

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



**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

**Виконала:**

Студентка групи ШІ-11

Гуменюк Анастасія Олександрівна

**Тема роботи:** Вивчення роботи з файлами у C++, зокрема текстових і бінарних файлів, а також основних операцій, таких як відкриття, читання, запис і закриття. Дослідження роботи з файловими дескрипторами, перевірки стану файлу та обробки помилок. Огляд роботи з символами та рядковими змінними (типи `char` і `string`), а також базових операцій з рядками: конкатенація, порівняння, пошук. Розгляд особливостей роботи з текстовими файлами (зчитування, обробка рядків) і форматуванням тексту. Використання стандартної бібліотеки для роботи з файлами (потoki `ifstream`, `ofstream`, `fstream`). Вивчення принципів створення власних бібліотек у C++ та правил їх структурування і застосування.

**Мета роботи:** Навчитися основним принципам роботи з файлами у C++ та розібратися з текстовими і бінарними файлами, включаючи операції відкриття, читання, запису та закриття. Опанувати перевірку стану файлу. Дослідити базові операції з символами та рядковими змінними, такі як конкатенація, порівняння і пошук у рядках. Навчитися формувати текстові файли при записі даних і застосовувати методи для обробки рядків з файлу. Зрозуміти принципи роботи з бінарними файлами. Ознайомитися з використанням стандартної бібліотеки для роботи з файлами (`ifstream`, `ofstream`, `fstream`). Спробувати створити власні бібліотеки у C++ та організувати їх структуру для полегшення роботи з файлами у майбутніх проектах.

### Теоретичні відомості:

Теоретичні відомості з переліком важливих тем:

- 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.

Індивідуальний план опрацювання теорії:

Тема №1: Вступ до Роботи з Файлами.

- Джерела:  
<https://youtu.be/FeNqHytI0fA?si=uU-vKhKXGhEFFofB>
- 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

- Статус: Ознайомлений

Тема №2: Символи і Рядкові Змінні.

- Джерела:

<https://www.youtube.com/watch?v=1DtZCv7xfb8&t=955s>

- Що опрацьовано:

- Робота з char та string: основні операції і методи
- Стрічкові літерали та екранування символів
- Конкатенація, порівняння та пошук у рядках

- Статус: Ознайомлений

- Початок опрацювання теми: 11.11.2024.

- Звершення опрацювання теми: 11.11.2024 (35хв.).

Тема №3: Текстові Файли.

- Джерела:

[https://youtu.be/SSNJ7alki-E?si=EAXljt\\_gw6hCG5hR](https://youtu.be/SSNJ7alki-E?si=EAXljt_gw6hCG5hR)

<https://acode.com.ua/urok-220-bazovyj-fajlovyj-vvid-i-vyvid/>

- Що опрацьовано:

- Особливості читання та запису текстових файлів
- Обробка рядків з файлу: getline, ignore, peek
- Форматування тексту при записі: setw, setfill, setprecision
- Обробка помилок при роботі з файлами

- Статус: Ознайомлений

Тема №4: Бінарні Файли.

- Джерела:

<https://studfile.net/preview/5994719/page:7/>

<https://acode.com.ua/urok-221-randomnyj-fajlovyj-vvid-i-vyvid/>

- 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, seekr
- Серіалізація об'єктів у бінарний формат

- Статус: Ознайомлений

Тема №5: Стандартна бібліотека та робота з файлами.

- Джерела:

[https://youtu.be/L7JGsi4sryc?si=\\_cEc4SG0Qu9c0NBE](https://youtu.be/L7JGsi4sryc?si=_cEc4SG0Qu9c0NBE)

[https://youtu.be/FvbiCKv1AHo?si=TZodZ2hAGH\\_dKDzy](https://youtu.be/FvbiCKv1AHo?si=TZodZ2hAGH_dKDzy)

- Що опрацьовано:

- Огляд стандартної бібліотеки для роботи з файлами

- Потоки вводу/виводу: ifstream, ofstream, fstream
- Обробка помилок при роботі з файлами
- Статус: Ознайомлений

## **Виконання роботи:**

### **Завдання №1 VNS Lab 6 – 13**

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.

Завдання:

Перетворити рядок таким чином, щоб цифри кожного слова в ньому були відсортовані за спаданням.

### **Завдання №2 VNS Lab 8 – 13**

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.

#### **13. Структура "Спортивна команда":**

- назва;
- місто;
- кількість гравців;
- кількість набраних очків.

