

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:

Практичних Робіт до блоку № 5

**Виконала:**

Студентка групи ІІІ-12  
Іванів Христина Вікторівна

**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:** ознайомитись з файлами, бінарними файлами, символами і рядковими змінними, текстовими файлами, стандартними бібліотеками, створенням і використанням бібліотек в мовах C та C++.

## **Теоретичні відомості:**

Теоретичні відомості з переліком важливих тем:

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

### 1) Індивідуальний план опрацювання теорії:

- Тема №1: Вступ до Роботи з Файлами
  - Джерела Інформації:
    - Лекції О. Пшеничного
    - Практичні М. Фаріон
  - Що опрацьовано:
    - Лекції О. Пшеничного
    - Практичні М. Фаріон
  - Статус: ознайомлена з файлами та виконанням елементарних дій над ними
- Тема №2: Символи і Рядкові Змінні
  - Джерела Інформації:
    - Лекції О. Пшеничного
    - Практичні М. Фаріон
    - Уроки 36, 37 з курсу C++ теорія з каналу «Блоган»
    - Інформація з chatgpt
  - Що опрацьовано:
    - Лекції О. Пшеничного
    - Практичні М. Фаріон
    - Інформація з chatgpt
  - Статус: ознайомлена з символами та рядковими змінними, вмію виконувати операції над ними
- Тема №3: Текстові Файли
  - Джерела Інформації:
    - Лекції О. Пшеничного
    - Практичні М. Фаріон
    - Урок 166 з курсу C++ теорія з каналу «Блоган»
  - Що опрацьовано:

- Лекції О. Пшеничного
  - Практичні М. Фаріон
  - Урок 166 з курсу C++ теорія з каналу «Блоган»
- Статус: ознайомлена з текстовими файлами у C та C++
- Тема №4: Бінарні Файли
  - Джерела Інформації:
    - Лекції О. Пшеничного
    - Практичні М. Фаріон
    - Урок 166 з курсу C++ теорія з каналу «Блоган»
  - Що опрацьовано:
    - Лекції О. Пшеничного
    - Практичні М. Фаріон
    - Урок 166 з курсу C++ теорія з каналу «Блоган»
  - Статус: ознайомлена з бінарними файлами у C та C++
- Тема №5: Стандартна бібліотека та робота з файлами
  - Джерела Інформації:
    - Лекції О. Пшеничного
    - Практичні М. Фаріон
  - Що опрацьовано:
    - Лекції О. Пшеничного
    - Практичні М. Фаріон
  - Статус: навчилась використовувати стандартні бібліотеки та працювати з файлами
- Тема №6: Створення й використання бібліотек
  - Джерела Інформації:
    - Лекції О. Пшеничного
    - Практичні М. Фаріон
  - Що опрацьовано:
    - Лекції О. Пшеничного
    - Практичні М. Фаріон
  - Статус: ознайомлена з бібліотеками

## Виконання роботи:

### *1. Опрацювання завдання та вимог до програм та середовища:*

#### Завдання №1 VNS. Лабораторна робота №6

- **Варіант завдання: 6**
- *Деталі завдання:*
- 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.

- *Важливі деталі для врахування:*  
Перетворити рядок так, щоб всі букви в ньому були відсортовані за зростанням.

## **Завдання №2 VNS. Лабораторна робота №8**

- **Варіант завдання: 6**

- *Деталі завдання:*

*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.*

- *Важливі деталі для врахування:*

*Структура "Школяр":*

- *прізвище, ім'я, по батькові;*
- *клас;*
- *номер телефону;*
- *оцінки по предметах (математика, фізика, українська мова, література).*

*Знищити всі елементи, у яких є 2 хоча б з одного предмету, додати елемент у початок файлу.*

## **Завдання №3 VNS. Лабораторна робота №9**

- **Варіант завдання: 6**

- *Деталі завдання:* Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію

*Виконати завдання..*

- *Важливі деталі для врахування:*

- 1) Скопіювати з файлу F1 у файл F2 рядки, починаючи з N до K.
- 2) Підрахувати кількість приголосних букв у файлі F2.

## **Завдання №4.1 VNS. Algotester task 4. V- 2**

▪ **Варіант завдання: 2**

• *Деталі завдання:*

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.

• *Важливі деталі для врахування:*

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

$1 \leq N, K \leq 1000$

$0 \leq a_i \leq 100$

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

**Завдання №4.2 VNS. Algotester task 4. V- 3**

▪ **Варіант завдання: 3**

• *Деталі завдання:*

Вам дано масив, який складається з  $N$  додатніх 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.

