[40];
10     char address[40];
11     int capacity;
12     char sports[40];
13 };
14
15 const int N = 2;
16
17 void create_file(const char* filename) {
18     FILE* file;
19
20     file = fopen(filename, "wb");
21     if (file == NULL) {
22         cerr << "ERROR while opening file for writing." << endl;
23         exit(1);
24     }
25
26     Stadium stadium;
27
28     for (int i = 0; i < N; i++) {
29         cout << "Name: "; cin >> stadium.name;
30         cout << "Address: "; cin >> stadium.address;
31         cout << "Capacity: "; cin >> stadium.capacity;
32         cout << "Sports: "; cin >> stadium.sports;
33
34         fwrite(&stadium, sizeof(Stadium), 1, file);
35         if (ferror(file)) {
36             cerr << "ERROR while writing in file." << endl;
37             exit(2);
38         }
39     }
40     fclose(file);
41 }
42
43 void print_file(const char* filename) {
44     FILE* file;
45     file = fopen(filename, "rb");
46     if (file == NULL) {
47         cerr << "ERROR while opening file for reading." << endl;
48         exit(3);
49     }
50
51     Stadium stadium;
52     cout << "File contains:" << endl;
53
54     while (fread(&stadium, sizeof(Stadium), 1, file) == 1) {
55         cout << stadium.name << ", " << stadium.address << ", "
56         << stadium.capacity << " " << stadium.sports << endl;

```

```

54     while (fread(&stadium, sizeof(Stadium), 1, file) == 1) {
55         cout << stadium.name << ", " << stadium.address << ", "
56         << stadium.capacity << ", " << stadium.sports << endl;
57     }
58
59     fclose(file);
60 }
61
62 void add_stadium(const char* filename, Stadium newStadium) {
63     FILE* file;
64     file = fopen(filename, "ab");
65     if (file == NULL) {
66         cerr << "ERROR while opening file for writing." << endl;
67         exit(4);
68     }
69
70     fwrite(&newStadium, sizeof(Stadium), 1, file);
71     if (ferror(file)) {
72         cerr << "ERROR writing while adding stadium." << endl;
73         exit(5);
74     }
75
76     fclose(file);
77 }
78
79
80 void add_stadium_after(const char* filename, const char* stadium_name, Stadium newStadium) {
81     FILE* file;
82     file = fopen(filename, "rb");
83     if (file == NULL) {
84         cerr << "ERROR while opening file for reading." << endl;
85         exit(4);
86     }
87
88     FILE* temp_file;
89     temp_file = fopen("temp.dat", "wb");
90     if (temp_file == NULL) {
91         cerr << "ERROR while creating temporary file" << endl;
92         fclose(file);
93         exit(5);
94     }
95
96     Stadium stadium;
97     bool found = false;
98
99     while (fread(&stadium, sizeof(Stadium), 1, file) == 1) {
100         fwrite(&stadium, sizeof(Stadium), 1, temp_file);
101         if (strcmp(stadium.name, stadium_name) == 0 && !found) {
102             fwrite(&newStadium, sizeof(Stadium), 1, temp_file);
103             found = true;
104         }
105     }

```

