**Міністерство освіти і науки України**

**Національний університет "Львівська Політехніка"**

**Кафедра систем штучного інтелекту**

**Епік №5**

з дисципліни

«Основи програмування»

**Виконав:**

студент групи ШІ-11

Гнатюк Ярослав

Львів – 2024 р.

**Епік №5**

**Тема:** Файли. Бінарні Файли. Символи, рядкові змінні та текстові файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.

**Мета роботи:** Ознайомитися з основними принципами роботи з файлами в програмуванні, зокрема з бінарними файлами, текстовими файлами, символами та рядковими змінними. Навчитися використовувати стандартну бібліотеку для маніпуляцій із файлами, а також розглянути методи роботи з ними. Продемонструвати процес створення та використання власних бібліотек для оптимізації роботи з файлами й підвищення повторного використання коду.

**Теоретичні відомості:**

* <stdio.h> fgets() – зчитування вводу користувача
* <stdio.h> getline() – для зчитування тексту з файлів
* <string.h> strcspn() – індекс першого входження специфічного елемента
* Робота з файлами: <https://www.w3schools.com/cpp/cpp_files.asp>
* ofstream – Створити та записати інформацію у файли
* ifstream – Читати інформацію з файлів
* fstream – Комбінація двох попередніх
* <algorithm> sort() – сортування елементів масиву
* <algorithm> unique() – видалення дублікатів з відсортованого масиву
* <algorithm> rotate() – обертає елементів в діапазоні таким чином, щоб елемент у заданій позиції ставав першим елементом.
* <algorithm> partition() – переставлення елементів в діапазоні таким чином, щоб всі елементи, які задовольняють предикат, розміщувалися перед іншими
* Створення та використання власних бібліотек:
* <https://www.youtube.com/watch?v=mnwDpO4zqLA&t=889s&pp=ygUm0LLQu9Cw0YHQvdGWINCx0ZbQsdC70ZbQvtGC0LXQutC4INGBKys%3D>
* ChatGPT

**Виконання роботи**

**Частина 1**

**Завдання №1**

**Назва:** Lab 6 Variant 5

**Опис:** 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), та перетворити рядок таким чином, щоб спочатку в ньому були надруковані тільки букви, а потім тільки цифри, не міняючи порядку проходження символів у рядку.

**Вимоги:** Використати get(s) для вводу рядка.

**Завдання №2**

**Назва:** Lab 8 Variant 5

**Опис:** Сформувати двійковий файл із елементів, заданої у варіанті структури, роздрукувати його вмістиме, 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 "Людина":

- прізвище, ім'я, по батькові;

- рік народження;

- ріст;

- вага.

Знищити усі елементи із зазначеним ростом і вагою, додати елемент після елемента із зазначеним прізвищем.

**Завдання №3**

**Назва:** Lab 9 Variant 5

**Опис:** Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію, а також реалізувати наступні дії:

1) Скопіювати з файлу F1 у файл F2 рядки, починаючи з K до K+5.

2) Підрахувати кількість голосних букв у файлі F2.

**Завдання №4**

**Назва:** Algotester Lab 4 Variant 2 (1, 2)

**Опис:** 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.

**Вимоги:** Реалізувати алгоритм двома способами:

1) З використанням засобів STL (std::unique, std::sort, std::rotate)

2) Зі своєю реалізацією

**Завдання №5**

**Назва:** Algotester Lab 4 Variant 3 (1, 2)

**Опис:** Вам дано масив, який складається з N додатних цілих чисел. Ваше завдання - розділити його на три частини, по остачі від ділення на 3, по зростанню остачі (тобто спочатку йдуть числа, у яких остача 0, далі числа з остачею 1 і тоді нарешті числа з остачею 2). Далі необхідно ті елементи, остача від ділення на 3 яких парна посортувати по зростанню, а ті, у яких остача 1 - по спаданню. Після цього видаліть усі дублікати з масиву. Виведіть результуючий масив.

**Вимоги:** Реалізувати алгоритм двома способами:

1) З використанням засобів STL (власноруч написаний компаратор або std::partition + std::sort + std::unique).

2) Зі своєю реалізацією. Алгоритм сортування можна вибрати будь який, окрім сортування бульбашкою і має працювати за N\*logN часу.

**Завдання №6**

**Назва:** Algotester Lab 6 Variant 3

**Опис:** У Клінта в черговий раз виключилось світло і йому немає чим зайнятися. Так як навіть це не заставить його подивитися збережені відео про програмування на YouTube, він вирішив придумати свою гру на основі судоку.

Гра виглядає так: Є поле розміром N×N, в якому частина клітинок заповнена цифрами, а частина клітинок порожні (позначаються нулем). Також у нього є Q пар координат X та Y.

Завданням гри є написати до кожної координати скільки чисел туди можна вписати (якщо вона пуста) і які це числа (обов’язково впорядковані по зростанню). В клітинку можна вписати лише ті числа, які не зустрічаються в рядку та стовпці, які перетинаються у цій клітинці.

Поле не змінюється!

Якщо є клітинка, в яку не можна вписати жодну цифру — виведіть 0.

Також допускаються рядки та стовпці, в яких цифра записана кілька разів.

**Завдання №7**

**Назва:** Practice work 1

**Опис:**

***Реалізувати функцію створення файла і запису в нього даних:***

enum FileOpResult { Success, Failure, … };

FileOpResult write\_to\_file(char \*name, char \*content);

***Умови задачі:***

- cтворити 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.

***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**

**Назва:** Self practice work

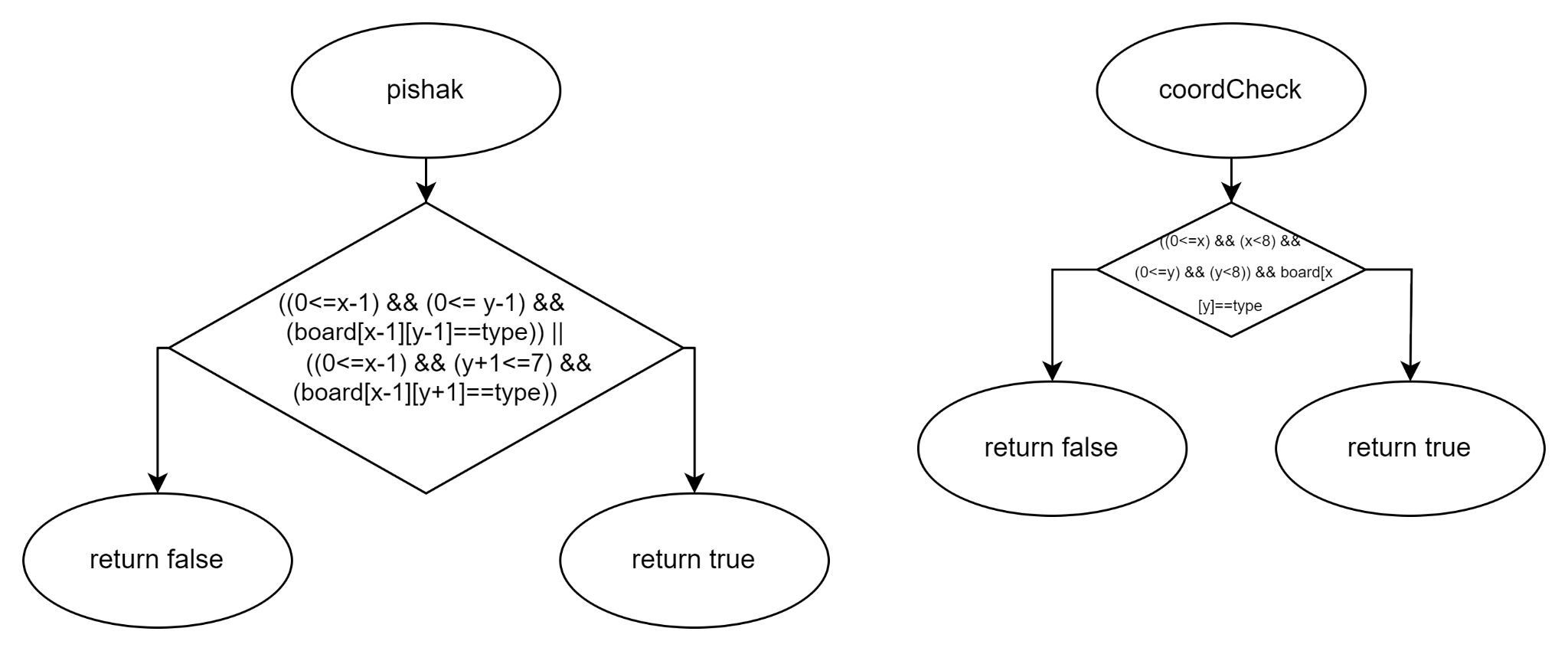
**Опис:** У вас є шахова дошка розміром 8×8 і дуже багато фігур. Кожна клітинка може мати одне із таких значень:

