

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
Фарина Арсеній Петрович

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 2)

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.

Надрукувати всі слова, які не містять голосних букв.

Завдання №2

VNS LAB 8 – TASK 1 (VARIANT 2)

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 "Співробітник":

- прізвище, ім'я, по батькові;

- посада

- рік народження;

- заробітна плата.

Знищити елемент із зазначеним прізвищем, додати елемент після елемента із зазначеним номером.

Завдання №3

VNS LAB 9 – TASK 1 (VARIANT 2)

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

- 1) Скопіювати у файл F2 тільки ті рядки з F1, які починаються з букви «А».
- 2) Підрахувати кількість слів в 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 2)

Вам дано масив a з N цілих чисел.

Спочатку видаліть масиву a усі елементи що повторюються, наприклад масив $[1, 3, 3, 4]$ має перетворитися у $[1, 3, 4]$. Після цього оберніть посортовану версію масиву a на K , тобто при $K = 3$ масив $[1, 2, 3, 4, 5, 6, 7]$ перетвориться на $[4, 5, 6, 7, 1, 2, 3]$.

Виведіть результат.

Вхідні дані

У першому рядку цілі числа N та K

У другому рядку N цілих чисел - елементи масиву a

Вихідні дані

У першому рядку ціле число N - розмір множини a

У наступному рядку N цілих чисел - множина a

Пам'ятайте, ви маєте написати 2 варіанти розв'язку, один з використанням засобів STL (std::unique, std::sort, std::rotate), інший зі своєю реалізацією.

Завдання №6

ALGOTESTER LAB 6 (VARIANT 2)

У вас є шахова дошка розміром 8×8 та дуже багато фігур.

Кожна клітинка може мати таке значення:

- Пуста клітинка O
- Пішак P
- Тура R
- Кінь N
- Слон B
- Король K
- Королева Q

Вам дають позиції фігур на дошці (всі фігури одного кольору, кількість королів може бути > 1). Далі йдуть Q запитів з координатами клітинки $\{x, y\}$. На кожен запит ви маєте вивести стрічку si - посортовані за алфавітом букви фігур, які атакують цю клітинку (пішаки атакують вниз). У випадку, якщо на клітинці стоїть якась фігура - виведіть символ X. У випадку, якщо клітинку не атакують - виведіть O. Наявність фігури у певній клітинці не блокує атаку для іншої фігури. Тобто якщо між турою та клітинкою стоїть інша фігура - вважається що тура атакує цю клітинку.

Вхідні дані

У перших 8 рядках стрічка rowi - стан i-го рядка дошки.

У наступному рядку ціле число Q - кількість записів

У наступних Q рядках 2 цілих числа x та y - координати клітинки

Вихідні дані

Q разів відповідь у наступному форматі:

Строка result - усі фігури, які атакують клітинку з запиту.

Завдання №7

CLASS PRACTICE WORK

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

Реалізувати функцію створення файла 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 файла у інший файл

Реалізувати функцію створення файла і запису в нього даних:

```
enum FileOpResult { Success, Failure, ... };
```

```
FileOpResult copy_file(char *file_from, char *file_to);
```

Умови задачі:

- копіювати вміст файла 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

Малята, напишіть на клаптику паперу ціле додатне число nn. Тепер уявіть собі, що

Ви можете переставляти цифри у його десятковому записі як завгодно. Єдина

умова — не повинно бути нулів на початку запису числа. Вам потрібно визначити

мінімальне та максимальне числа, які можна отримати таким способом.

Бажаю успіху! І не забувайте, що ви все знаєте, просто можливо щось забули.

Вхідні дані

У єдиному рядку задано одне ціле число nn.

Вихідні дані

У єдиному рядку виведіть два цілих числа — мінімальне та максимальне числа, які можна отримати перестановкою цифр числа n.

2. Вимоги та планувальна оцінка часу виконання завдань:

Програма №1

- Важливі деталі для реалізації програми.

- Використовувати функцію `gets()`, а також працювати з варіантом стрічок у мові C. Перевіряти на голосні букви як великі так і малі.
- Плановий час на реалізацію 1.5 години.

Програма №2

- Важливі деталі для реалізації програми.
- Працювати з файлами як у мові C, не забувати після того як відкрив файл його закрити для коректної роботи програми, а також щоразу перевіряти чи не сталася помилка під час відкриття файла для читання чи запису.
- Плановий час на реалізацію 3 години.

Програма №3

- Блок - схема
- Важливі деталі для реалізації програми.
- Працювати з файлами як у мові C++, а саме з директивою `fstream` і її функціями `fstream` та `ofstream`.
- Плановий час на реалізацію 1.5 години.