```

105     }
106
107     fclose(file);
108     fclose(temp_file);
109
110     if (found) {
111         remove(filename);
112         rename("temp.dat", filename);
113         cout << "Stadium added after " << stadium_name << endl;
114     } else {
115         cout << "Stadium with name " << stadium_name << " isn't found." << endl;
116         remove("temp.dat");
117     }
118 }
119
120 void delete_stadium_by_name(const char* filename, const char* stadium_name) {
121     FILE* file;
122     file = fopen(filename, "rb");
123     if (file == NULL) {
124         cerr << "ERROR while opening file for deleting." << endl;
125         exit(6);
126     }
127
128     FILE* temp_file;
129     temp_file = fopen("temp.dat", "wb");
130     if (temp_file == NULL) {
131         cerr << "ERROR while creating temporary file" << endl;
132         fclose(file);
133         exit(7);
134     }
135
136     Stadium stadium;
137     bool deleted = false;
138     while (fread(&stadium, sizeof(Stadium), 1, file)) {
139         if (strcmp(stadium.name, stadium_name) == 0) {
140             deleted = true;
141         } else {
142             fwrite(&stadium, sizeof(Stadium), 1, temp_file);
143         }
144     }
145
146     fclose(file);
147     fclose(temp_file);
148
149     if (deleted) {
150         remove(filename);
151         rename("temp.dat", filename);
152         cout << "Stadium " << stadium_name << " removed from file." << endl;
153     } else {
154         cout << "Stadium with name " << stadium_name << " isn't found." << endl;
155         remove("temp.dat");
156     }
157 }
158

```

```

158
159 int main() {
160     const char* filename = "stadiums.dat";
161     string name_to_delete;
162     cout << "Стадіон який будемо видаляти: ";
163     getline(cin, name_to_delete);
164
165     cout << "\nСтворення файлу\n";
166     create_file(filename);
167
168     cout << "\nДрук файлу\n";
169     print_file(filename);
170
171     cout << "\nДодавання нового стадіону\n";
172     Stadium newStadium = {"Arena Lviv", "Lviv, Ukraine", 35000, "Football"};
173     add_stadium(filename, newStadium);
174
175     cout << "\nДодавання нового стадіону після певного стадіону\n";
176     Stadium anotherStadium = {"Dnipro Arena", "Dnipro, Ukraine", 31000, "Football"};
177     add_stadium_after(filename, "Arena Lviv", anotherStadium);
178
179     cout << "\nВидалення стадіону за назвою\n";
180     delete_stadium_by_name(filename, name_to_delete.c_str());
181
182     cout << "\nПісля змін: \n";
183     print_file(filename);
184
185     return 0;
186 }
187

```

Рисунок 3.3. Код до програми № 2

```

✓ TERMINAL
PS C:\Github\ai_programming_playground_2024\ai_11\mykola_fedoryshyn\epic_5> .\41jca.euw' '--stderr=Microsoft-MIEngine-Error-gx2qum1c.2wn' '--pid=Microsoft-
Стадіон який будемо видаляти: name1

Створення файлу
Name: name1
Address: address1
Capacity: 100
Sports: sports1
Name: name2
Address: address2
Capacity: 200
Sports: sports2

Друк файлу
File contains:
name1, address1, 100, sports1
name2, address2, 200, sports2

Додавання нового стадіону

Додавання нового стадіону після певного стадіону
Stadium added after Arena Lviv

Видалення стадіону за назвою
Stadium name1 removed from file.

Після змін:
File contains:
name2, address2, 200, sports2
Arena Lviv, Lviv, Ukraine, 35000, Football
Dnipro Arena, Dnipro, Ukraine, 31000, Football
PS C:\Github\ai_programming_playground_2024\ai_11\mykola_fedoryshyn\epic_5>

```

Рисунок 3.4. Приклад виконання програми №2

Фактично затрачений час 5 годин.

Посилання на файл у пулл реквесті

Завдання №3

```
lab9 > vns_lab_9_task_2_variant_1_john_black.cpp > ...
1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include <vector>
5  using namespace std;
6
7  int main() {
8      int vowels_count[10] = {0,0,0,0,0,0,0,0,0,0};
9
10     ofstream file1("F1.txt");
11     if (!file1.is_open()) {
12         cerr << "Не вдалося відкрити файл F1 для запису!" << endl;
13         return 1;
14     }
15     vector<string> lines = {
16         "This is the first line.",
17         "The second line with the letter A.",
18         "The third line with the letter A.",
19         "Fourth line.",
20         "The fifth line.",
21         "The sixth line with the letter A.",
22         "The seventh line with the letter A.",
23         "Eighth line.",
24         "Ninth line.",
25         "The tenth line with the letter A."
26     };
27
28     for (const string &line : lines) {
29         file1 << line << endl;
30     }
31     file1.close();
32     int N1, N2;
33     cout << "Введіть номер початкового рядка (N1): ";
34     cin >> N1;
35     cout << "Введіть номер кінцевого рядка (N2): ";
36     cin >> N2;
37
38     if (N1 > N2 || N1 < 1 || N2 > lines.size()) {
39         cerr << "Некоректні значення N1 або N2!" << endl;
40         return 1;
41     }
42
43     ifstream inFile("F1.txt");
44     ofstream outFile("F2.txt");
45     if (!inFile.is_open() || !outFile.is_open()) {
46         cerr << "Не вдалося відкрити файл для читання або запису!" << endl;
```