Знищити всі елементи з кількістю очків менше заданого, додати 2 елементи на початок файлу.

### **Завдання №3 VNS Lab 9 – 13**

Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію. Виконати завдання.

Завдання:

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

### **Завдання №4 Algotester Lab 4 – 2**

Вам дано масив  $a$  з  $N$  цілих чисел.

Спочатку видалить 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  $a$

### Обмеження

$$1 \leq N, K \leq 1000$$

$$0 \leq a_i \leq 100$$

## Завдання №5 Algotester Lab 6 – 2

У вас є шахова дошка розміром  $8 \times 8$  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 сі - 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 row $i$  – 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.

Обмеження

$|row_i|=N$

$row_i \in \{O, P, R, N, B, K, Q\}$

$1 \leq Q \leq 64$

$1 \leq x, y \leq 8$

### Завдання №6 Class Practice Task

#### 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.

#### Мета задачі

**Розуміння методів роботи з файлами:** Робота з файлами є одним з базових навиків програмування. Реалізація функції створення та запису в файл допоможе освоїти практичні навички роботи з файлами з використанням стандартної бібліотеки C++. Для виконання завдання студент має навчитись використовувати методи відкриття файла, запису масиву даних у файл, закриття файла та обробки помилок чи станів операції на кожному з етапів.

**Розвиток алгоритмічне мислення:** Запис у файл включає набір операцій, які якнайкраще вкладаються в концепцію алгоритма, як списка детальних кроків. Імплементация цієї функції наочно демонструє створення алгоритмів у програмуванні.

**Освоїти навички роботи з текстовими стрічками:** завдання допоможе освоїти роботу з C стрічками, які є масивами з нульовим символом в кінці. Типові концепції при роботі з C стрічками це арифметика вказівників,

ітерація по стрічці, копіювання частини стрічки, розбиття на токени по заданому символу.

**Розвинути навички розв'язувати задачі:** Запис у файл може супроводжуватись набором станів (немає доступу на створення, недостатньо місця, ін.), які необхідно передбачити у алгоритмі. Аналіз цих станів дозволяє розвинути навик розв'язання інженерних задач у програмуванні.

1) **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.

**Мета задачі**

**Розуміння методів роботи з файлами:** Робота з файлами є одним з базових навиків програмування. Реалізація функції копіювання вмісту файла допоможе освоїти практичні навички роботи з файлами з використанням стандартної бібліотеки C++. Для виконання завдання студент має навчитись використовувати методи відкриття файла, читання вмісту файла, запису масиву даних у файл, закриття файла та обробки помилок чи станів операції на кожному з етапів.

**Розвиток алгоритмічне мислення:** Читання та запис у файл включає набір операцій, які якнайкраще вкладаються в концепцію алгоритма, як списка детальних кроків. Імплементация цієї функції наочно демонструє створення алгоритмів у програмуванні.

**Освоїти навички роботи з потоком даних:** завдання допоможе освоїти роботу з потоками даних (концепція реалізована в STL як набір класів \*stream\* - fstream, stringstream, streambuf та ін.). Концепція потоку даних дозволяє абстрагувати роботу з джерелами та приймачами даних та писати з її допомогою високорівневий код.

**Розвинути навички розв'язувати задачі:** Операції читання з файла та запис у файл можуть супроводжуватись набором різних станів (немає доступу на читання чи створення, недостатньо місця, ін.), які необхідно передбачити у алгоритмі. Аналіз цих станів дозволяє розвинути навик розв'язання інженерних задач у програмуванні.