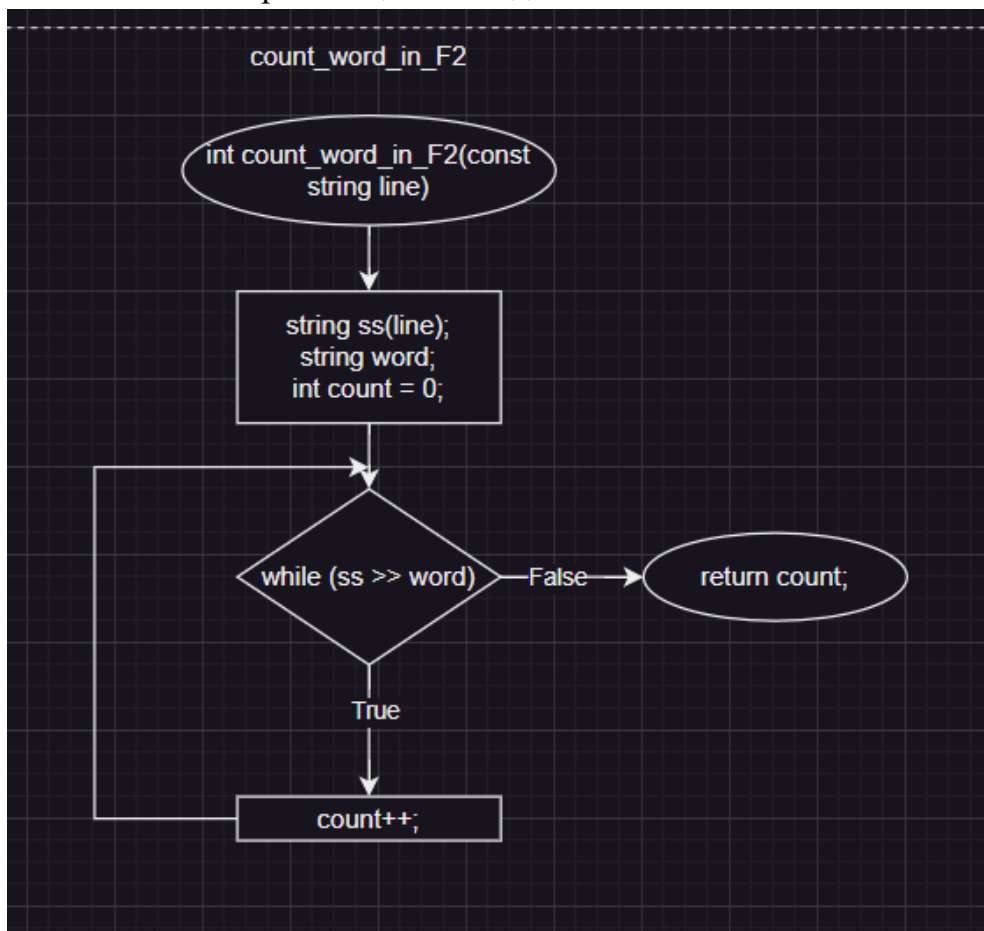


Рисунок 2.1. Функція `count_word_in_F2`

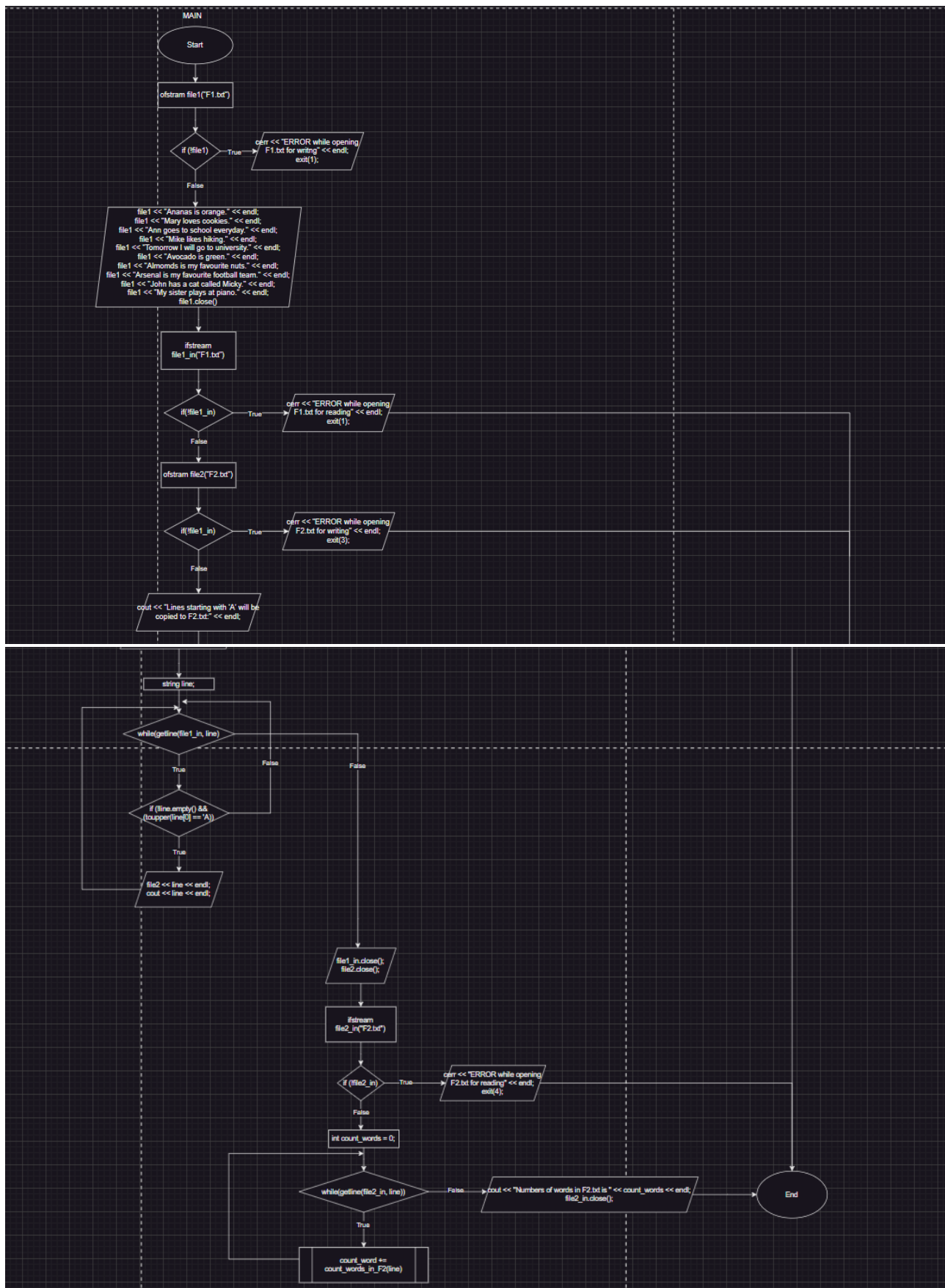


Рисунок 2.2. Функція main

Програма №4

- Важливі деталі для реалізації програми.
- Розібратися з функціями директиви algorithm для сортування, об'єднання, перетину та симетричної різниці двох масивів.
- Плановий час на реалізацію 2 години.

Програма №5

- Блок – схема.
- Важливі деталі для реалізації програми.
- Спробувати написати програму без вбудованих функцій директиви `algorithm`(окрім `sort`), використовувати вектор для зберігання значень, цикли для того щоб проходитися по елементах у векторі.
- Плановий час на реалізацію 4 години.

Програма №6

- Важливі деталі для реалізації програми.
- Написати використовуючи функції директиви `set` та `algorithm`, а саме сортування масиву, видалення однакових елементів та оберт масиву, тобто перенесення елементів відсортованого масиву на K , ті які були в кінці наперед перенести.
- Плановий час на реалізацію 5 години.

Програма №7

- Важливі деталі для реалізації програми(задачі 1 та задачі 2).
- Для задачі 1 створити функцію `FileOpResult` яка буде приймати назву файлу та його вміст і будемо записувати цей вміст у файл. Важливо перевіряти чи добре йде запис файлу і якщо помилка виводити на екран відповідний помилку. Також потрібно не забувати відкривати і закривати файл, щоб не сталася помилка
- Плановий час на реалізацію 4 години.