• *Важливі деталі для врахування:*

Вхідні дані

У першому рядку  $N$  - кількість чисел.

У другому рядку  $N$  чисел  $a_i$  - елементи масиву.

Вихідні дані

У першому рядку  $M$  - кількість чисел у масиву

У другому рядку  $M$  посортованих за умовою чисел.

Обмеження

$$1 \leq N \leq 103$$

$$0 \leq a_i \leq 103$$

Пам'ятайте, ви маєте написати 2 варіанти розв'язку, один з використанням засобів STL (власноруч написаний компаратор або `std::partition + std::sort + std::unique`), інший зі своєю реалізацією. Алгоритм сортування можна вибрати будь який, окрім сортування бульбашкою і має працювати за  $N \cdot \log N$  часу.

## Завдання №5. Algotester task 6. V- 2

- *Деталі завдання:*

У вас є шахова дошка розміром  $8 \times 8$  та дуже багато фігур.

Кожна клітинка може мати таке значення:

- Пуста клітинка OO
- Пішак PP
- Тура RR
- Кінь NN
- Слон BB
- Король KK
- Королева QQ

Вам дають позиції фігур на дошці (всі фігури одного кольору, кількість королів може бути  $> 1$ ).

Далі йдуть QQ запитів з координатами клітинки  $\{x, y\}$ . На кожен запит ви маєте вивести стрічку зі - посортовані за алфавітом букви фігур, які атакують цю клітинку (пішаки атакують вниз).

У випадку, якщо на клітинці стоїть якась фігура - виведіть символ XX.

У випадку, якщо клітинку не атакують - виведіть OO.

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.

- *Важливі деталі для врахування:*

## Input

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

У наступному рядку ціле число QQ - кількість записів

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

## Output

QQ разів відповідь у наступному форматі:

Строка `resultresult` - усі фігури, які атакують клітинку з запиту.

## Constraints

$$|row_i| = N$$

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

$$1 \leq Q \leq 64$$

$$1 \leq x, y \leq 8$$

## Завдання №6 Class Practice 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 №7 Self Practice Algotester Task**

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 ютубі - він вирішив придумати свою гру на основі sudoku.

Гра виглядає так:

Є поле розміром  $N \times N$ , в якому частина клітинок заповнена цифрами, а частина клітинок пусті (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 в посортовані по зростанню!). В клітинку можна вписати лише ті числа, які не зустрічаються в рядку та стовбці, які перетинаються у цій клітинці.

Під час гри поле не міняється!

Також необов'язково, щоб це було валідне sudoku! Якщо є клітинка, в яку не можна вписати ніяку цифру - виведіть 0.