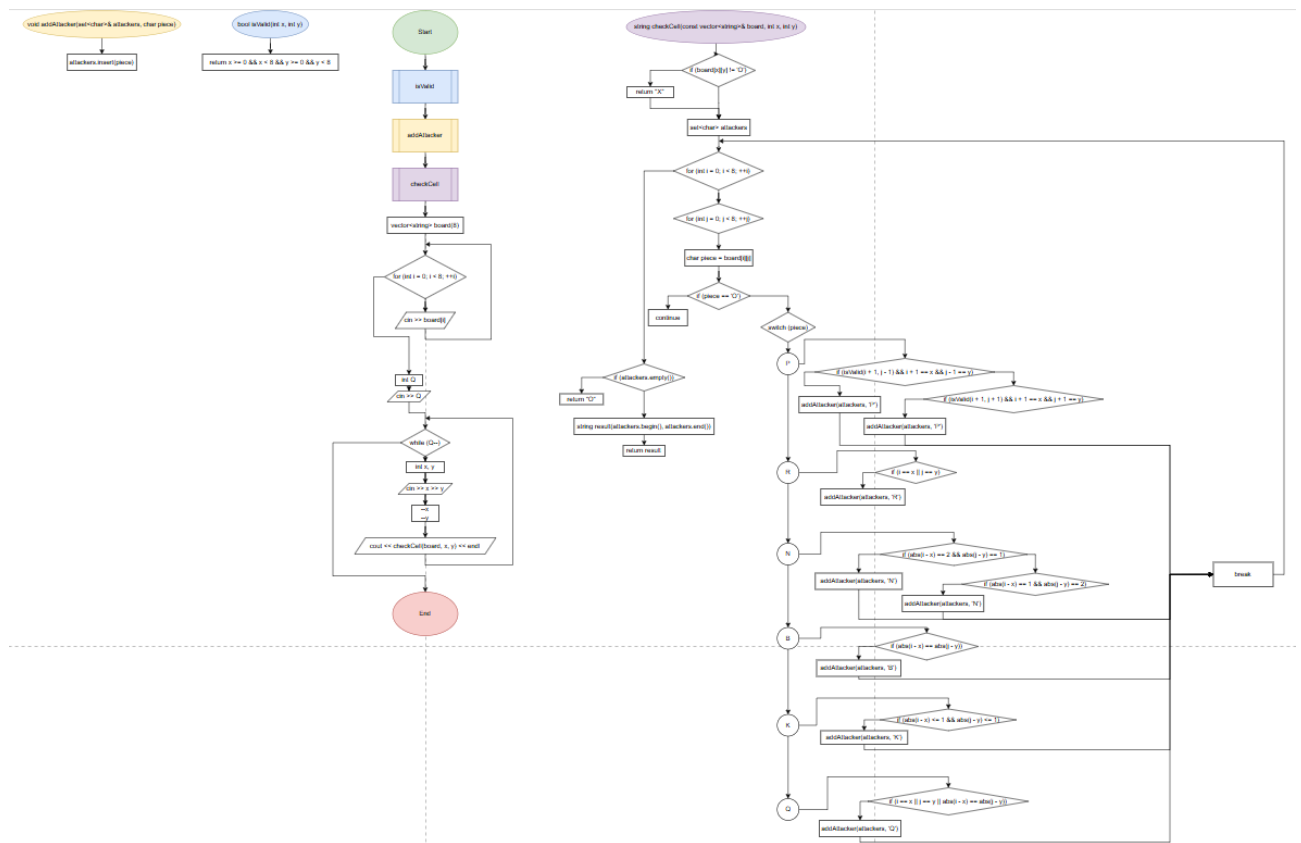
**Завдання №7 Self Practice Work**

Ця програма дозволяє зручно працювати з текстовими файлами. Вона надає користувачу такі можливості:

- **Створення нового текстового файлу** – програма автоматично створює файл, готовий до роботи.
- **Запис даних** – користувач може вводити текст, який буде збережено у файлі.
- **Перегляд вмісту** – в будь-який момент можна відкрити файл і переглянути його вміст без необхідності використовувати сторонні редактори.

Інтерфейс простий та інтуїтивно зрозумілий, що робить роботу з файлами максимально комфортною.

## Дизайн виконання завдань:



**Код програм з посиланням на зовнішні ресурси:  
Завдання №1 VNS Lab 6 – 13**



```

1  #include <iostream>
2  #include <string>
3  #include <sstream>
4  #include <cctype>
5  #include <algorithm>
6
7  using namespace std;
8
9  void sort_digits_desc(string &word) {
10     string digits;
11
12     for (char ch : word) {
13         if (isdigit(ch)) {
14             digits += ch;
15         }
16     }
17
18     sort(digits.begin(), digits.end(), greater<char>());
19
20     size_t digit_index = 0;
21     for (char &ch : word) {
22         if (isdigit(ch)) {
23             ch = digits[digit_index++];
24         }
25     }
26 }
27
28 int main() {
29     string input;
30     cout << "Введіть рядок: ";
31     getline(cin, input);
32
33     if (!input.empty() && input.back() == '.') {
34         input.pop_back();
35     }
36
37     stringstream ss(input);
38     string word, result;
39
40     while (ss >> word) {
41         sort_digits_desc(word);
42         result += word + " ";
43     }
44
45     if (!result.empty()) {
46         result.pop_back();
47         result += ".";
48     }
49
50     cout << "Результат: " << result << endl;
51
52     return 0;
53 }