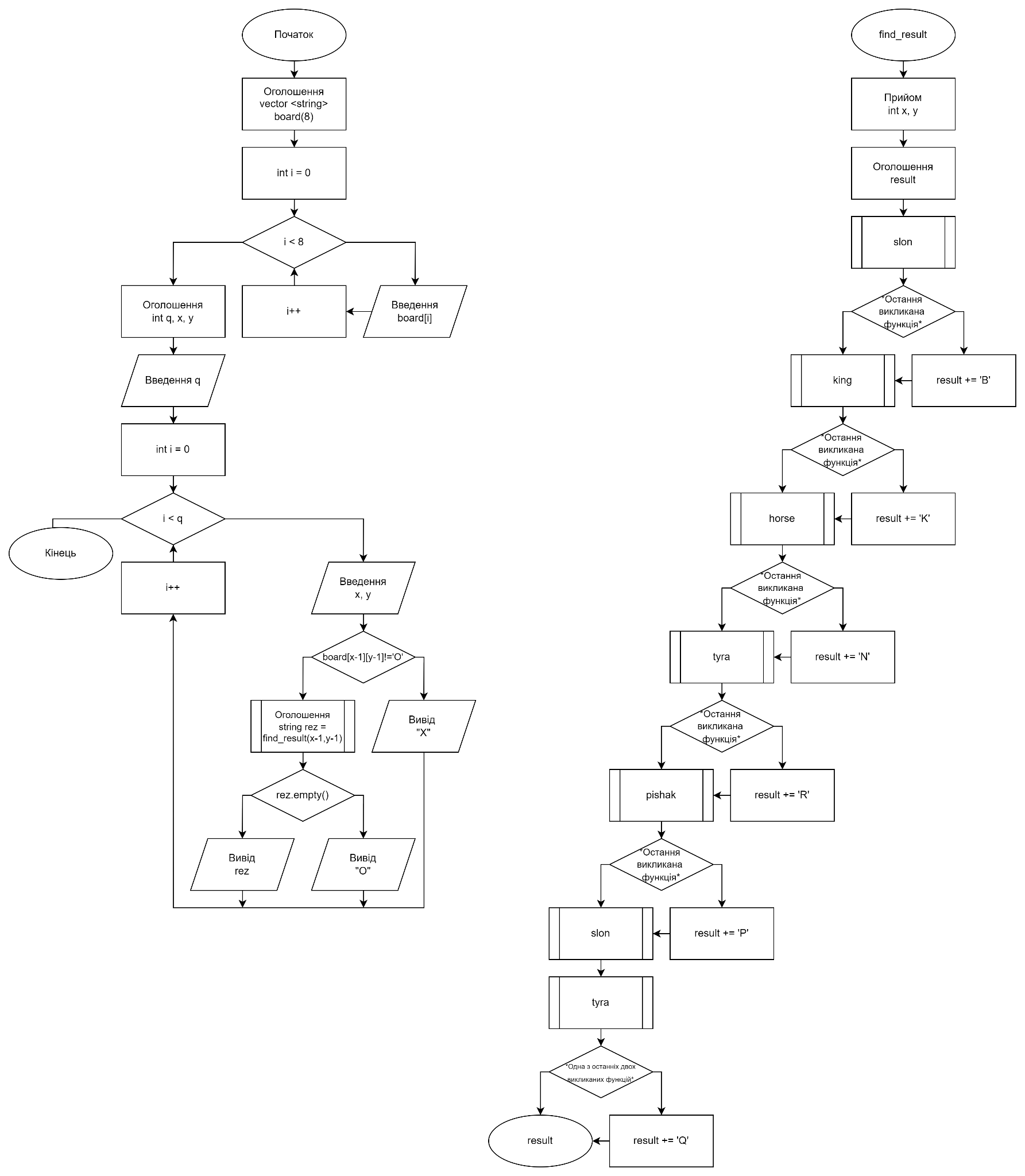
* Пуста клітинка: O
* Пішак: P
* Тура: R
* Кінь: N
* Слон: B
* Король: K
* Королева: Q

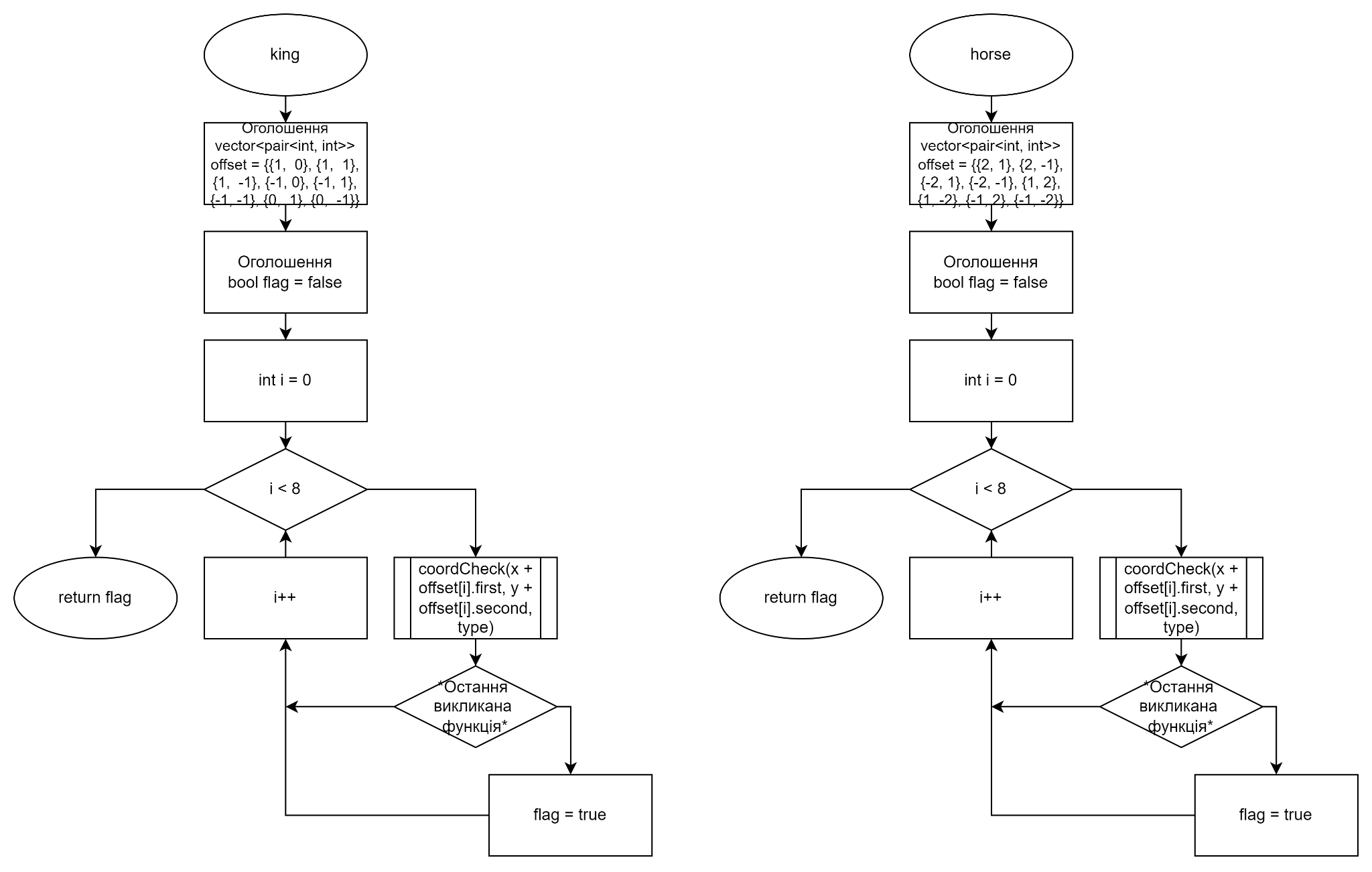
Вам дають позиції фігур на дошці (всі фігури одного кольору, кількість королів може бути > 1). Далі йдуть **Q** запитів з координатами клітинки x,y{x, y}x,y. На кожен запит ви маєте вивести стрічку sis\_isi​ — посортовані за алфавітом букви фігур, які атакують цю клітинку (пішаки атакують вниз). У випадку, якщо на клітинці стоїть якась фігура — виведіть символ **X**. 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.