```

9 > ≡ F2.txt
1   Fourth line.
2   The fifth line.
3   Eighth line.
4   Ninth line.
5

```

Рисунок 3.8. У файлі F2.txt тільки ті, які починаються на літеру 'A'
Фактично затрачений час 3 години.

- Посилання на файл у пулл реквесті

Завдання №4

З використанням STL

```

algotester_lab4 > C:\algotester_lab_4_task_mykola_fedoryshyn.cpp > ...
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  void display(const vector<int>& vec) {
8      int s = vec.size();
9      cout << s << '\n';
10     for (int i = 0; i < s; i++) {
11         cout << vec[i] << ' ';
12     }
13     cout << "\n\n";
14 }
15
16 int main() {
17     vector<int> diff1, diff2, common, combined, symDiff;
18     int size1, size2;
19     cin >> size1;
20     vector<int> list1(size1);
21     for (int &element : list1) {
22         cin >> element;
23     }
24
25     cin >> size2;
26     vector<int> list2(size2);
27     for (int &element : list2) {
28         cin >> element;
29     }
30
31     sort(list1.begin(), list1.end());
32     sort(list2.begin(), list2.end());
33
34
35     set_difference(list1.begin(), list1.end(), list2.begin(), list2.end(), back_inserter(diff1));
36     set_difference(list2.begin(), list2.end(), list1.begin(), list1.end(), back_inserter(diff2));
37     set_intersection(list1.begin(), list1.end(), list2.begin(), list2.end(), back_inserter(common));
38     set_union(list1.begin(), list1.end(), list2.begin(), list2.end(), back_inserter(combined));
39     set_symmetric_difference(list1.begin(), list1.end(), list2.begin(), list2.end(), back_inserter(symDiff));
40
41     display(diff1);
42     display(diff2);
43     display(common);
44     display(combined);
45     display(symDiff);
46
47     return 0;
48 }

```

Рисунок 3.9. Код до програми №3 (з STL)