```

**Завдання №2 VNS Lab 8 – 13**

```

1  #include <iostream>
2  #include <fstream>
3  #include <vector>
4  #include <string>
5
6  using namespace std;
7
8  struct Team {
9      char name[50];
10     char city[50];
11     int players;
12     int points;
13 };
14
15 void writeTeam(ofstream &file, const Team &team) {
16     file.write(reinterpret_cast<const char*>(&team), sizeof(Team));
17 }
18
19 bool readTeam(ifstream &file, Team &team) {
20     return file.read(reinterpret_cast<char*>(&team), sizeof(Team)) ? true : false;
21 }
22
23 void createBinaryFile(const string &filename) {
24     ofstream file(filename, ios::binary);
25     if (!file) {
26         cerr << "Не вдалося відкрити файл для запису." << endl;
27         return;
28     }
29
30     Team teams[] = {
31         {"Ukraine", "Lviv", 11, 50},
32         {"Sport", "kyiv", 12, 30},
33         {"Gym", "Mykolaiv", 10, 20},
34         {"Power", "Kharkiv", 9, 70}
35     };
36
37     for (const auto &team : teams) {
38         writeTeam(file, team);
39     }
40
41     file.close();
42     cout << "Файл створено успішно." << endl;
43 }
44
45 void printBinaryFile(const string &filename) {
46     ifstream file(filename, ios::binary);
47     if (!file) {
48         cerr << "Не вдалося відкрити файл для читання." << endl;
49         return;
50     }
51
52     Team team;
53     while (readTeam(file, team)) {
54         cout << "Назва: " << team.name << ", Місто: " << team.city
55             << ", Гравці: " << team.players << ", Очки: " << team.points << endl;
56     }
57
58     file.close();
59 }

```

```

61 void deleteTeamsWithFewPoints(const string &filename, int minPoints) {
62     ifstream inputFile(filename, ios::binary);
63     if (!inputFile) {
64         cerr << "Не вдалося відкрити файл для читання." << endl;
65         return;
66     }
67
68     vector<Team> teams;
69     Team team;
70     while (readTeam(inputFile, team)) {
71         if (team.points >= minPoints) {
72             teams.push_back(team);
73         }
74     }
75     inputFile.close();
76
77     ofstream outputFile(filename, ios::binary | ios::trunc);
78     if (!outputFile) {
79         cerr << "Не вдалося відкрити файл для запису." << endl;
80         return;
81     }
82
83     for (const auto &t : teams) {
84         writeTeam(outputFile, t);
85     }
86     outputFile.close();
87 }
88
89 void addTeamsToFile(const string &filename, const vector<Team> &newTeams) {
90     ifstream inputFile(filename, ios::binary);
91     if (!inputFile) {
92         cerr << "Не вдалося відкрити файл для читання." << endl;
93         return;
94     }
95
96     vector<Team> teams(newTeams);
97     Team team;
98     while (readTeam(inputFile, team)) {
99         teams.push_back(team);
100     }
101     inputFile.close();
102
103     ofstream outputFile(filename, ios::binary | ios::trunc);
104     if (!outputFile) {
105         cerr << "Не вдалося відкрити файл для запису." << endl;
106         return;
107     }
108
109     for (const auto &t : teams) {
110         writeTeam(outputFile, t);
111     }
112     outputFile.close();
113 }

```

```

115 int main() {
116     string filename = "teams.bin";
117
118     createBinaryFile(filename);
119     cout << "Початковий вміст файлу:" << endl;
120     printBinaryFile(filename);
121
122     int minPoints = 40;
123     deleteTeamsWithFewPoints(filename, minPoints);
124     cout << "\nВміст файлу після видалення команд з очками менше " << minPoints << ":" << endl;
125     printBinaryFile(filename);
126
127     vector<Team> newTeams = {
128         {"Leaders", "Donetsk", 15, 60},
129         {"Strong", "Odesa", 13, 80}
130     };
131     addTeamsToFile(filename, newTeams);
132     cout << "\nВміст файлу після додавання нових команд:" << endl;
133     printBinaryFile(filename);
134
135     return 0;
136 }