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

## 2. Дизайн та планована оцінка часу виконання завдань:

### Завдання №1 VNS. Лабораторна робота №6

Плановий час на реалізацію: 1.5 години

### Завдання №2 VNS. Лабораторна робота №5

Плановий час на реалізацію: 3 години

### Завдання №3 VNS. Лабораторна робота №9

Плановий час на реалізацію: 2 години

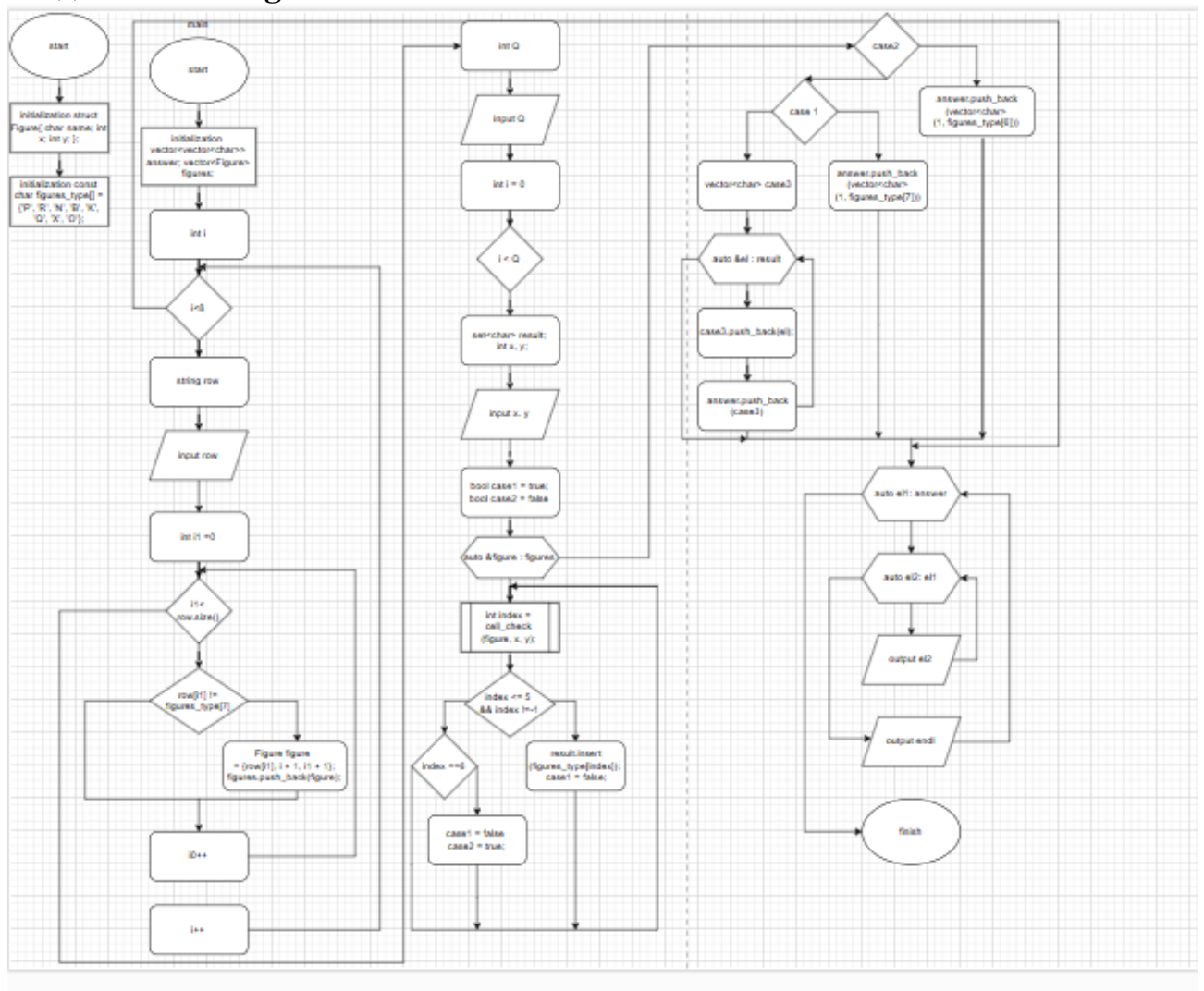
### Завдання №4.1 VNS. Algotester task 4. V- 2

Плановий час на реалізацію: 2 години

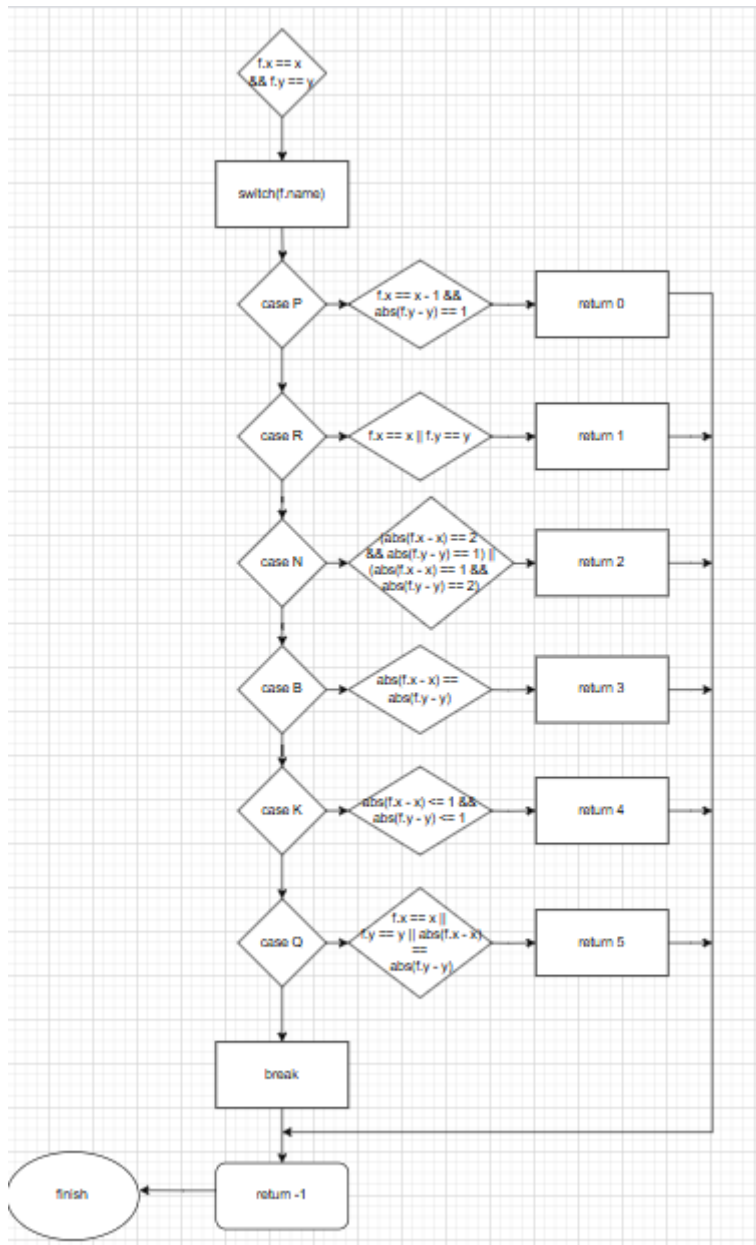
### Завдання №4.2 VNS. Algotester task 4. V- 3