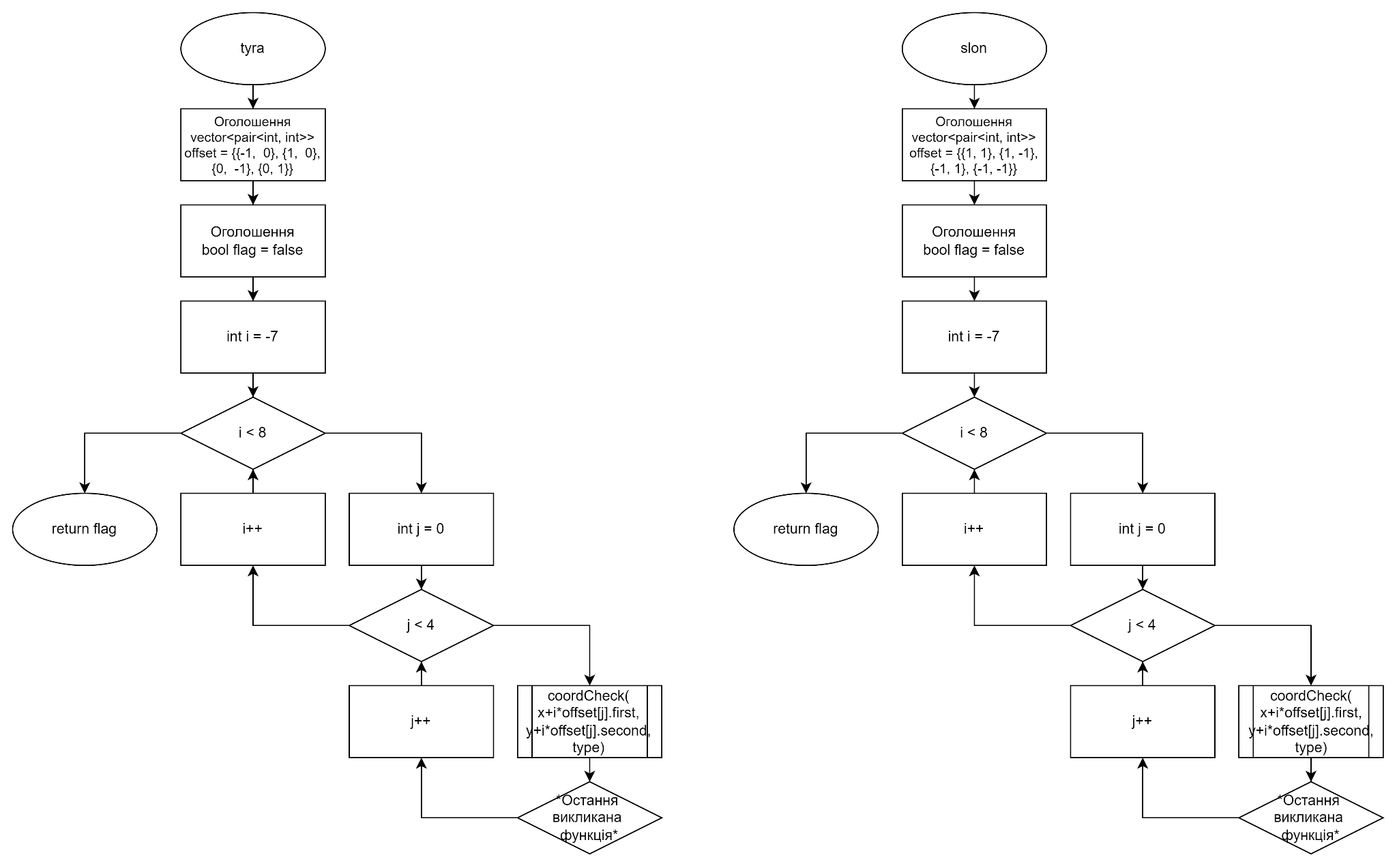
**Частина 2**

**Завдання №7**

****

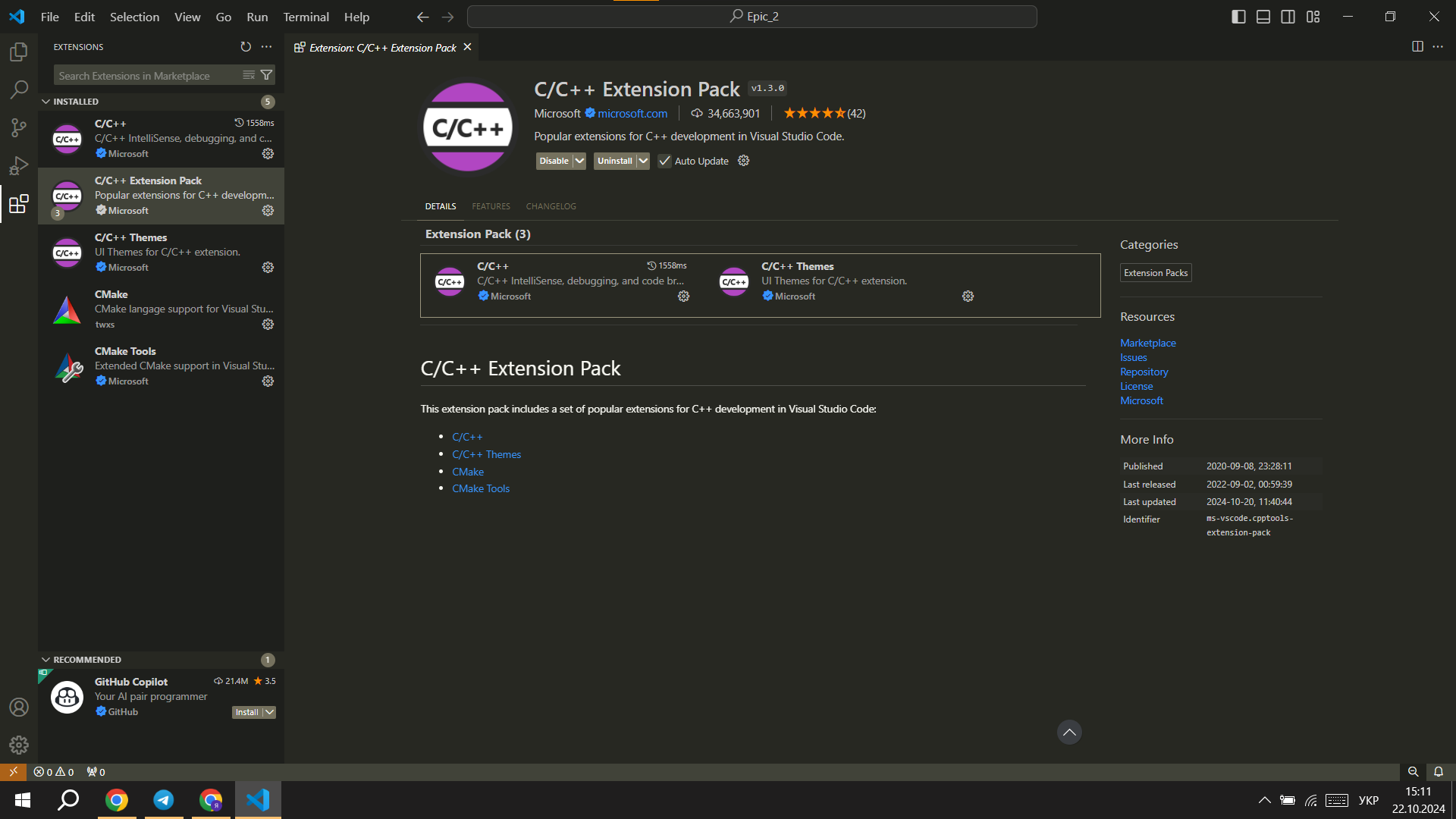
****

****

****

**Частина 3**

Для виконання роботи використовується середовище **Visual Studio Code** зі встановленим розширенням **С/C++ Extension Pack.**