Програма №8

- Важливі деталі для реалізації програми.
- Використовувати для реалізації стрічку, потім відсортувати у порядку спадання та зростання цифри і з'єднати їх у число. Якщо на початку буде нуль то поміняти його з наступним елементом.
- Плановий час на реалізацію 30 хвилин.

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

Завдання №1

```
1  #include <iostream>
2  #include <cstring>
3
4  using namespace std;
5
6  // Функція для перевірки на наявність голосних
7  int check_for_vowels(char word[]) {
8      char vowels[] = "AEIOUaeiou";
9      for (int i = 0; word[i] != '\0'; i++) {
10         for (int j = 0; vowels[j] != '\0'; j++) {
11             if (word[i] == vowels[j]) {
12                 return 1; // Повертаємо 1, якщо є голосна
13             }
14         }
15     }
16     return 0; // Повертаємо 0, якщо голосних немає
17 }
18
19 int main() {
20     char str[256];
21     char word[256];
22
23     cout << "Enter a string: ";
24     fgets(str, 256, stdin);
25
26     int index = 0;
27
28     // Поки не дійшли до кінця рядка або крапки
29     while (str[index] != '.' && str[index] != '\0') {
30         int word_len = 0;
31
32         // Пропускаємо пробіли
33         while (str[index] == ' ') {
34             index++;
35         }
36
37         // Збираємо символи в слово
38         while (str[index] != '.' && str[index] != ' ' && str[index] != '\0') {
39             word[word_len++] = str[index++];
40         }
41
42         word[word_len] = '\0'; // Завершуємо слово
43
44         // Перевіряємо, чи немає голосних, і виводимо слово
45         if (word_len > 0 && !check_for_vowels(word)) {
46             cout << word << endl;
47         }
48     }
49
50     return 0;
51 }
52
```

Рисунок 3.1. Код до програми № 1

```
Enter a string: Hello my name is Arsenii
my
```

Рисунок 3.2. Приклад виконання програми № 1

Фактично затрачений час 2.5 години.

- Посилання на файл у пулл реквесті
https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/355
Завдання №2

```
1  #include <iostream>
2  #include <fstream>
3  #include <stdio>
4  #include <cstring>
5
6  using namespace std;
7
8  struct EMPLOYEE {
9      char name[40];
10     char surname[40];
11     char last_name[40];
12     char position[40];
13     int birth_year;
14     float salary;
15 };
16
17 const int N = 2; // Кількість співробітників для створення файлу
18
19 // Створення файлу з співробітниками
20 void create_file(const char* filename) {
21     FILE* file;
22
23     file = fopen(filename, "wb");
24     if (file == NULL) {
25         cerr << "ERROR while opening file for writing." << endl;
26         exit(1);
27     }
28
29     EMPLOYEE emp;
30
31     for (int i = 0; i < N; i++) {
32         cout << "Name: "; cin >> emp.name;
33         cout << "Surname: "; cin >> emp.surname;
34         cout << "Last Name: "; cin >> emp.last_name;
35         cout << "Position: "; cin >> emp.position;
36         cout << "Birth Year: "; cin >> emp.birth_year;
37         cout << "Salary: "; cin >> emp.salary;
38
39         fwrite(&emp, sizeof(EMPLOYEE), 1, file);
40         if (ferror(file)) {
41             cerr << "ERROR while writing in file." << endl;
42             exit(2);
43         }
44     }
45     fclose(file);
46 }
47
48 // Друк файлу
49 void print_file(const char* filename) {
50     FILE* file;
51     file = fopen(filename, "rb");
52     if (file == NULL) {
53         cerr << "ERROR while opening file for reading." << endl;
54         exit(3);
55     }
56
57     EMPLOYEE emp;
58     cout << "File contains:" << endl;
59
60     while (fread(&emp, sizeof(EMPLOYEE), 1, file) == 1) {
61         cout << emp.name << " " << emp.surname << " " << emp.last_name << " "
62             << emp.position << " " << emp.birth_year << " " << emp.salary << endl;
63     }
64
65     fclose(file);
66 }
```

```

68 // Додавання нового співробітника в кінець файлу
69 void add_employee(const char* filename, EMPLOYEE newEmp) {
70     FILE* file;
71     file = fopen(filename, "ab");
72     if (file == NULL) {
73         cerr << "ERROR while opening file for writing." << endl;
74         exit(4);
75     }
76
77     fwrite(&newEmp, sizeof(EMPLOYEE), 1, file);
78     if (ferror(file)) {
79         cerr << "ERROR writing while adding employee." << endl;
80         exit(5);
81     }
82
83     fclose(file);
84 }
85
86 // Додавання співробітника після співробітника з певним прізвищем
87 void add_employee_after(const char* filename, const char* surname) {
88     FILE* file;
89     file = fopen(filename, "rb");
90     if (file == NULL) {
91         cerr << "ERROR while opening file for reading." << endl;
92         exit(4);
93     }
94
95     FILE* temp_file;
96     temp_file = fopen("temp.dat", "wb");
97     if (temp_file == NULL) {
98         cerr << "ERROR while creating temporary file" << endl;
99         fclose(file);
100        exit(5);
101    }
102
103    EMPLOYEE emp;
104    bool found = false;
105
106    while (fread(&emp, sizeof(EMPLOYEE), 1, file) == 1) {
107        fwrite(&emp, sizeof(EMPLOYEE), 1, temp_file); // записуємо поточний запис
108        if (strcmp(emp.surname, surname) == 0 && !found) {
109            // Додаємо нового співробітника після знайденого
110            fwrite(&emp, sizeof(EMPLOYEE), 1, temp_file);
111            found = true;
112        }
113    }
114
115    fclose(file);
116    fclose(temp_file);
117
118    if (found) {
119        remove(filename);
120        rename("temp.dat", filename);
121        cout << "Employee added after " << surname << endl;
122    } else {
123        cout << "Employee with surname " << surname << " isn't found." << endl;
124        remove("temp.dat");
125    }
126 }
127
128 // Видалення співробітника за прізвищем
129 void delete_employee_with_surname(const char* filename, const char* surname) {
130     FILE* file;
131     file = fopen(filename, "rb");
132     if (file == NULL) {
133         cerr << "ERROR while opening file for deleting." << endl;
134         exit(6);
135     }
136
137     FILE* temp_file;
138     temp_file = fopen("temp.dat", "wb");
139     if (temp_file == NULL) {
140         cerr << "ERROR while creating temporary file" << endl;
141         fclose(file);
142         exit(7);
143     }
144
145     EMPLOYEE emp;
146     bool deleted_surname = false;
147     while (fread(&emp, sizeof(EMPLOYEE), 1, file)) {
148         if (strcmp(emp.surname, surname) == 0) {
149             deleted_surname = true;
150         } else {
151             fwrite(&emp, sizeof(EMPLOYEE), 1, temp_file);
152         }
153     }
154
155     fclose(file);
156     fclose(temp_file);
157
158     if (deleted_surname) {
159         remove(filename);
160         rename("temp.dat", filename);
161         cout << "Employee " << surname << " removed from file." << endl;
162     } else {
163         cout << "Employee with surname " << surname << " isn't found." << endl;
164         remove("temp.dat");
165     }
166 }
167 }
168