```
✓ TERMINAL

5
4 5 6 7 8
3
1 2 3

3
6 7 8

2
4 5

8
1 2 3 4 5 6 7 8

6
1 2 3 6 7 8
```


Рисунок 3.10. Приклад виконання програми №3 (з STL)

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 20	Зараховано	0.003	1.258	Перегляд

Рисунок 3.11. Статус задачі на Algotester

Фактично затрачений час 20 хв.

Завдання №5

algotester_lab4 >  algotester_lab_4_variant_1_mykola_fedoryshyn.cpp > ...

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  void display(const vector<int>& vec) {
8      cout << vec.size() << '\n';
9      for (int num : vec) {
10         cout << num << ' ';
11     }
12     cout << "\n\n";
13 }
14
15
16 vector<int> Difference(const vector<int>& first, const vector<int>& second) {
17     vector<int> diff;
18     int i = 0, j = 0;
19     while (i < first.size() && j < second.size()) {
20         if (first[i] < second[j]) {
21             diff.push_back(first[i++]);
22         } else if (first[i] > second[j]) {
23             ++j;
24         } else {
25             ++i;
26             ++j;
27         }
28     }
29     while (i < first.size()) {
30         diff.push_back(first[i++]);
31     }
32     return diff;
33 }
34
35
36 vector<int> Intersection(const vector<int>& first, const vector<int>& second) {
37     vector<int> result;
38     int i = 0, j = 0;
39     while (i < first.size() && j < second.size()) {
40         if (first[i] == second[j]) {
41             result.push_back(first[i]);
42             ++i;
43             ++j;
44         } else if (first[i] < second[j]) {
45             ++i;
46         } else {
47             ++j;
48         }
49     }
50     return result;
51 }
52
```

```

52
53 vector<int> Union(const vector<int>& first, const vector<int>& second) {
54     vector<int> result;
55     int i = 0, j = 0;
56     while (i < first.size() || j < second.size()) {
57         if (j == second.size() || (i < first.size() && first[i] < second[j])) {
58             result.push_back(first[i++]);
59         } else if (i == first.size() || second[j] < first[i]) {
60             result.push_back(second[j++]);
61         } else {
62             result.push_back(first[i]);
63             ++i;
64             ++j;
65         }
66     }
67     return result;
68 }
69
70 vector<int> SymmetricDifference(const vector<int>& first, const vector<int>& second) {
71     vector<int> result;
72     int i = 0, j = 0;
73     while (i < first.size() || j < second.size()) {
74         if (j == second.size() || (i < first.size() && first[i] < second[j])) {
75             result.push_back(first[i++]);
76         } else if (i == first.size() || second[j] < first[i]) {
77             result.push_back(second[j++]);
78         } else {
79             ++i;
80             ++j;
81         }
82     }
83     return result;
84 }
85
86 int main() {
87     int size1, size2;
88     cin >> size1;
89     vector<int> list1(size1);
90     for (int &element : list1) {
91         cin >> element;
92     }
93
94     cin >> size2;
95     vector<int> list2(size2);
96     for (int &element : list2) {
97         cin >> element;
98     }
99
100     sort(list1.begin(), list1.end());
101     sort(list2.begin(), list2.end());
102
103
104     vector<int> diff1 = Difference(list1, list2);
105     vector<int> diff2 = Difference(list2, list1);

```

```

103
104     vector<int> diff1 = Difference(list1, list2);
105     vector<int> diff2 = Difference(list2, list1);
106     vector<int> common = Intersection(list1, list2);
107     vector<int> combined = Union(list1, list2);
108     vector<int> symDiff = SymmetricDifference(list1, list2);
109
110
111     display(diff1);
112     display(diff2);
113     display(common);
114     display(combined);
115     display(symDiff);
116
117     return 0;
118 }

```

Рисунок 3.12. Код до програми №4 (без STL)

✓ TERMINAL

```

5
4 5 6 7 8
3
1 2 3

3
6 7 8

2
4 5

8
1 2 3 4 5 6 7 8

6
1 2 3 6 7 8

```

Рисунок 3.13. Приклад виконання програми №4 (без STL)

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 20	Зараховано	0.003	1.258	Перегляд

Рисунок 3.14. Статус задачі на Algotester

Фактично затрачений час 4 години.

Посилання на файл у пулл реквесті

Завдання №6

```
algotester_lab6 > G: algotester_lab_6_variant_1_mykola_fedoryshyncpp.cpp > ...
1  #include <iostream>
2  #include <vector>
3  #include <unordered_map>
4  #include <algorithm>
5  using namespace std;
6
7  int main() {
8      int n, k;
9      cin >> n >> k;
10     cin.ignore();
11
12     vector<string> words(n);
13
14     for (int i = 0; i < n; ++i) {
15         getline(cin, words[i]);
16         transform(words[i].begin(), words[i].end(), words[i].begin(), ::tolower);
17     }
18     unordered_map<string, int> freq_map;
19     for (const auto& word : words) { /*створюємо map де кожному слові буде відповідати 1*/
20         ++freq_map[word];
21     }
22
23     vector<int> char_freq(26, 0);
24     for (const auto& pair : freq_map) {
25         if (pair.second >= k) {
26             for (char c : pair.first) {
27                 char_freq[c - 'a'] = 1;
28             }
29         }
30     }
31     vector<char> result;
32
33     for (int i = 25; i >= 0; --i) {
34         if (char_freq[i]) {
35             result.push_back(i + 'a');
36         }
37     }
38
39     if (result.empty()) {
40         cout << "Empty!";
41     } else {
42         cout << result.size() << endl;
43         for (char c : result) {
44             cout << c << " ";
45         }
46     }
47
48     return 0;
49 }
```