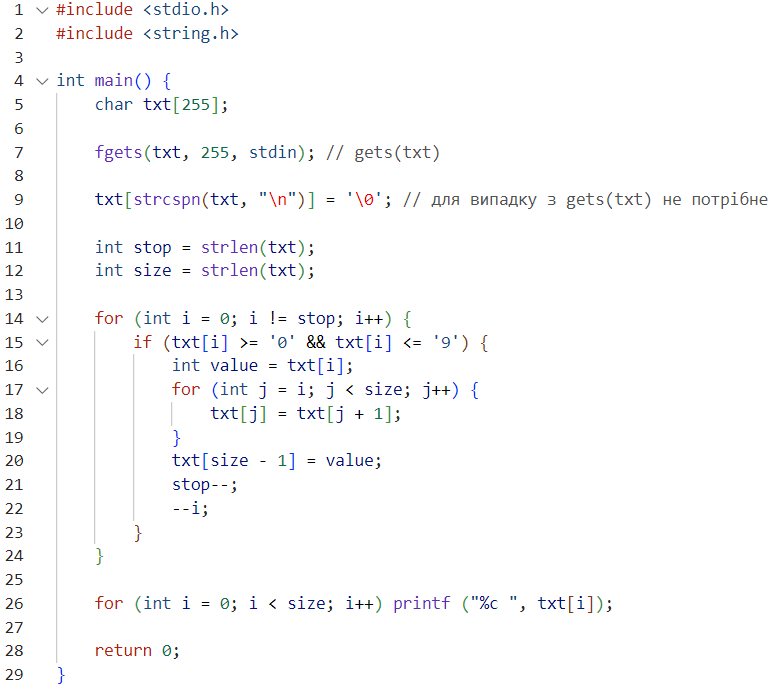
****

**Використані бібліотеки:**

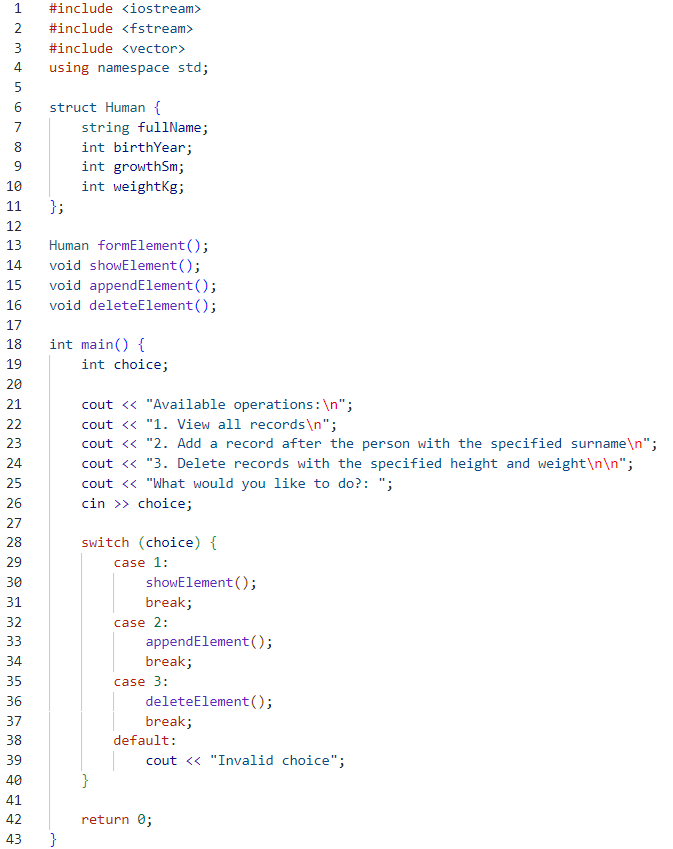
* #include <stdio.h>
* #include <string.h>
* #include <iostream>
* #include <fstream>
* #include <vector>
* #include <random>
* #include <algorithm>
* #include <string>