```

```

168
169 int main() {
170     const char* filename = "employees.dat";
171
172     // Створення файлу
173     create_file(filename);
174
175     // Друк вмісту файлу
176     print_file(filename);
177
178     // Додавання нового співробітника
179     EMPLOYEE newEmp = {"Danulo", "Siatetski", "Romanovych", "Student", 2007, 500};
180     add_employee(filename, newEmp);
181
182     // Видалення співробітника за прізвищем
183     delete_employee_with_surname(filename, "Faryna");
184
185     // Друк вмісту файлу після змін
186     cout << "\nAfter deleting employee surname: \n";
187     print_file(filename);
188
189     return 0;
190 }
191

```

Рисунок 3.3. Код до програми № 2

```

Name: Arsenii
Surname: Faryna
Last Name: Petrovych
Position: Student
Birth Year: 2007
Salary: 300
Name: Demyan
Surname: Vorobets
Last Name: Vasylovych
Position: Student
Birth Year: 2007
Salary: 350
File contains:
Arsenii Faryna Petrovych Student 2007 300
Demyan Vorobets Vasylovych Student 2007 350
Employee Faryna removed from file.

After deleting employee surname and adding new employee:
File contains:
Demyan Vorobets Vasylovych Student 2007 350
Danulo Siatetski Romanovych Student 2007 500

```

Рисунок 3.4. Приклад виконання програми №2

Фактично затрачений час 5 годин.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/355

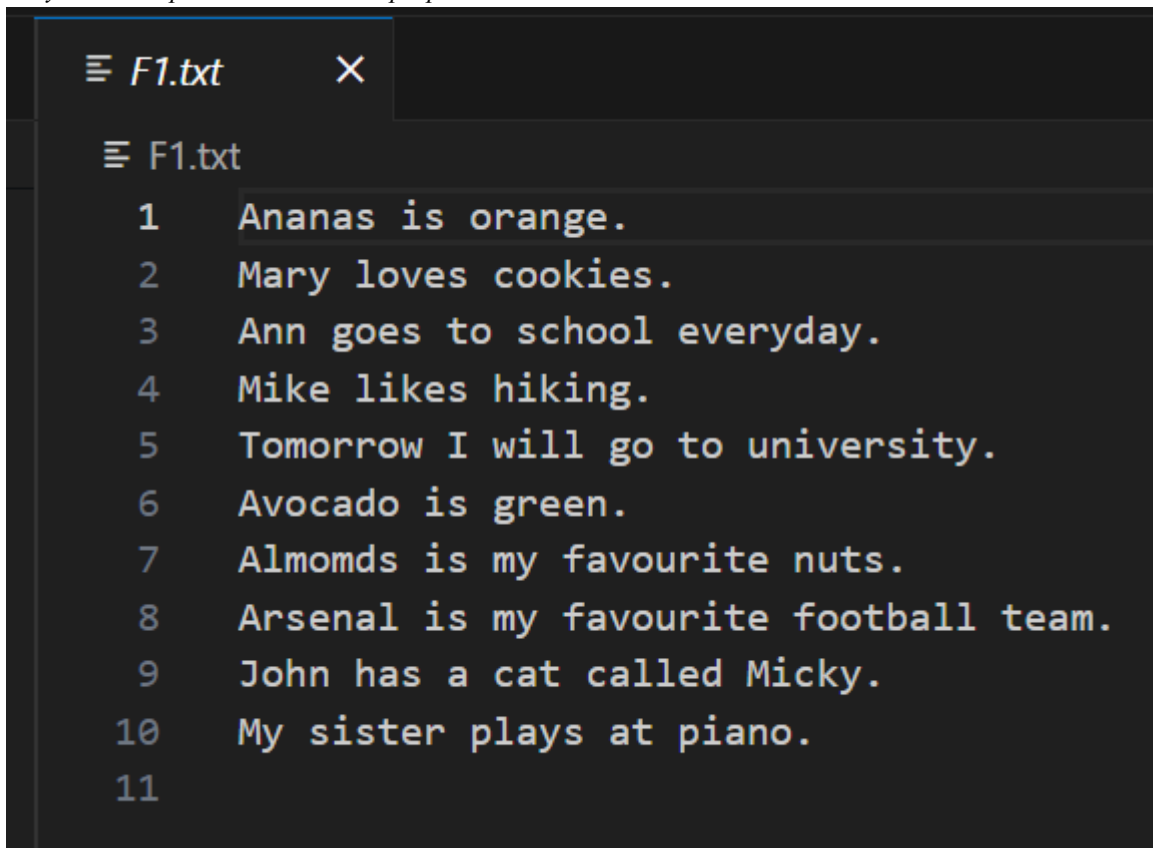
Завдання №3

```
1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include <cctype>
5  #include <sstream>
6
7  using namespace std;
8
9  int count_word_in_F2(const string line){
10
11     stringstream ss(line);
12     string word;
13     int count = 0;
14
15     while (ss >> word){
16         count++;
17     }
18
19     return count;
20 }
21
22 int main (){
23
24     ofstream file1("F1.txt");
25     if(!file1){
26         cerr << "ERROR while opening F1.txt for writing" << endl;
27         exit(1);
28     }
29
30     file1 << "Ananas is orange." << endl;
31     file1 << "Mary loves cookies." << endl;
32     file1 << "Ann goes to school everyday." << endl;
33     file1 << "Mike likes hiking." << endl;
34     file1 << "Tomorrow I will go to university." << endl;
35     file1 << "Avocado is green." << endl;
36     file1 << "Almonds is my favourite nuts." << endl;
37     file1 << "Arsenal is my favourite football team." << endl;
38     file1 << "John has a cat called Micky." << endl;
39     file1 << "My sister plays at piano." << endl;
40
41     file1.close();
42
43     ifstream file1_in("F1.txt");
44     if(!file1_in){
45         cerr << "ERROR while opening F1.txt for reading" << endl;
46         exit(2);
47     }
48
49     ofstream file2("F2.txt");
50     if(!file2){
51         cerr << "ERROR while opening F2.txt for writing" << endl;
52         exit(3);
53     }
54
55     cout << "Lines starting with 'A' will be copied to F2.txt:" << endl;
56
57     string line;
58     while (getline(file1_in, line)) {
59         // Перевіряємо, чи перший символ є літерою 'A' або 'a'
60         if (!line.empty() && (toupper(line[0]) == 'A')) {
61             file2 << line << endl; // Записуємо рядок у F2
62             cout << line << endl; // Виводимо рядок на екран
63         }
64     }
65
66     file1_in.close();
67     file2.close();
68
69     ifstream file2_in("F2.txt");
70     if (!file2_in){
71         cerr << "ERROR while opening file F2.txt for reading" << endl;
72         exit(4);
73     }
74
75     int count_words = 0;
76     while(getline(file2_in, line)){
77         count_words += count_word_in_F2(line);
78     }
79
80     cout << "Numbers of words in F2.txt is " << count_words << endl;
81     file2_in.close();
82
83     return 0;
84 }
```