Рисунок 3.15. Код до програми №5

```
5 2
stugna
neptune
grim
oplot
Grim
4
r m i g
```

Рисунок 3.16. Приклад виконання програми №5

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 20	Зараховано	0.034	1.293	Перегляд

Рисунок 3.17.Статус задачі на Algotester

Фактично затрачений час 5 годин.

Посилання на файл у пулл реквесті

Завдання №7

```
practice > G+ practice_work_task_1_mykola_fedoryshyn.cpp > main()
1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  enum FileOpResult {
6      Success,
7      Failure
8  };
9
10 FileOpResult write_to_file(const char *name, const char *content) {
11     if (name == nullptr || content == nullptr) {
12         return Failure;
13     }
14
15     std::ofstream file(name, std::ios::out | std::ios::trunc); // Відкриваємо файл у режимі перезапису
16     if (!file.is_open()) {
17         return Failure;
18     }
19
20     file << content;
21     if (file.fail()) {
22         return Failure;
23     }
24
25     file.close();
26     return file.fail() ? Failure : Success;
27 }
28
29 int main() {
30     std::string filename;
31     std::string content;
32
33     std::cout << "Введіть ім'я файлу: ";
34     std::getline(std::cin, filename);
35
36     std::cout << "Введіть вміст файлу: ";
37     std::getline(std::cin, content);
38
39     FileOpResult result = write_to_file(filename.c_str(), content.c_str());
40
41     if (result == Success) {
42         std::cout << "Запис успішний!" << std::endl;
43     } else {
44         std::cout << "Помилка запису!" << std::endl;
45     }
46
47     return 0;
48 }
49
```

Рисунок 3.19. Код до програми №7(задача 1)

```
Введіть ім'я файлу: text
Введіть вміст файлу: Hello world!
Запис успішний!
PS C:\GitHub\ai_programming_playgro
```

Рисунок 3.20. Приклад виконання програми №7 (задача 1)

```
practice > ≡ text
1  Hello world|
```

Рисунок 3.21. Запис у файлі (задача 1)


```

ice > G+ practice_work_task_2_mykola_fedoryshyn.cpp > copy_file(const char *, const char *)
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
enum FileOpResult {
    Success,
    Failure
};

FileOpResult copy_file(const char *file_from, const char *file_to);
int main(){
    string file_from, file_to;

    cout << "Name of file_from: ";
    getline(cin, file_from);
    cout << "Name of file_to: ";
    getline(cin, file_to);

    FileOpResult result = copy_file(file_from.c_str(), file_to.c_str());
    if (result == Success) {
        std::cout << "Запис успішний!" << std::endl;
    } else {
        std::cout << "Помилка запису!" << std::endl;
    }
    return 0;
}

```

Рисунок 3.22. Код до програми №7 (задача 2)

```

Name of file_from: Text
Name of file_to: created_file
Запис успішний!
PS C:\GitHub\ai_programming_pl

```

Рисунок 3.23. Приклад виконання програми №7 (задача 2)

Фактично затрачений час 2.5 години.