```

### Завдання №3 VNS Lab 9 – 13

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include <cctype>
5  #include <vector>
6
7  bool isConsonant(char c) {
8      c = std::tolower(c);
9      return std::isalpha(c) && c != 'a' && c != 'e' && c != 'i' && c != 'o' && c != 'u' && c != 'y';
10 }
11
12 int main() {
13     const std::string file1 = "F1.txt";
14     const std::string file2 = "F2.txt";
15     int N1, N2;
16
17     std::cout << "Введіть номер N1: ";
18     std::cin >> N1;
19     std::cout << "Введіть номер N2: ";
20     std::cin >> N2;
21
22     std::ofstream outFile(file1);
23     if (!outFile) {
24         std::cerr << "Помилка створення файлу F1!" << std::endl;
25         return 1;
26     }
27
28     outFile << "Apple\n"
29             << "Banana\n"
30             << "Apricot\n"
31             << "Orange\n"
32             << "Avocado\n";
33     outFile.close();
34
35     std::ifstream inFile(file1);
36     if (!inFile) {
37         std::cerr << "Помилка відкриття файлу F1!" << std::endl;
38         return 1;
39     }
40
41     std::ofstream outFile2(file2);
42     if (!outFile2) {
43         std::cerr << "Помилка створення файлу F2!" << std::endl;
44         return 1;
45     }
46
47     std::string line;
48     int lineNumber = 0;
49     std::vector<std::string> selectedLines;
50
51     while (std::getline(inFile, line)) {
52         ++lineNumber;
53         if (lineNumber >= N1 && lineNumber <= N2 && !line.empty() && line[0] == 'A') {
54             outFile2 << line << "\n";
55             selectedLines.push_back(line);
56         }
57     }

```

```

59     inFile.close();
60     outFile2.close();
61
62     int maxConsonants = 0;
63     int maxLineIndex = -1;
64
65     for (size_t i = 0; i < selectedLines.size(); ++i) {
66         int consonantCount = 0;
67         for (char c : selectedLines[i]) {
68             if (isConsonant(c)) {
69                 ++consonantCount;
70             }
71         }
72         if (consonantCount > maxConsonants) {
73             maxConsonants = consonantCount;
74             maxLineIndex = i + 1;
75         }
76     }
77
78     if (maxLineIndex != -1) {
79         std::cout << "Рядок з найбільшою кількістю приголосних файлі F2: " << maxLineIndex << std::endl;
80     } else {
81         std::cout << "файлі F2 немає рядків для аналізу." << std::endl;
82     }
83
84     return 0;
85 }

```

## Завдання №4 Algotester Lab 4 – 2

### 3 STL:

```

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

```

Без STL:

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  #include <iterator>
5
6  using namespace std;
7
8  void delDuplicat(vector<int>& arr, int k) {
9      sort(arr.begin(), arr.end());
10
11      vector<int> Arr;
12      for (size_t i = 0; i < arr.size(); ++i) {
13          if (find(Arr.begin(), Arr.end(), arr[i]) == Arr.end()) {
14              Arr.push_back(arr[i]);
15          }
16      }
17
18      k = k % Arr.size();
19      rotate(Arr.begin(), Arr.begin() + k, Arr.end());
20
21      cout << Arr.size() << '\n';
22      for (const int& num : Arr) {
23          cout << num << ' ';
24      }
25      cout << '\n';
26 }
27
28 int main() {
29     int n, k;
30     cin >> n >> k;
31     vector<int> arr(n);
32     for (int i = 0; i < n; ++i) {
33         cin >> arr[i];
34     }
35
36     delDuplicat(arr, k);
37     return 0;
38 }
```