Рисунок 3.5. Код до програми №3

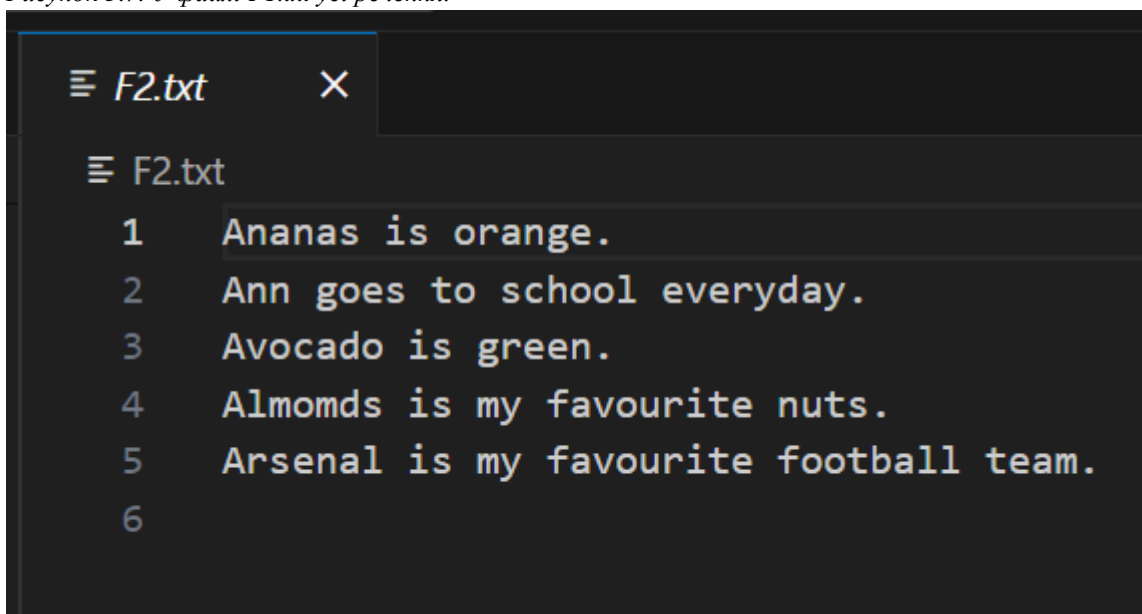
```
Lines starting with 'A' will be copied to F2.txt:  
Ananas is orange.  
Ann goes to school everyday.  
Avocado is green.  
Almomds is my favourite nuts.  
Arsenal is my favourite football team.  
Numbers of words in F2.txt is 22
```

Рисунок 3.6. Приклад виконання програми № 3



```
≡ F1.txt X  
≡ F1.txt  
1 Ananas is orange.  
2 Mary loves cookies.  
3 Ann goes to school everyday.  
4 Mike likes hiking.  
5 Tomorrow I will go to university.  
6 Avocado is green.  
7 Almomds is my favourite nuts.  
8 Arsenal is my favourite football team.  
9 John has a cat called Micky.  
10 My sister plays at piano.  
11
```

Рисунок 3.7. У файлі F1.txt усі речення.



```
≡ F2.txt X  
≡ F2.txt  
1 Ananas is orange.  
2 Ann goes to school everyday.  
3 Avocado is green.  
4 Almomds is my favourite nuts.  
5 Arsenal is my favourite football team.  
6
```

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
https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/355
Завдання №4
З використанням STL

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4
5 using namespace std;
6
7 void operations_with_sets(){
8     int N, M;
9
10    cin >> N;
11    vector<int> arr1(N);
12    for (int i = 0; i < N; i++){
13        cin >> arr1[i];
14    }
15
16    cin >> M;
17    vector<int> arr2(M);
18    for (int i = 0; i < M; i++){
19        cin >> arr2[i];
20    }
21
22    vector<int> result1;
23    vector<int> result2;
24    vector<int> intersection;
25    vector<int> union_result;
26    vector<int> sym_diff_result;
27
28    sort(arr1.begin(), arr1.end());
29    sort(arr2.begin(), arr2.end());
30
31    set_difference(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(result1));
32    set_difference(arr2.begin(), arr2.end(), arr1.begin(), arr1.end(), back_inserter(result2));
33    set_intersection(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(intersection));
34    set_union(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(union_result));
35    set_symmetric_difference(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(sym_diff_result));
36
37    //result for (N - M)
38    cout << result1.size() << endl;
39    for (int num1 : result1){
40        cout << num1 << " ";
41    }
42
43    cout << endl;
44
45    cout << result2.size() << endl;
46
47    //result for (M - N)
48
49    for (int num2 : result2){
50        cout << num2 << " ";
51    }
52    cout << endl;
53
54    //result for intersection
55
56    cout << intersection.size() << endl;
57    for (int num3 : intersection){
58        cout << num3 << " ";
59    }
60    cout << endl;
61
62    //result for union
63
64    cout << union_result.size() << endl;
65    for (int num4 : union_result){
66        cout << num4 << " ";
67    }
68    cout << endl;
69
70    //result for symmetric difference
71
72    cout << sym_diff_result.size() << endl;
73    for (int num5 : sym_diff_result){
74        cout << num5 << " ";
75    }
76    cout << endl;
77 }
78
79 int main(){
80     operations_with_sets();
81
82     return 0;
83 }
```

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