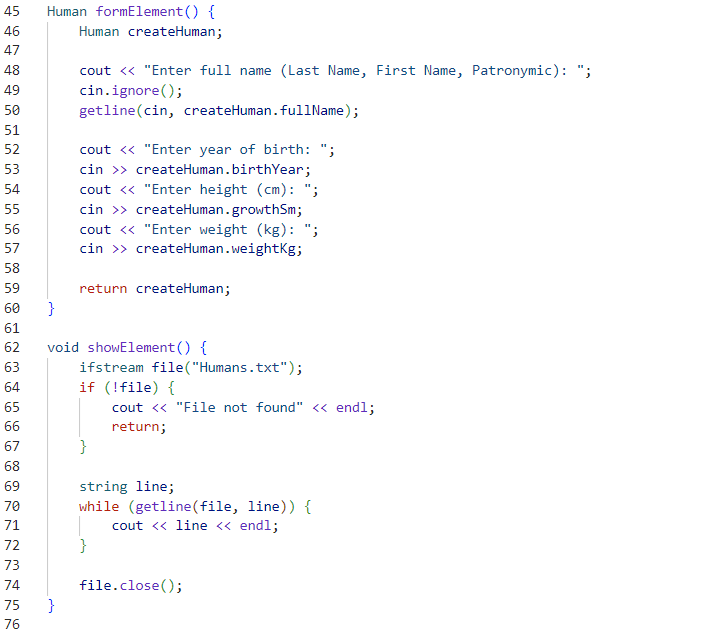
**Частина 4**

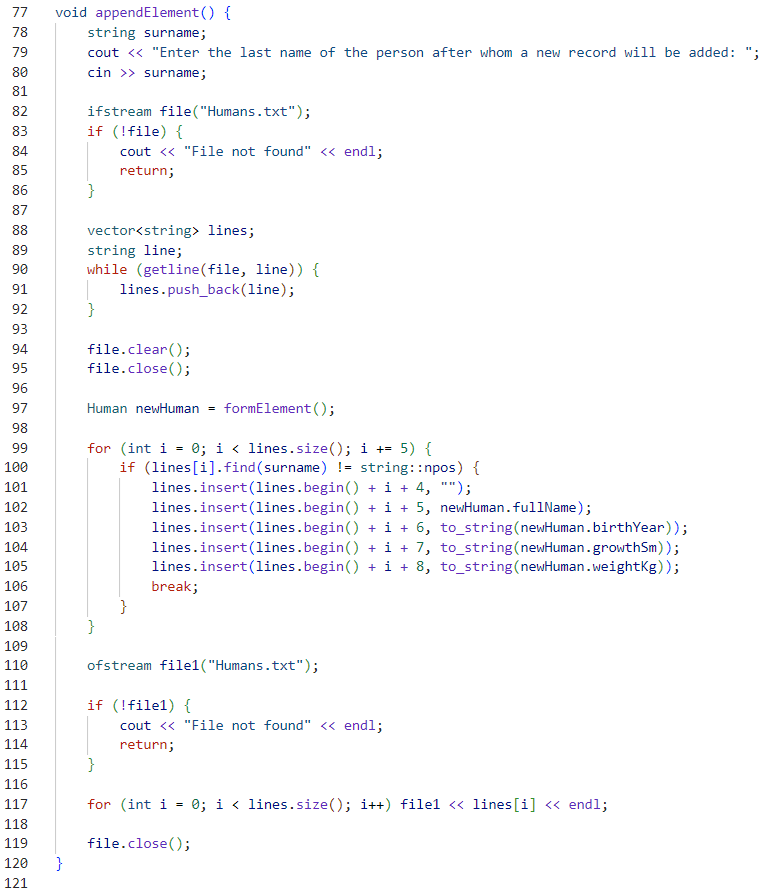
**Завдання №1**

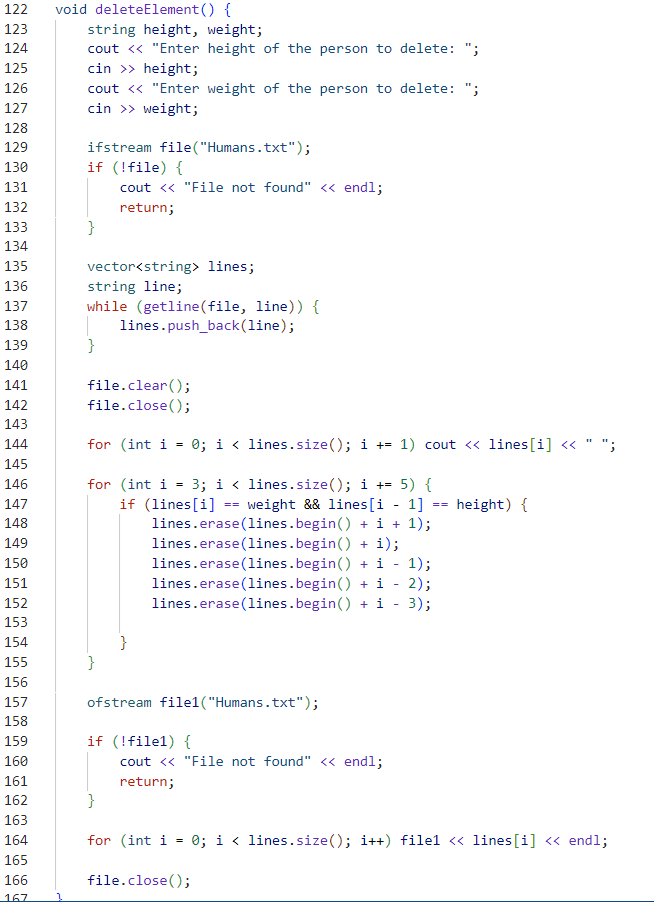
****

**Завдання №2**

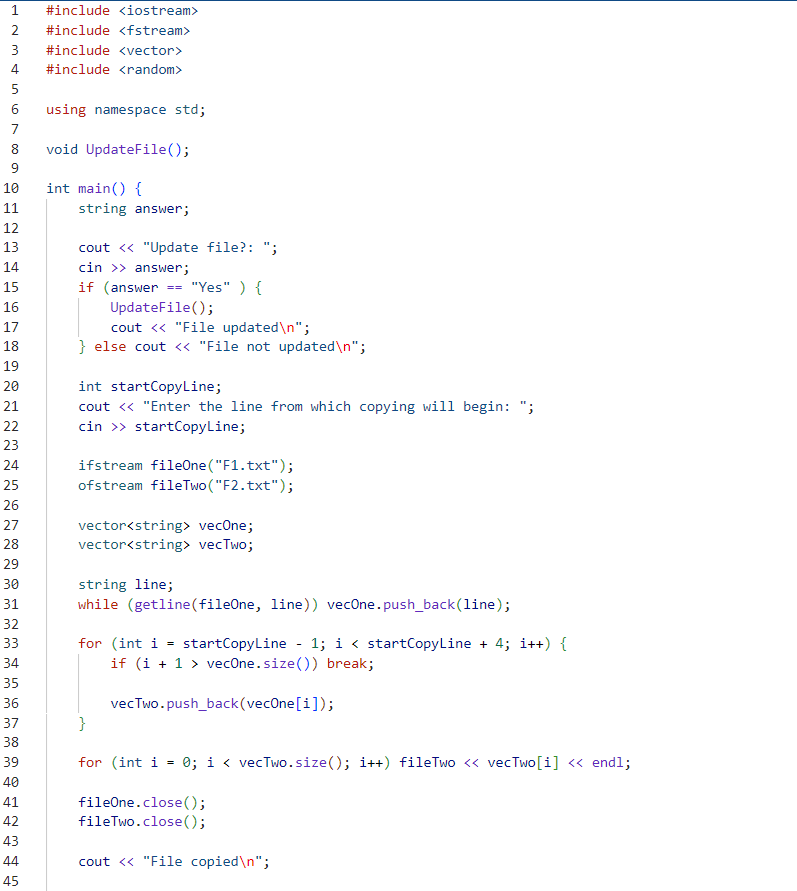
****

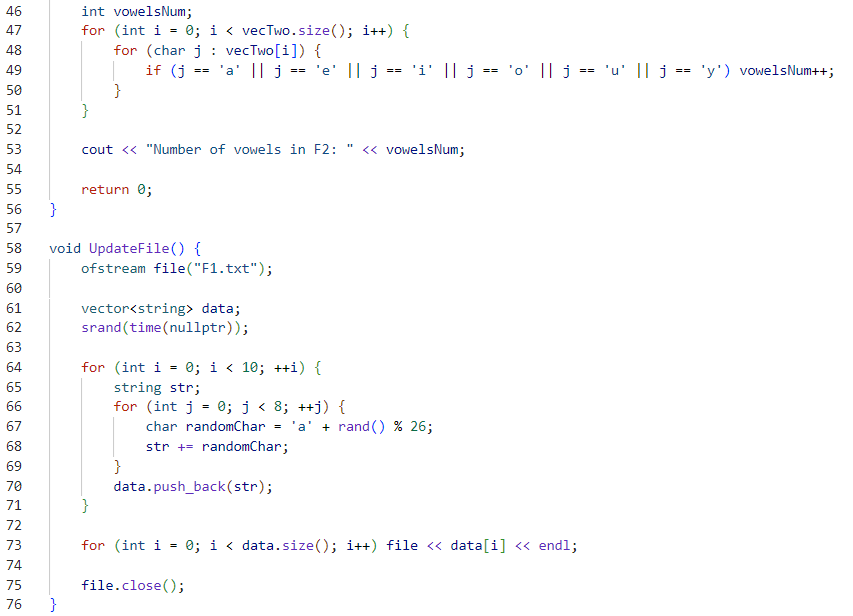
****

****

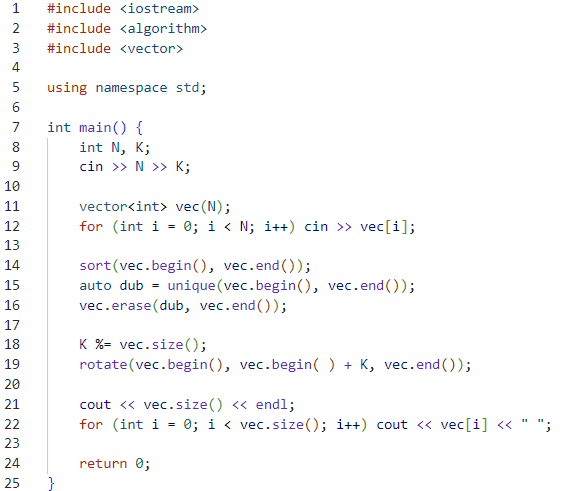
****

**Завдання №3**

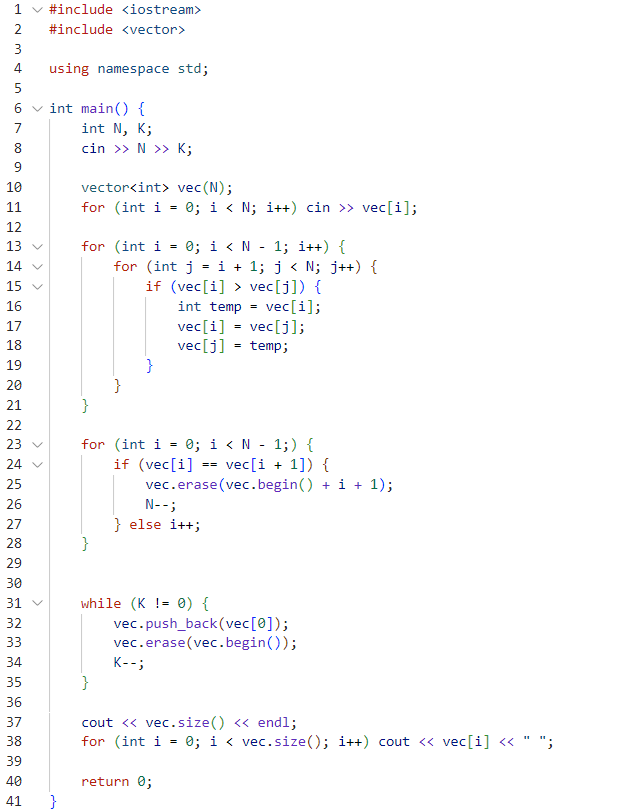
****

****

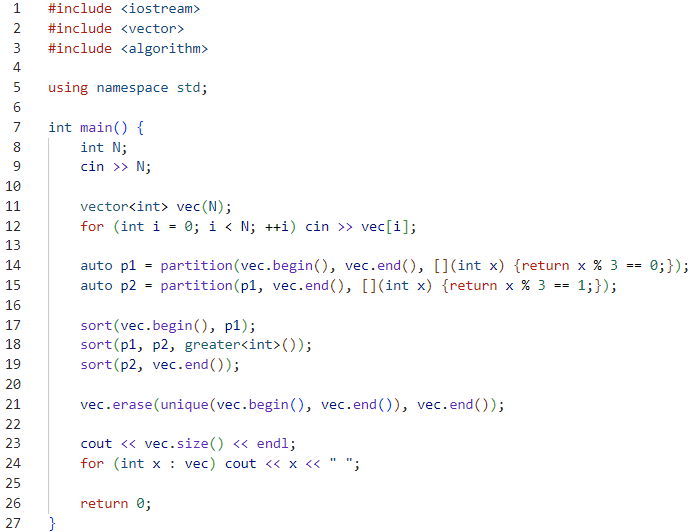
**Завдання №4.1**

****

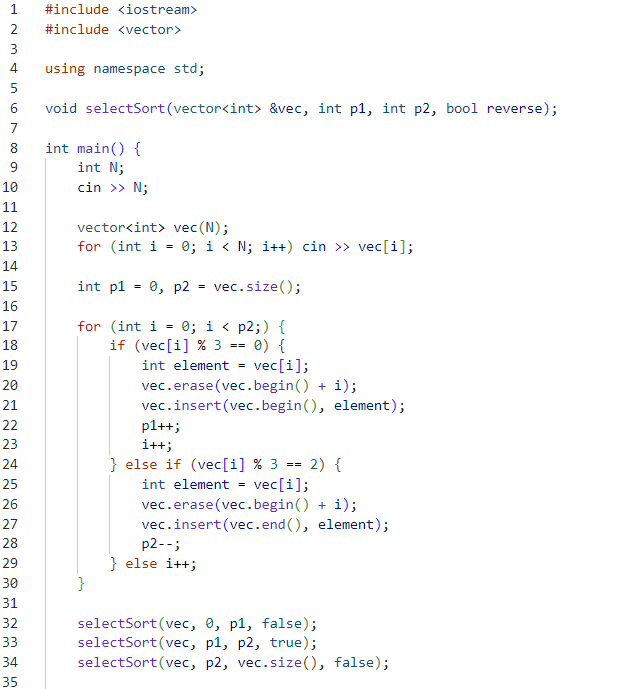
**Завдання №4.2**

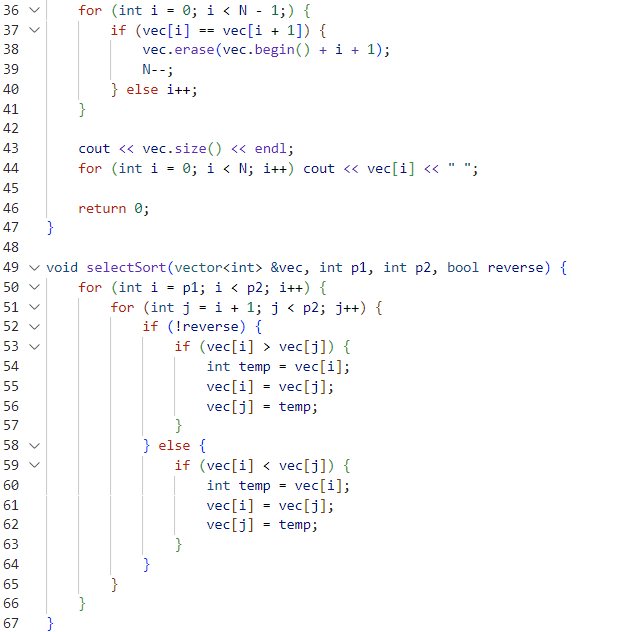
****

**Завдання №5.1**

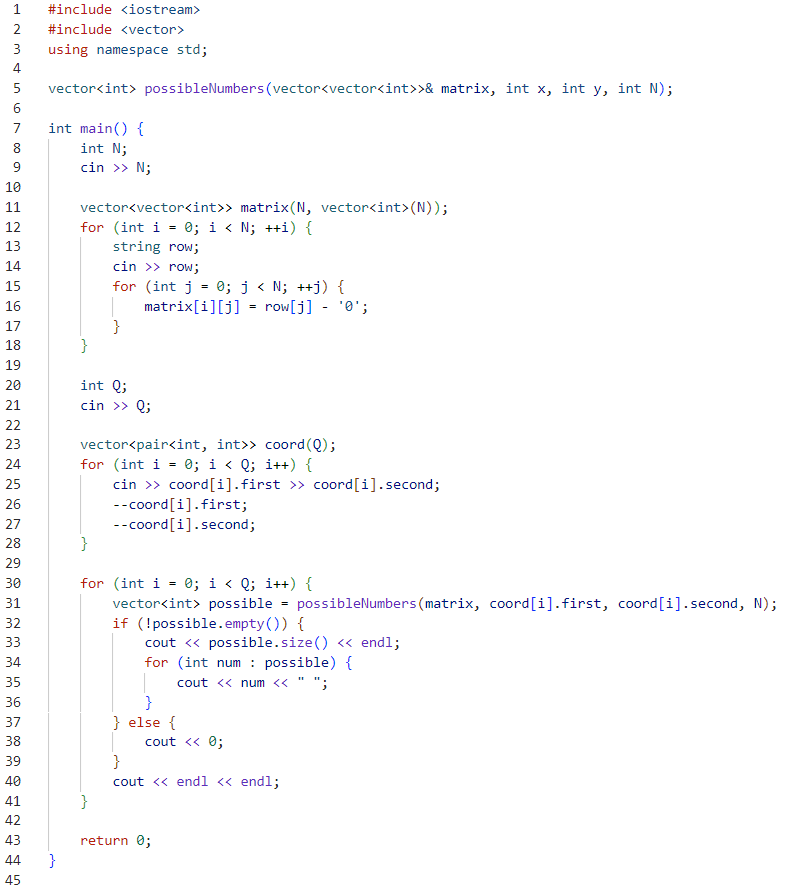
****

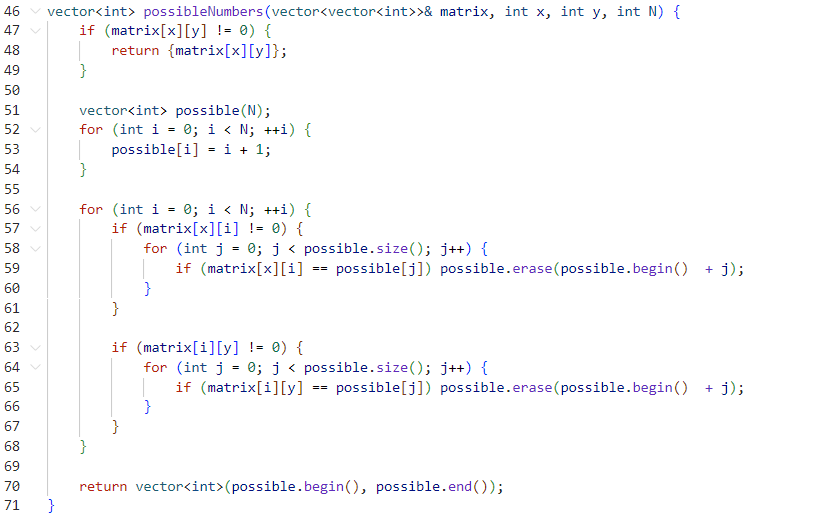
**Завдання №5.2**

****

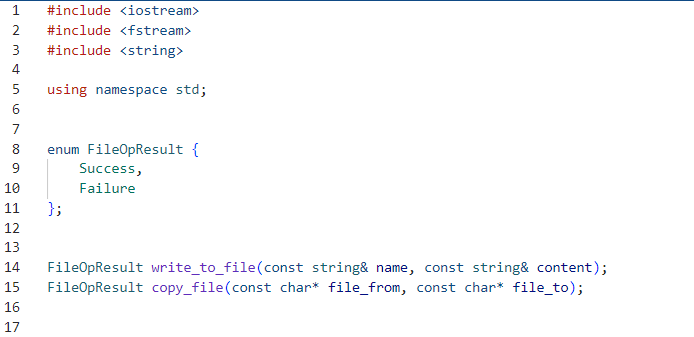