**Завдання №5 Algotester Lab 6 – 2**

```

1  #include <iostream>
2  #include <vector>
3  #include <string>
4  #include <set>
5  #include <algorithm>
6
7  using namespace std;
8
9  bool isValid(int x, int y) {
10     return x >= 0 && x < 8 && y >= 0 && y < 8;
11 }
12
13 void addAttacker(set<char>& attackers, char piece) {
14     attackers.insert(piece);
15 }
16
17 string checkCell(const vector<string>& board, int x, int y) {
18     if (board[x][y] != 'O') {
19         return "X";
20     }
21
22     set<char> attackers;
23
24     for (int i = 0; i < 8; ++i) {
25         for (int j = 0; j < 8; ++j) {
26             char piece = board[i][j];
27             if (piece == 'O') continue;
28
29             switch (piece) {
30                 case 'P':
31                     if (isValid(i + 1, j - 1) && i + 1 == x && j - 1 == y) addAttacker(attackers, 'P');
32                     if (isValid(i + 1, j + 1) && i + 1 == x && j + 1 == y) addAttacker(attackers, 'P');
33                     break;
34                 case 'R':
35                     if (i == x || j == y) addAttacker(attackers, 'R');
36                     break;
37                 case 'N':
38                     if (abs(i - x) == 2 && abs(j - y) == 1) addAttacker(attackers, 'N');
39                     if (abs(i - x) == 1 && abs(j - y) == 2) addAttacker(attackers, 'N');
40                     break;
41                 case 'B':
42                     if (abs(i - x) == abs(j - y)) addAttacker(attackers, 'B');
43                     break;
44                 case 'K':
45                     if (abs(i - x) <= 1 && abs(j - y) <= 1) addAttacker(attackers, 'K');
46                     break;
47                 case 'Q':
48                     if (i == x || j == y || abs(i - x) == abs(j - y)) addAttacker(attackers, 'Q');
49                     break;
50             }
51         }
52     }
53
54     if (attackers.empty()) {
55         return "O";
56     }

```



```

54     if (attackers.empty()) {
55         return "0";
56     }
57
58     string result(attackers.begin(), attackers.end());
59     return result;
60 }
61
62 int main() {
63     vector<string> board(8);
64     for (int i = 0; i < 8; ++i) {
65         cin >> board[i];
66     }
67
68     int Q;
69     cin >> Q;
70
71     while (Q--) {
72         int x, y;
73         cin >> x >> y;
74         --x;
75         --y;
76
77         cout << checkCell(board, x, y) << endl;
78     }
79
80     return 0;
81 }

```

## Завдання №6 Class Practice Task

```

1  #include <iostream>
2  #include <fstream>
3  #include <cstring>
4
5  using namespace std;
6
7  enum FileOpResult { Success, Failure };
8
9  FileOpResult write_to_file(const char *name, const char *content) {
10     if (!name || !content) {
11         cerr << "Помилка: файл є порожнім" << endl;
12         return Failure;
13     }
14
15     ofstream file(name);
16     if (!file.is_open()) {
17         cerr << "Не вдалося відкрити або створити файл: " << name << endl;
18         return Failure;
19     }
20
21     file << content;
22
23     if (file.fail()) {
24         cerr << "Помилка запису в файл: " << name << endl;
25         return Failure;
26     }
27
28     file.close();
29     if (file.fail()) {
30         cerr << "Помилка закриття файлу: " << name << endl;
31         return Failure;
32     }
33
34     return Success;
35 }
36
37 int main() {
38     char fileName[256];
39     char content[1024];
40
41     cout << "Введіть ім'я файлу: ";
42     cin.getline(fileName, 256);
43
44     cout << "Введіть текст для запису в файл: ";
45     cin.getline(content, 1024);
46
47     FileOpResult result = write_to_file(fileName, content);
48     if (result == Success) {
49         cout << "Вміст успішно записано в файл." << endl;
50     } else {
51         cerr << "Не вдалося записати вміст в файл." << endl;
52     }
53
54     return 0;
55 }

```

```

1  #include <iostream>
2  #include <fstream>
3
4  using namespace std;
5
6  enum FileOpResult { Success, Failure };
7
8  FileOpResult copy_file(const char *file_from, const char *file_to) {
9      if (!file_from || !file_to) {
10         cerr << "Помилка: вихідне або цільове ім'я файлу є порожнім." << endl;
11         return Failure;
12     }
13
14     ifstream src(file_from, ios::binary);
15     if (!src.is_open()) {
16         cerr << "Не вдалося відкрити вихідний файл: " << file_from << endl;
17         return Failure;
18     }
19
20     ofstream dest(file_to, ios::binary);
21     if (!dest.is_open()) {
22         cerr << "Не вдалося відкрити або створити цільовий файл: " << file_to << endl;
23         src.close();
24         return Failure;
25     }
26
27     dest << src.rdbuf(); // Ефективне копіювання всього вмісту
28
29     if (dest.fail() || src.fail()) {
30         cerr << "Сталася помилка під час копіювання файлу." << endl;
31         src.close();
32         dest.close();
33         return Failure;
34     }
35
36     src.close();
37     dest.close();
38
39     if (src.fail() || dest.fail()) {
40         cerr << "Сталася помилка під час закриття файлів." << endl;
41         return Failure;
42     }
43
44     return Success;
45 }