```
5
1 2 3 4 5
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)

Рисунок 3.11. Код до програми №3 (без STL)

```
5
1 2 3 4 5
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 3.12. Приклад виконання програми №3 (без STL)

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
a few seconds ago	C++ 23	Accepted	0.003	1.195	View

Рисунок 3.13. Статус задачі на Algotester

Фактично затрачений час 5 годин.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/355

Завдання №5

З використанням STL

```
1  #include <iostream>
2  #include <algorithm>
3  #include <vector>
4  #include <set>
5
6  using namespace std;
7
8  int main (){
9
10     int N, K;
11     cin >> N >> K;
12     vector<int> arr(N);
13     for (int i = 0; i < N; i++){
14         cin >> arr[i];
15     }
16
17     sort(arr.begin(), arr.end());
18     auto last = unique(arr.begin(), arr.end());
19     arr.erase(last, arr.end());
20
21     int n = arr.size();
22     K = K % n;
23     rotate(arr.begin(), arr.begin() + K, arr.end());
24
25     cout << n << endl;
26
27     for (int i = 0; i < n; i++){
28         cout << arr[i] << " ";
29     }
30     cout << endl;
31
32     return 0;
33 }
```

Рисунок 3.14. Код до програми №4 (з STL)

```
10 3
1 2 2 3 3 3 4 5 6 7
7
4 5 6 7 1 2 3
```

Рисунок 3.15. Приклад виконання програми №4 (з STL)

```

1  #include <iostream>
2  #include <algorithm>
3  #include <vector>
4
5  using namespace std;
6
7  vector<int> difference_arr1_arr2(vector<int> arr1, vector<int> arr2){
8      vector<int> result;
9      for (int a : arr1){
10         bool found = false;
11         for (int b : arr2){
12             if (a == b){
13                 found = true;
14                 break;
15             }
16         }
17         if (!found){
18             result.push_back(a);
19         }
20     }
21     return result;
22 }
23
24 vector<int> difference_arr2_arr1(vector<int> arr2, vector<int> arr1){
25     vector<int> result;
26     for (int b : arr2){
27         bool found = false;
28         for (int a : arr1){
29             if (b == a){
30                 found = true;
31                 break;
32             }
33         }
34         if (!found){
35             result.push_back(b);
36         }
37     }
38     return result;
39 }
40
41 vector<int> intersection(vector<int> arr1, vector<int> arr2){
42     vector<int> result;
43     for(int a : arr1){
44         for (int b : arr2){
45             if (a == b){
46                 result.push_back(a);
47                 break;
48             }
49         }
50     }
51     return result;
52 }
53
54 vector<int> union_array(vector<int> arr1, vector<int> arr2){
55     vector<int> result = arr1;
56     for(int b : arr2){
57         bool found = false;
58         for (int a : arr1){
59             if (a == b){
60                 found = true;
61                 break;
62             }
63         }
64         if (!found){
65             result.push_back(b);
66         }
67     }
68     return result;
69 }
70

```

```

70
71 vector<int> symmetric_difference(vector<int> arr1, vector<int> arr2){
72     vector<int> result;
73     for (int a : arr1){
74         bool found = false;
75         for (int b : arr2){
76             if (a == b){
77                 found = true;
78                 break;
79             }
80         }
81         if (!found){
82             result.push_back(a);
83         }
84     }
85
86     for (int b : arr2){
87         bool found = false;
88         for (int a : arr1){
89             if (a == b){
90                 found = true;
91                 break;
92             }
93         }
94         if (!found){
95             result.push_back(b);
96         }
97     }
98     return result;
99 }
100
101 int main (){
102
103     int N, M;
104
105     cin >> N;
106     vector<int> arr1(N);
107     for(int i = 0; i < N; i++){
108         cin >> arr1[i];
109     }
110
111     cin >> M;
112     vector<int> arr2(M);
113     for(int i = 0; i < M; i++){
114         cin >> arr2[i];
115     }
116
117     sort(arr1.begin(), arr1.end());
118     sort(arr2.begin(), arr2.end());
119
120     vector<int> diff1 = difference_arr1_arr2(arr1, arr2);
121     vector<int> diff2 = difference_arr2_arr1(arr2, arr1);
122     vector<int> intersection1 = intersection(arr1, arr2);
123     vector<int> union_set = union_array(arr1, arr2);
124     vector<int> sym_diff = symmetric_difference(arr1, arr2);
125
126     cout << diff1.size() << endl;
127     for (int num : diff1) {
128         cout << num << " ";
129     }
130     cout << endl;
131
132     cout << diff2.size() << endl;
133     for (int num : diff2) {
134         cout << num << " ";
135     }
136     cout << endl;
137
138     cout << intersection1.size() << endl;
139     for (int num : intersection1) {
140         cout << num << " ";
141     }
142     cout << endl;
143
144     cout << union_set.size() << endl;
145     for (int num : union_set) {
146         cout << num << " ";
147     }
148     cout << endl;
149
150     cout << sym_diff.size() << endl;
151     for (int num : sym_diff) {
152         cout << num << " ";
153     }
154     cout << endl;
155
156     return 0;
157 }

```

Рисунок 3.16. Код до програми №4 (без STL)

```
10 3
1 2 2 3 3 3 4 5 6 7
7
4 5 6 7 1 2 3
```

Рисунок 3.17. Приклад виконання програми №4 (без STL)

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
7 hours ago	C++ 23	Accepted	0.003	1.199	View

Рисунок 3.18. Статус задачі на Algotester

Фактично затрачений час 4 години.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/355

Завдання №6