****

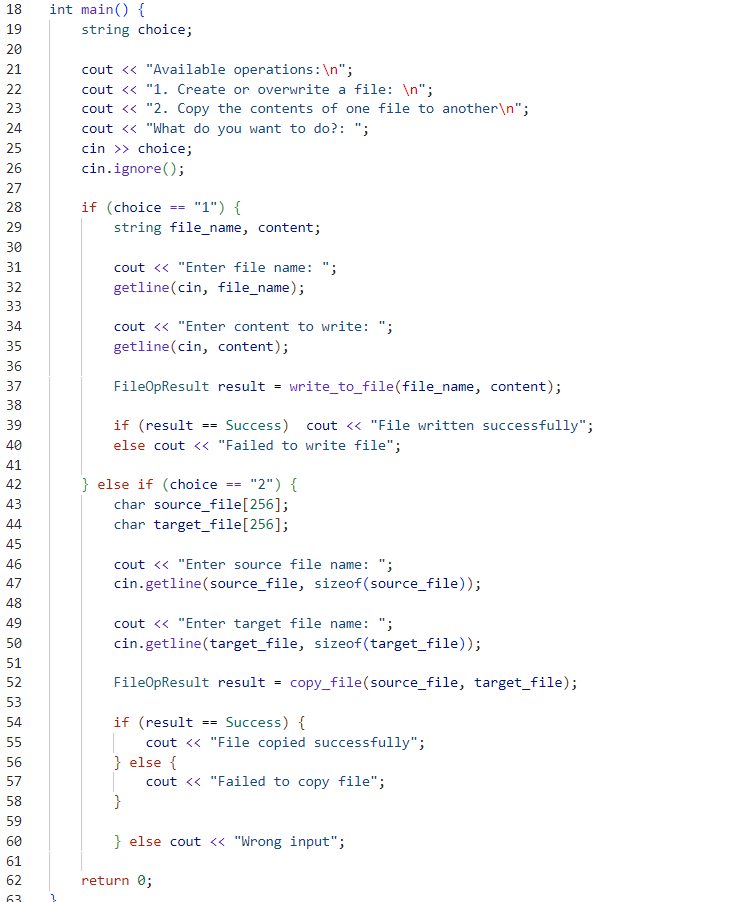
**Завдання №6**

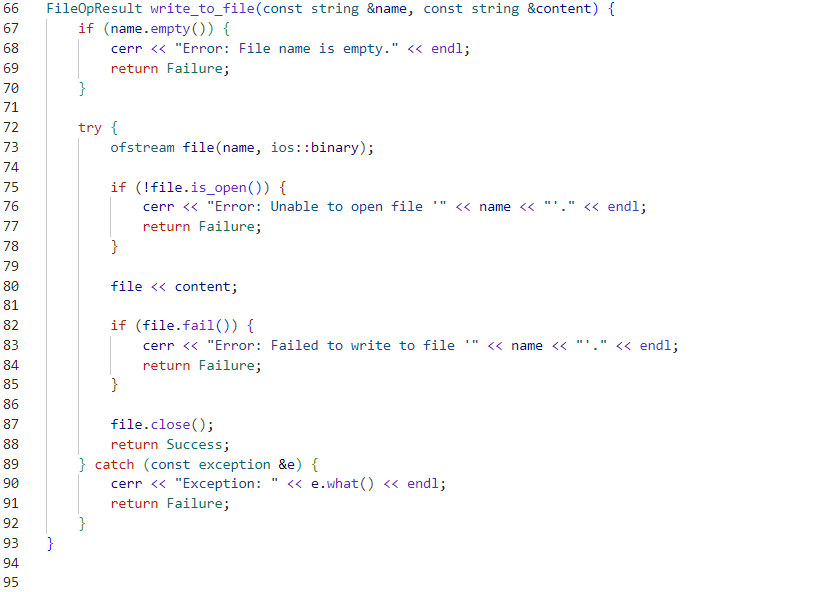
****

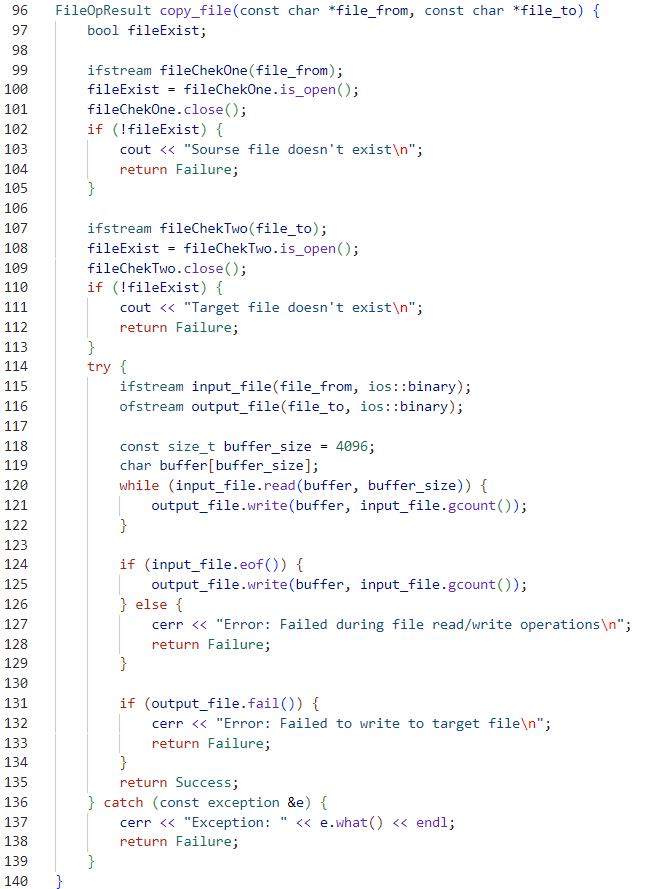
****

**Завдання №7**

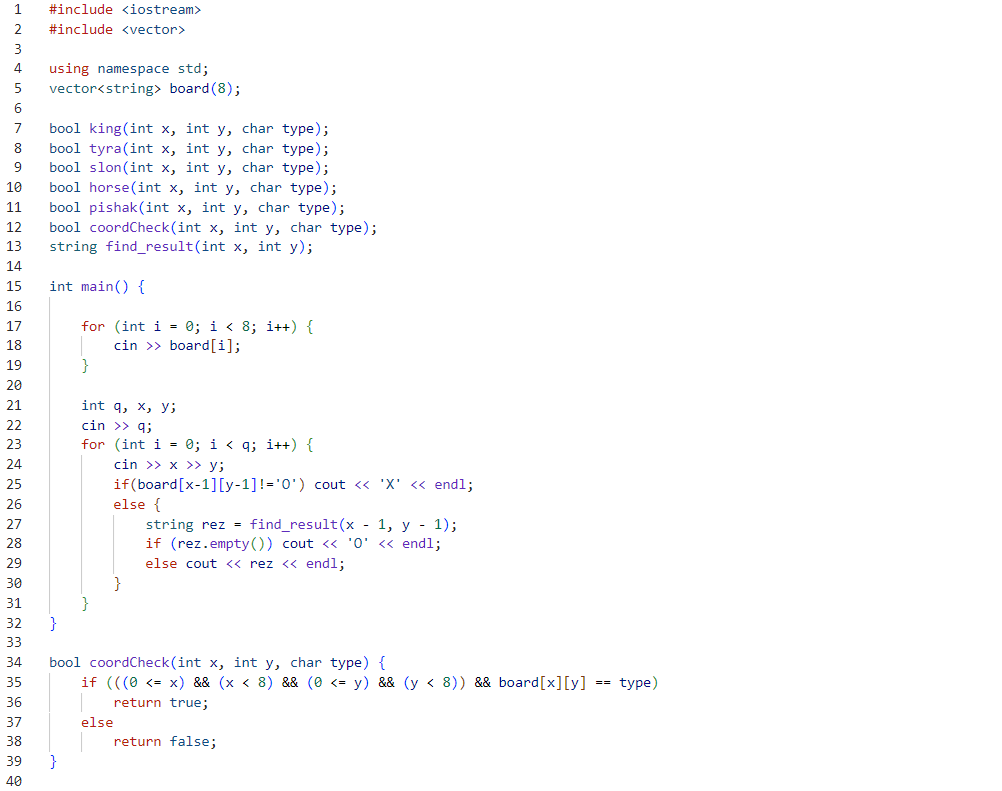
****

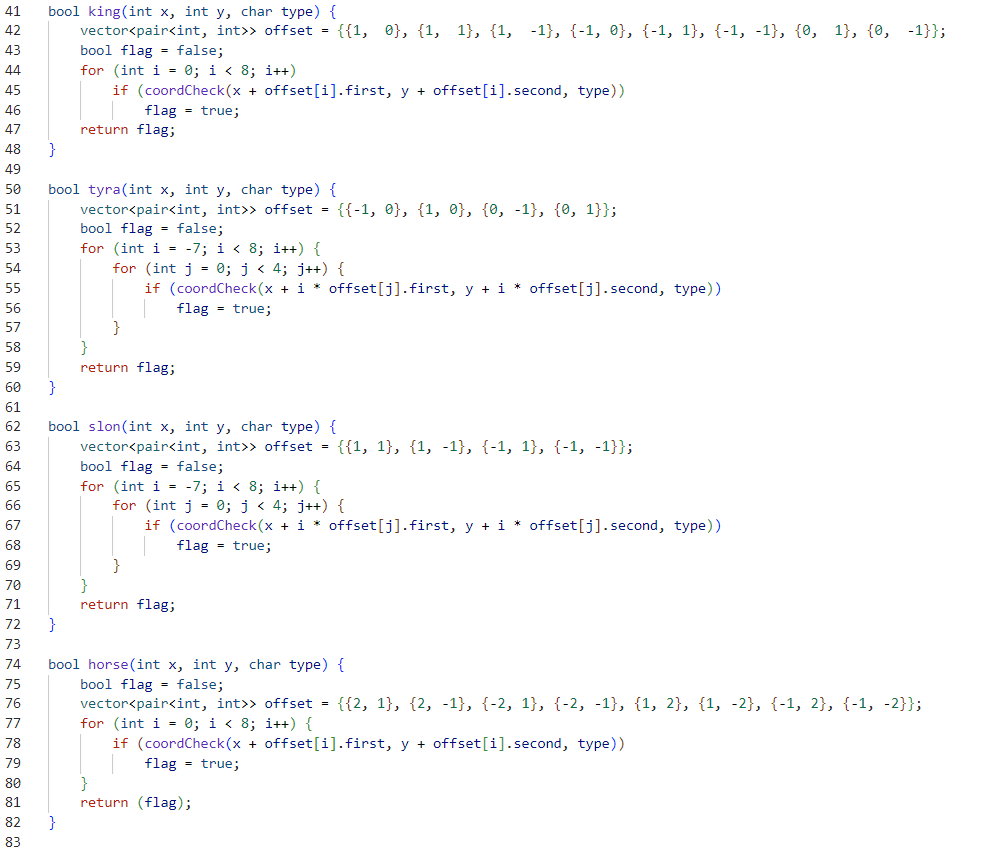
****

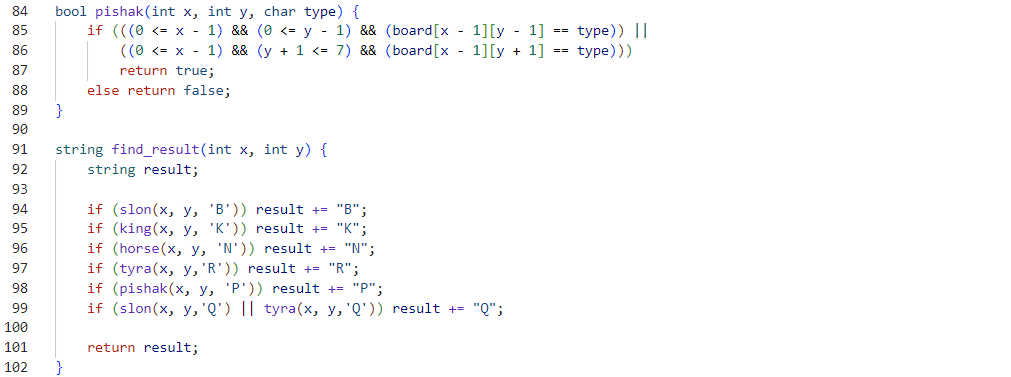
****

****

**Завдання №8**

****

****

****

**Частина 5**

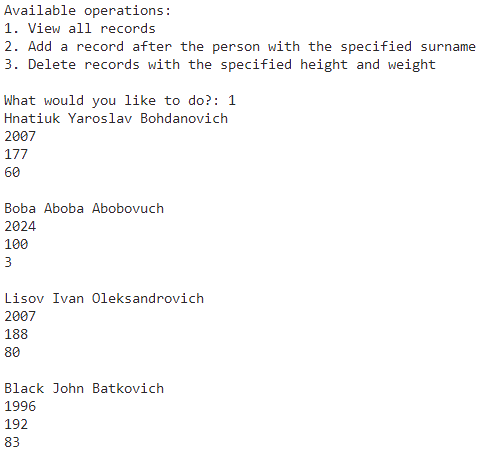
**Завдання №1**

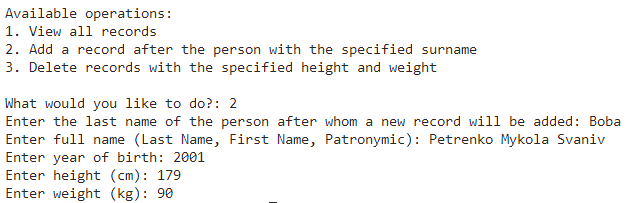
****

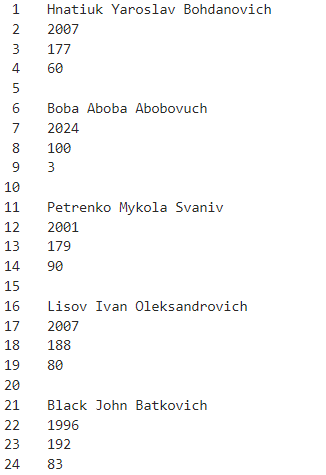
**Орієнтовний час виконання:** 1 год

**Фактично затрачений час:** 1 год

**Завдання №2**

****

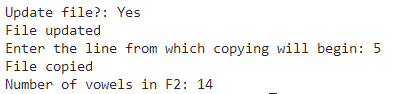
****

****

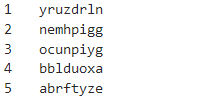
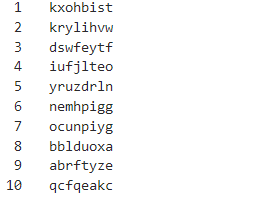