Плановий час на реалізацію: 2 години

### Завдання №5. Algotester task 6. V- 2







Плановий час на реалізацію: 4 години

### Завдання №6 Class Practice Work

Плановий час на реалізацію: 1 година

### Завдання №7 Self Practice Algotester Task

Плановий час на реалізацію: 1 година

## 3. Код програми та фактичний час на реалізацію:

### Завдання №1 VNS. Лабораторна робота №6

```

1  #include <iostream>
2  #include <cstring>
3  #include <algorithm>
4
5  using namespace std;
6
7  void string_sort(char *string)
8  {
9
10     char *word = strtok(string, " .");
11     while (word != nullptr)
12     {
13         sort(word, word + strlen(word));
14         cout << word << " ";
15         word = strtok(nullptr, " .");
16     }
17
18     cout << "." << endl;
19 }
20
21 int main()
22 {
23     const int max_lenght = 255;
24     char string[max_lenght + 1];
25
26     gets(string);
27     string_sort(string);
28     return 0;
29 }

```

Фактичний час на реалізацію: 1 година

**Завдання №2 VNS.** Лабораторна робота №8

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include <vector>
5
6  using namespace std;
7
8  struct Pupil
9  {
10     char last_name[30];
11     char first_name[30];
12     char patronymic[30];
13     int grade;
14     char phone[15];
15     int math;
16     int physics;
17     int ukrainian;
18     int literature;
19
20     bool two_low_gades() const
21     {
22         int low_count = 0;
23         if (math < 3)
24             low_count++;
25         if (physics < 3)
26             low_count++;
27         if (ukrainian < 3)
28             low_count++;
29         if (literature < 3)
30             low_count++;
31         return low_count >= 2;
32     }
33 };
34
35 // Функція для створення та запису даних у файл
36 void create_file(const string &filename, const vector<Pupil> &pupils)
37 {
38     ofstream file(filename, ios::binary);
39     if (!file)
40     {
41         cout << "Error: Cannot open file for writing.\n";
42         return;
43     }
44     for (const auto &pupil : pupils)
45     {
46         file.write(reinterpret_cast<const char *>(&pupil), sizeof(Pupil));
47     }
48     file.close();
49 }
50
51 // Функція для виведення вмісту файлу
52 void print_file(const string &filename)
53 {
54     ifstream file(filename, ios::binary);
55     if (!file)
56     {
57         cout << "Error: Cannot open file for reading.\n";
58         return;

```