```

1  #include <iostream>
2  #include <vector>
3  #include <set>
4  #include <algorithm>
5
6  using namespace std;
7
8  #define pb insert // Замінюємо pb на insert, щоб використовувати set
9
10 char board[8][8];
11
12 // Перевірка на валідні координати
13 bool validCoordinate(int x, int y) {
14     return x >= 0 && x < 8 && y >= 0 && y < 8;
15 }
16
17 void attackPawn(int x, int y, set<char>& attackers) {
18     if (validCoordinate(x - 1, y - 1) && board[x - 1][y - 1] == 'P') {
19         attackers.insert('P'); // Pawn attacks from upper left
20     }
21
22     if (validCoordinate(x - 1, y + 1) && board[x - 1][y + 1] == 'P') {
23         attackers.insert('P'); // Pawn attacks from upper right
24     }
25 }
26
27 // Атака конем
28 void attackKnight(int x, int y, set<char>& attackers) {
29     vector<pair<int, int>> knightMoves = {
30         {-2, -1}, {-2, 1}, {2, -1}, {2, 1},
31         {-1, -2}, {1, -2}, {-1, 2}, {1, 2}
32     };
33
34     for (auto move : knightMoves) {
35         int new_x = x + move.first;
36         int new_y = y + move.second;
37         if (validCoordinate(new_x, new_y) && board[new_x][new_y] == 'N') {
38             attackers.insert('N');
39         }
40     }
41 }
42
43 void attackBishop(int x, int y, set<char>& attackers) {
44     // Directions for bishop movement (diagonals)
45     vector<pair<int, int>> directions = { {1, 1}, {1, -1}, {-1, 1}, {-1, -1} };
46
47     for (auto dir : directions) {
48         int dx = dir.first, dy = dir.second;
49         for (int i = 1; i < 8; i++) { // Bishop can move up to 7 squares
50             int new_x = x + i * dx;
51             int new_y = y + i * dy;
52
53             if (!validCoordinate(new_x, new_y)) break; // Stop if out of bounds
54
55             char piece = board[new_x][new_y];
56             if (piece == 'Q' || piece == 'B') { // Queen or bishop attacking
57                 attackers.insert(piece);
58                 break; // Stop further checks in this direction
59             }
60             else if (piece != '0') {
61                 break; // Stop if blocked by any other piece
62             }
63         }
64     }
65 }
66
67

```

```

67
68 void attackRook(int x, int y, set<char>& attackers) {
69     // Перевіряємо весь рядок
70     for (int col = 1; col < 8; col++) {
71         if (col != y) { // Уникаємо перевірки клітинки, де стоїть тура
72             if (board[x][col] == 'R' || board[x][col] == 'Q') {
73                 attackers.insert(board[x][col]);
74                 break; // Тура або ферзь атакують, перериваємо
75             }
76             if (board[x][col] != 'O') break; // Інша фігура блокує
77         }
78     }
79
80     // Перевіряємо весь стовпчик
81     for (int row = 1; row < 8; row++) {
82         if (row != x) { // Уникаємо перевірки клітинки, де стоїть тура
83             if (board[row][y] == 'R' || board[row][y] == 'Q') {
84                 attackers.insert(board[row][y]);
85                 break; // Тура або ферзь атакують, перериваємо
86             }
87             if (board[row][y] != 'O') break; // Інша фігура блокує
88         }
89     }
90 }
91 // Атака королевою (поєднання тури та слона)
92 void attackQueen(int x, int y, set<char>& attackers) {
93     attackRook(x, y, attackers); // Check rook-like movements
94     attackBishop(x, y, attackers); // Check bishop-like movements
95 }
96
97 // Атака королем
98 void attackKing(int x, int y, set<char>& attackers) {
99     // Можливі напрямки для короля (по вертикалі, горизонталі і діагоналях)
100     vector<pair<int, int>> kingMoves = {
101         {-1, -1}, {-1, 0}, {-1, 1}, {0, -1},
102         {0, 1}, {1, -1}, {1, 0}, {1, 1}
103     };
104
105     // Перевіряємо кожен напрямок
106     for (auto move : kingMoves) {
107         int new_x = x + move.first;
108         int new_y = y + move.second;
109
110         // Перевірка, чи координати в межах дошки та чи є на новому полі король
111         if (validCoordinate(new_x, new_y) && board[new_x][new_y] == 'K') {
112             attackers.insert('K'); // Додаємо короля до атаки
113         }
114     }
115 }
116
117 int main() {
118     for (int i = 0; i < 8; i++) {
119         for (int j = 0; j < 8; j++) {
120             cin >> board[i][j];
121         }
122     }
123
124     int Q;
125     cin >> Q;
126     while (Q--) {
127         int x, y;
128         cin >> x >> y;
129         x--; y--;
130
131         set<char> attackers;
132
133         if (board[x][y] != 'O') {
134             cout << "X" << endl;
135         }
136         else {
137             attackPawn(x, y, attackers);
138             attackKnight(x, y, attackers);
139             attackBishop(x, y, attackers);
140             attackRook(x, y, attackers);
141             attackQueen(x, y, attackers);
142             attackKing(x, y, attackers);
143
144             if (attackers.empty()) {
145                 cout << "0" << endl;
146             }
147             else {
148                 for (char c : attackers) {
149                     cout << c;
150                 }
151                 cout << endl;
152             }
153         }
154     }
155     return 0;
156 }
157
158 }
159
160

```

Рисунок 3.15. Код до програми №5

```

R0000P00
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000

4
1 1
X
1 5
0
1 6
X
1 8
0

```

Рисунок 3.16. Приклад виконання програми №5

Рисунок 3.17. Статус задачі на Algotester

Фактично затрачений час 5 годин.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/355

Завдання №7

```
1  #include <iostream>
2
3  using namespace std;
4
5  enum FileOpResult { Success, Failure };
6
7  FileOpResult write_to_file(char* name, char* content){
8
9      if (name == NULL || content == NULL){
10         cout << "Error, because null file provided." << endl;
11         return Failure;
12     }
13
14     FILE* file = fopen(name, "w");
15     if (file == NULL){
16         cout << "Error while writing to file." << endl;
17         return Failure;
18     }
19
20     if (fputs(content, file) == EOF){
21         cout << "Error while writing to file." << endl;
22         fclose(file);
23         return Failure;
24     }
25
26     if (fclose(file) != 0){
27         cout << "Error while closing the file." << endl;
28         return Failure;
29     }
30
31     return Success;
32 }
33
34 int main (){
35
36     char filename[255];
37     char content[1000];
38
39     cout << "Enter the filename" << endl;
40     cin.getline(filename, 255);
41
42     cout << "Enter the content of file: " << endl;
43     cin.getline(content, 1000);
44
45     FileOpResult result = write_to_file(filename, content);
46
47     if (result == Success){
48         cout << "File has been written successfully." << endl;
49     }
50     else{
51         cout << "Error while writing the file." << endl;
52     }
53
54     return 0;
55 }
```

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