**Орієнтовний час виконання:** 2 год

**Фактично затрачений час:** 3 год

**Завдання №3**

****

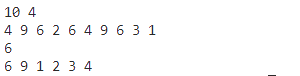
**F1: F2:**

****

**Орієнтовний час виконання:** 2 год

**Фактично затрачений час:** 2 год

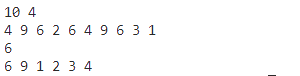
**Завдання №4.1**

****

**Орієнтовний час виконання:** 30 хв

**Фактично затрачений час:** 30 хв

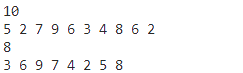
**Завдання №4.2**

****

**Орієнтовний час виконання:** 1 год

**Фактично затрачений час:** 1 год

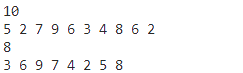
**Завдання №5.1**

****

**Орієнтовний час виконання:** 30 хв

**Фактично затрачений час:** 30 хв

**Завдання №5.2**

****

**Орієнтовний час виконання:** 1 год

**Фактично затрачений час:** 1 год

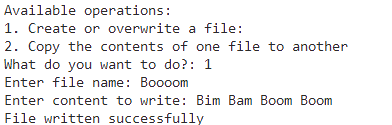
**Завдання №6**

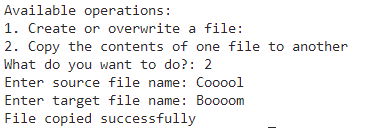
****

**Орієнтовний час виконання:** 1 год

**Фактично затрачений час:** 1.5 год

**Завдання №7**

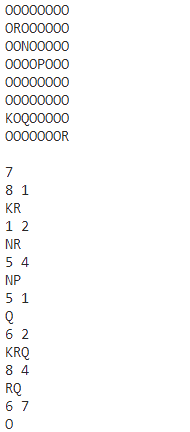
****

****

**Орієнтовний час виконання:** 3 год

**Фактично затрачений час:** 5 год

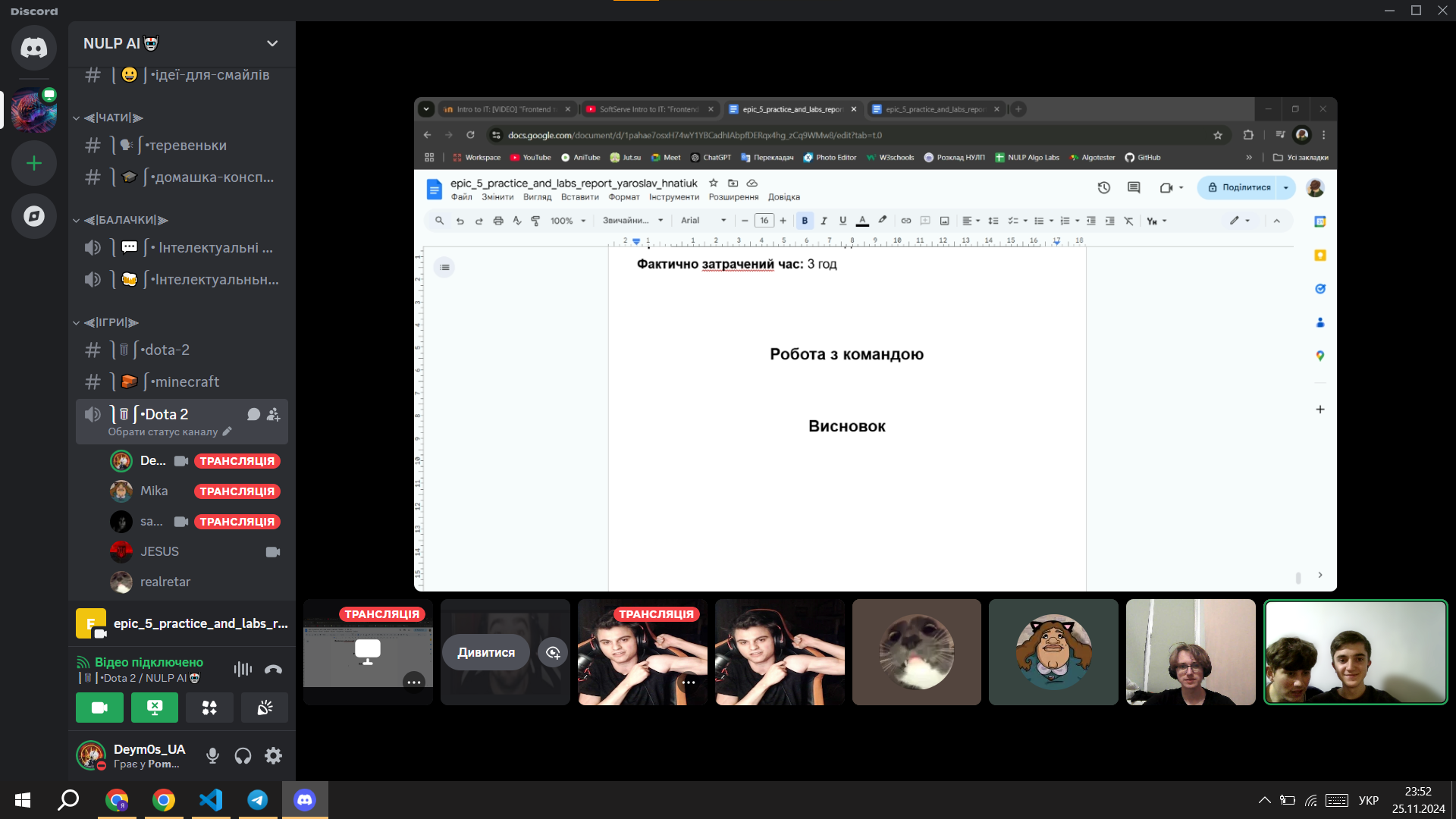
**Завдання №8**

****

**Орієнтовний час виконання:** 1 год

**Фактично затрачений час:** 3 год

**Робота з командою**

****

**Висновок**

Я ознайомився з основами роботи з текстовими та бінарними файлами, стандартними бібліотеками та методами їх використання. Навчився працювати з символами, рядковими змінними й файлами, а також створювати та застосовувати власні бібліотеки для оптимізації коду.