```

58     }
59 }
60 Pupil pupil;
61 while (file.read(reinterpret_cast<char *>(&pupil), sizeof(Pupil)))
62 {
63     cout << "Last name: " << pupil.last_name << ", First name: " << pupil.first_name
64         << ", Patronomic: " << pupil.patronymic << ", Phone: " << pupil.phone
65         << "Math: " << pupil.math << ", Physics: " << pupil.physics
66         << ", Ukrainian: " << pupil.ukrainian << ", Literature: " << pupil.literature << "\n";
67 }
68 file.close();
69 }
70
71 // Функція для видалення елементів, де є хоча б 2 оцінки менші за 3
72 void delete_low_grades(const string &fileName)
73 {
74     ifstream in_file(fileName, ios::binary);
75     if (!in_file)
76     {
77         cerr << "Помилка при відкритті файлу для читання!" << endl;
78         return;
79     }
80
81     vector<Pupil> pupils;
82     Pupil pupil;
83     while (in_file.read(reinterpret_cast<char *>(&pupil), sizeof(Pupil)))
84     {
85         if (!pupil.two_low_gades())
86         {
87             pupils.push_back(pupil);
88         }
89     }
90     in_file.close();
91
92     // Перезаписуємо файл з оновленими даними
93     ofstream out_file(fileName, ios::binary | ios::trunc);
94     if (!out_file)
95     {
96         cerr << "Помилка при відкритті файлу для запису!" << endl;
97         return;
98     }
99
100     for (const auto &pupil : pupils)
101     {
102         out_file.write(reinterpret_cast<const char *>(&pupil), sizeof(Pupil));
103     }
104
105     out_file.close();
106 }
107
108 // Функція для додавання елементів у початок файлу
109 void add_to_file_at_start(const string& fileName, const Pupil& new_pupil) {
110     // Читання всіх даних з файлу
111     ifstream in_file(fileName, ios::binary);
112     if (!in_file) {

```

```

111 ifstream in_file(fileName, ios::binary);
112 if (!in_file) {
113     cerr << "Помилка при відкритті файлу для читання!" << endl;
114     return;
115 }
116
117 vector<Pupil> pupils;
118 Pupil pupil;
119 while (in_file.read(reinterpret_cast<char*>(&pupil), sizeof(Pupil))) {
120     pupils.push_back(pupil);
121 }
122 in_file.close();
123
124 // Запис нового елемента на початок файлу
125 ofstream out_file(fileName, ios::binary | ios::trunc);
126 if (!out_file) {
127     cerr << "Помилка при відкритті файлу для запису!" << endl;
128     return;
129 }
130
131 // Записуємо новий елемент на початок
132 out_file.write(reinterpret_cast<const char*>(&new_pupil), sizeof(Pupil));
133
134 // Порім записуємо всі інші елементи
135 for (const auto& pupil : pupils) {
136     out_file.write(reinterpret_cast<const char*>(&pupil), sizeof(Pupil));
137 }
138
139 out_file.close();
140 }
141 int main() {
142
143     string fileName = "pupils.data";
144
145     vector<Pupil> pupils = {
146         {"Ivaniv", "Khrystyna", "Victoriivna", 11, "123-456", 5, 5, 5, 5},
147         {"Petrovy", "Petro", "Petrovych", 9, "987-654", 4, 5, 3, 4},
148         {"Ivanovy", "Ivan", "Ivanovych", 11, "555-123", 2, 2, 4, 4}
149     };
150
151     create_file(fileName, pupils);
152
153     cout << "Відкрі файлу до зміни:" << endl;
154     print_file(fileName);
155
156     delete_low_grades(fileName);
157     cout << "Відкрі файлу після видалення учнів з низькими оцінками:" << endl;
158     print_file(fileName);
159
160     Pupil new_pupil = {"Black", "John", "Stepanovych", 8, "111-222", 5, 5, 5, 5};
161     add_to_file_at_start(fileName, new_pupil);
162     cout << "Відкрі файлу після додавання нового елемента на початок:" << endl;
163     print_file(fileName);
164
165     return 0;
166 }

```

Фактичний час на реалізацію: 3 години

## Завдання №3 Лабораторна робота №9