```

```

47 int main() {
48     char sourceFile[256];
49     char destinationFile[256];
50
51     cout << "Введіть ім'я вихідного файлу: ";
52     cin.getline(sourceFile, 256);
53
54     cout << "Введіть ім'я цільового файлу: ";
55     cin.getline(destinationFile, 256);
56
57     FileOpResult result = copy_file(sourceFile, destinationFile);
58     if (result == Success) {
59         cout << "Файл успішно скопійовано." << endl;
60     } else {
61         cerr << "Не вдалося скопіювати файл." << endl;
62     }
63
64     return 0;
65 }

```

## Завдання №7 Self Practice Works

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  using namespace std;
6
7  int main() {
8      string fileName = "Test.txt";
9      string userInput;
10
11     ofstream outFile(fileName , ios::app);
12     if (!outFile) {
13         cerr << "Не вдалося відкрити файл для запису!" << endl;
14         return 1;
15     }
16
17     cout << "Введіть текст для запису у файл (для завершення введіть exit):" << endl;
18
19     while (true) {
20         getline(cin, userInput);
21         if (userInput == "exit") {
22             break;
23         }
24         outFile << userInput << endl;
25     }
26
27     outFile.close();
28
29     cout << "Ви хочете зчитати вміст файлу? (Y/N): ";
30     string answer;
31     getline(cin, answer);
32
33     if (answer == "Y") {
34         ifstream inFile(fileName);
35         if (!inFile) {
36             cerr << "Не вдалося відкрити файл для читання!" << endl;
37             return 1;
38         }
39
40         cout << "======" << endl;
41         cout << "Вміст файлу:" << endl;
42         string line;
43         while (getline(inFile, line)) {
44             cout << line << endl;
45         }
46         inFile.close();
47     } else {
48         cout << "Читання файлу скасовано." << endl;
49     }
50     return 0;
51 }

```

## Результати виконання завдань, тестування та фактично затрачений час

### Завдання №1 VNS Lab 6 – 13

Введіть рядок: 12345543216223111  
 Результат: 6554433322211111.

Фактично затрачений час: 1 год

## Завдання №2 VNS Lab 8 – 13

```
Файл створено успішно.  
Початковий вміст файлу:  
Назва: Ukraine, Місто: Lviv, Гравці: 11, Очки: 50  
Назва: Sport, Місто: kyiv, Гравці: 12, Очки: 30  
Назва: Гум, Місто: Mykolaiv, Гравці: 10, Очки: 20  
Назва: Power, Місто: Kharkiv, Гравці: 9, Очки: 70  
  
Вміст файлу після видалення команд з очками менше 40:  
Назва: Ukraine, Місто: Lviv, Гравці: 11, Очки: 50  
Назва: Power, Місто: Kharkiv, Гравці: 9, Очки: 70  
  
Вміст файлу після додавання нових команд:  
Назва: Leaders, Місто: Donetsk, Гравці: 15, Очки: 60  
Назва: Strong, Місто: Odesa, Гравці: 13, Очки: 80  
Назва: Ukraine, Місто: Lviv, Гравці: 11, Очки: 50  
Назва: Power, Місто: Kharkiv, Гравці: 9, Очки: 70
```

Фактично затрачений час: 2,5 год

## Завдання №3 VNS Lab 9 – 13

```
Введіть номер N1: 1  
Введіть номер N2: 4  
Рядок з найбільшою кількістю пригосних у файлі F2: 2
```

Фактично затрачений час: 1 год

## Завдання №4 Algotester Lab 4 – 2

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

Фактично затрачений час: 2 год

## Завдання №5 Algotester Lab 6 – 2

```
K0000000  
00000000  
00000000  
00000000  
00000000  
00000000  
00000000  
00000000  
00000000  
5  
1 1  
X  
1 2  
K  
2 1  
K  
2 2  
K  
3 1  
0
```

Фактично затрачений час: 3,5 год

## Завдання №6 Class Practice Task

Practice task 1

```
Введіть ім'я файлу: Test
Введіть текст для запису у файл: Hello World!
Вміст успішно записано у файл.
```

## Practice task 2