Посилання на файл у пулл реквесті

Завдання №8

```

1  #include <iostream>
2  #include <string>
3
4  int main() {
5      long long x;
6      std::cin >> x;
7
8      std::string str_x = std::to_string(x);
9      int lucky_count = 0;
10
11     for (char digit : str_x) {
12         if (digit == '4' || digit == '7') {
13             lucky_count++;
14         }
15     }
16     std::cout << lucky_count << std::endl;
17     return 0;
18 }
19

```

Рисунок 3.24. Код до програми №8

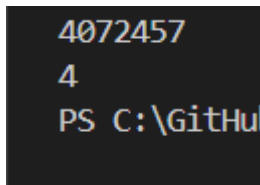


Рисунок 3.25. Приклад виконання програми №8

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
2 години тому	C++ 20	Зараховано	0.003	1.020	Перегляд

Рисунок 3.26. Статус задачі на Algotester

Фактично затрачений час 1 година.

Посилання на файл у пулл реквесті

4. Робота з командою:

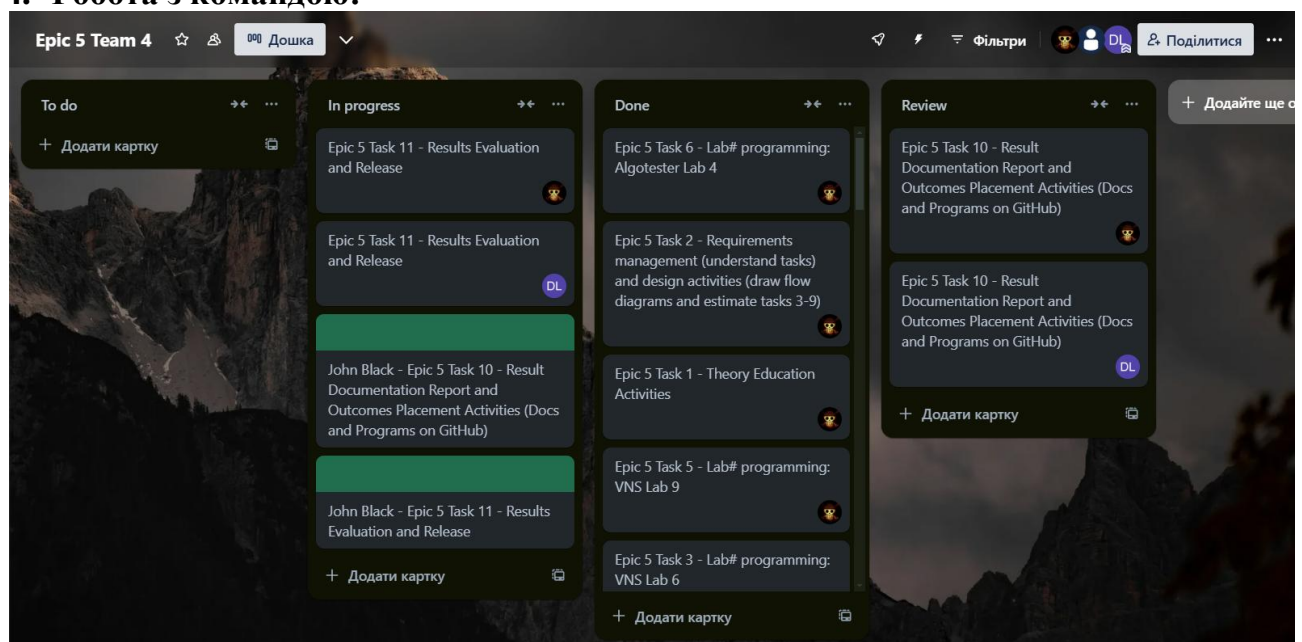


Рисунок 4.1. Командна дошка в Trello

Висновок: У межах практичних та лабораторних робіт блоку №5, я вивчив багато нового матеріалу, такого як: файли, бінарні файли символи і рядкові змінні та текстові файли та деталі/методи роботи з файлами у варіанті мови C та C++. Застосувавши вивчений матеріал на практиці краще зрозумів як це все працює і як це реалізовувати. Також створив блок-схему до найважчої задачі, і краще зрозумів як працює програма. Створив дошку в Trello для роботи з командою.