```
Enter the filename
file.txt
Enter the content of file:
This my first file
File has been written successfully.
```

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

```
file.txt
1 This my first file
```

Рисунок 3.21. Запис у файлі (задача 1)

```
1  #include <iostream>
2
3  using namespace std;
4
5  enum FileOpResult { Success, Failure };
6
7  FileOpResult copy_file(const char* file_from, const char* file_to) {
8
9      if (file_from == nullptr || file_to == nullptr) {
10         cout << "Error: null file or content provided." << endl;
11         return Failure;
12     }
13
14     FILE* file_from_1 = fopen(file_from, "rb");
15     if (file_from_1 == nullptr) {
16         cout << "Error while reading the file: " << file_from << endl;
17         return Failure;
18     }
19
20     FILE* file_to_1 = fopen(file_to, "wb");
21     if (file_to_1 == nullptr) {
22         cout << "Error while opening file for writing: " << file_to << endl;
23         fclose(file_from_1);
24         return Failure;
25     }
26
27     char buffer[1024];
28     size_t bytes_read;
29
30     while ((bytes_read = fread(buffer, 1, sizeof(buffer), file_from_1)) > 0) {
31         // Якщо помилка при записі
32         if (fwrite(buffer, 1, bytes_read, file_to_1) != bytes_read) {
33             cout << "Error while writing to file: " << file_to << endl;
34             fclose(file_from_1);
35             fclose(file_to_1);
36             return Failure;
37         }
38     }
39
40     if (ferror(file_from_1)) {
41         cout << "Error while reading from source file: " << file_from << endl;
42         fclose(file_from_1);
43         fclose(file_to_1);
44         return Failure;
45     }
46
47     fclose(file_from_1);
48     fclose(file_to_1);
49
50     return Success;
51 }
52
53 int main() {
54
55     char file_from_name[40] = "f1.txt";
56     char file_to_name[40] = "f2.txt";
57
58     FileOpResult result = copy_file(file_from_name, file_to_name);
59
60     if (result == Success) {
61         cout << "From f1.txt to f2.txt data has been copied successfully." << endl;
62     } else {
63         cout << "Error while copying the file." << endl;
64     }
65
66     return 0;
67 }
68
```

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

From f1.txt to f2.txt data has been coppied successfully.

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

Фактично затрачений час 2.5 години.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/355

Завдання №8

```
1  #include <iostream>
2  #include <string>
3  #include <algorithm>
4
5  using namespace std;
6
7  int main (){
8
9      string n;
10     cin >> n;
11
12     string max_num = n;
13     sort(max_num.rbegin(), max_num.rend()); //за спаданням
14
15     string min_num = n;
16     sort(min_num.begin(), min_num.end()); //за зростанням
17
18     if (min_num[0] == '0'){
19         for (int i = 1; i < min_num.size(); i++){
20             if (min_num[i] != '0'){
21                 swap (min_num[0], min_num[i]);
22                 break;
23             }
24         }
25     }
26
27     cout << min_num << " " << max_num << endl;
28
29     return 0;
30 }
31
```

Рисунок 3.24. Код до програми №8

4725202
2022457 7542220

Рисунок 3.25. Приклад виконання програми №8

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.002	1.156	Перегляд

Рисунок 3.26. Статус задачі на Algotester

Фактично затрачений час 1 година.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/355

- 4. Робота з командою:

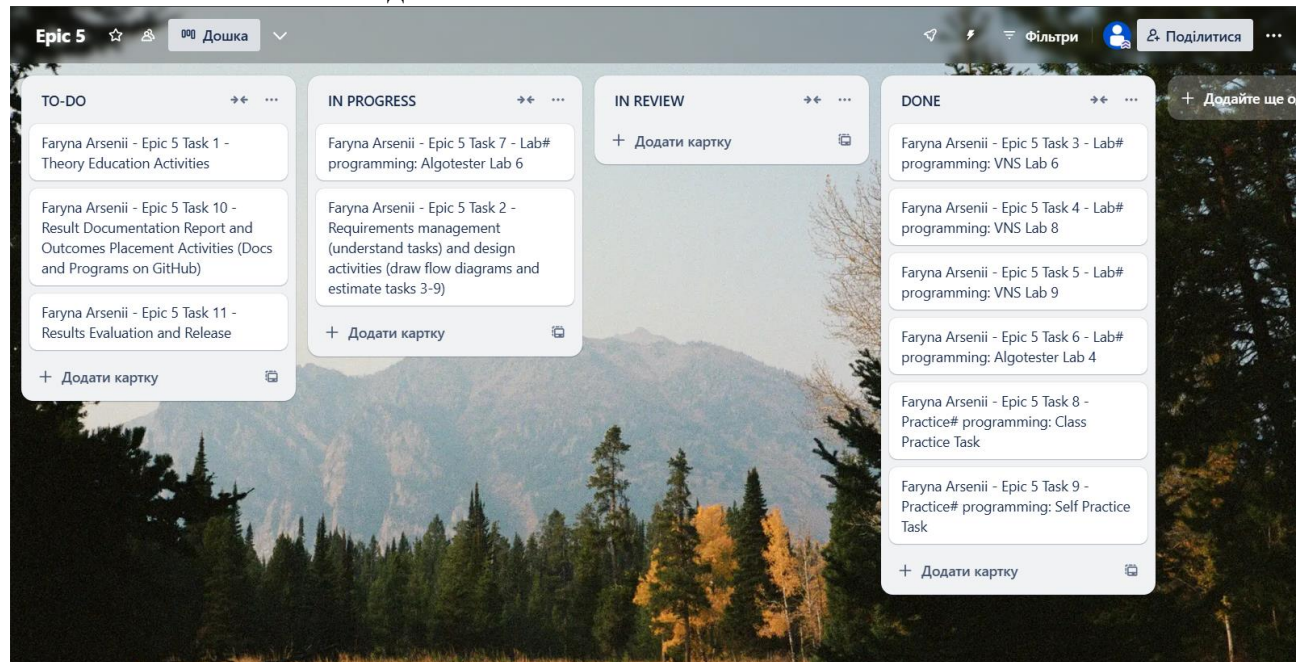


Рисунок 4.1. Командна дошка в Trello

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++. Застосовувавши 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.