```
Введіть ім'я вихідного файлу: Test
Введіть ім'я цільового файлу: TestCopi
Файл успішно скопійовано.
```

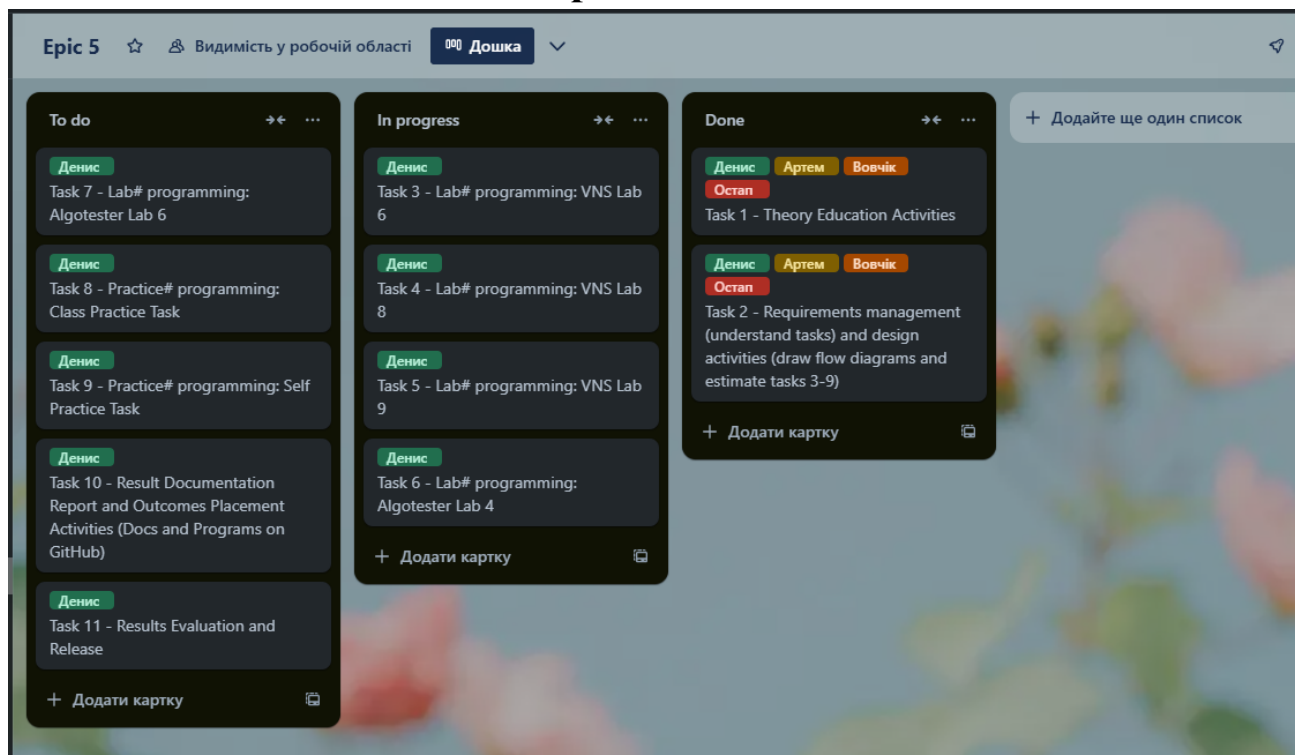
Фактично затрачений час: 3 год

## Завдання №7 Self Practice Work

```
Введіть текст для запису у файл (для завершення введіть exit):
Hello my friend!
I love my University!
exit
Ви хочете зчитати вміст файлу? (Y/N): Y
=====
Вміст файлу:
Hello my friend!
I love my University!
```

Фактично затрачений час: 1,5 год

## Кооперація з командою:



**Висновок:** Виконуючи 5 епік, я ознайомився з основними принципами роботи з файлами в C++. Вивчив роботу з текстовими та бінарними файлами, включаючи операції відкриття, читання, запису й закриття. Навчився перевіряти стан файлу для обробки помилок і забезпечення надійності. Особливу увагу приділив базовим операціям із символами та рядками: конкатенації, порівнянню й пошуку. Відпрацював форматування текстових файлів при записі даних та методи обробки рядків із файлів. Розібрався з ефективністю та використанням бінарних файлів у програмах. Окрім цього, вивчив стандартні бібліотеки для

роботи з файлами (ifstream, ofstream, fstream). Також брав участь у створенні дошки Trello та обговоренні задач із командою.