```

1 #include <iostream>
2 #include <fstream>
3 #include <string>
4 #include <cctype>
5
6 using namespace std;
7
8 bool is_consonant(char c) {
9     c = tolower(c);
10    return (c >= 'a' && c <= 'z') && (c != 'a' && c != 'e' && c != 'i' && c != 'o' && c != 'u' && c != 'y');
11 }
12
13 int count_consonants(const string &line) {
14     int count = 0;
15     for (char c : line) {
16         if (is_consonant(c)) {
17             count++;
18         }
19     }
20     return count;
21 }
22
23 int main() {
24
25     ofstream file_F1("F1.txt");
26     if (!file_F1) {
27         cout << "Error: Cannot create file F1.txt\n";
28         return 1;
29     }
30
31     file_F1 << "All clouds have souls.\n";
32     file_F1 << "We just sat there and said nothing.\n";
33     file_F1 << "I hope this turns into something great.\n";
34     file_F1 << "We chose a table near the fountain and talked all evening.\n";
35     file_F1 << "It's late at night and everyone is asleep.\n";
36     file_F1 << "The feelings of strangeness are deeper and stronger today.\n";
37     file_F1 << "I often wonder why artists marry so much.\n";
38     file_F1 << "The sun is really angry this afternoon.\n";
39     file_F1 << "Nothing is stranger than a good day.\n";
40     file_F1 << "It felt wrong to read.\n";
41
42     file_F1.close();
43
44     int N, K;
45     cout << "Enter N: ";
46     cin >> N;
47     cout << "Enter K: ";
48     cin >> K;

```

```

49
50     if (N < 1 || K < N || K > 10) {
51         cout << "Invalid range of lines.\n";
52         return 1;
53     }
54
55     ifstream file_F1_read("F1.txt");
56     ofstream file_F2("F2.txt");
57
58     if (!file_F1_read || !file_F2) {
59         cout << "Error: Cannot open files.\n";
60         return 1;
61     }
62
63     string line;
64     int current_line = 1;
65
66     while (getline(file_F1_read, line)) {
67         if (current_line >= N && current_line <= K) {
68             file_F2 << line << "\n";
69         }
70         current_line++;
71     }
72
73     file_F1_read.close();
74     file_F2.close();
75
76     ifstream file_F2_read("F2.txt");
77
78     if (!file_F2_read) {
79         cout << "Error: Cannot open file F2.txt for reading.\n";
80         return 1;
81     }
82
83     int consonant_count = 0;
84     while (getline(file_F2_read, line)) {
85         consonant_count += count_consonants(line);
86     }
87
88     file_F2_read.close();
89
90     cout << "The amount of consonant letters in file F2: " << consonant_count << endl;
91
92     return 0;
93 }
94

```

Фактичний час на реалізацію: 2 години

## Завдання №4.1 VNS. Algotester task 4. V- 2

```

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

```

Фактичний час на реалізацію: 2 години

#### Завдання №4.2 VNS. Algotester task 4. V- 3

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  void show_array(const vector<int>& array){
8      cout << array.size() << endl;
9      for (const int& num : array) {
10         cout << num << " ";
11     }
12     cout << endl;
13 }
14
15 void insert_sorted(vector<int>& array, int value, int& p1, int& p2, int remainder ) {
16     bool is_copy = false;
17
18     if (remainder == 0) {
19         if (p1 == 0 || array[p1 - 1] < value) {
20             array.insert(array.begin() + p1, value);
21         } else {
22             int l = 0, r = p1 - 1, n = r / 2;
23             while (l < r) {
24                 int& val = array[n];
25                 if (val < value) {
26                     l = n + 1;
27                 } else if (val > value) {
28                     r = n;
29                 } else {
30                     is_copy = true;
31                     break;
32                 }
33                 n = (r + l) / 2;
34             }
35             if (l >= r) {
36                 if (array[l] != value) array.insert(array.begin() + l, value);
37                 else is_copy = true;
38             }
39         }
40         if (!is_copy) {
41             p1++;
42             p2++;
43         }
44     }
45 }
46
```

```

47     else if (remainder == 1) {
48         if (p1 == p2 || p2 == 0 || array[p2 - 1] > value) {
49             array.insert(array.begin() + p2, value);
50         } else {
51             int l = p1, r = p2 - 1, n = (r + 1) / 2;
52             while (l < r) {
53                 int& val = array[n];
54                 if (val > value) {
55                     l = n + 1;
56                 } else if (val < value) {
57                     r = n;
58                 } else {
59                     is_copy = true;
60                     break;
61                 }
62                 n = (r + 1) / 2;
63             }
64             if (l >= r) {
65                 if (array[l] != value) array.insert(array.begin() + l, value);
66                 else is_copy = true;
67             }
68         }
69         if (!is_copy) {
70             p2++;
71         }
72     }
73
74     else {
75         const int SIZE = array.size();
76         if (p2 == SIZE || SIZE == 0 || array[SIZE - 1] < value) {
77             array.push_back(value);
78         } else {
79             int l = p2, r = SIZE - 1, n = (r + 1) / 2;
80             while (l < r) {
81                 int& val = array[n];
82                 if (val < value) {
83                     l = n + 1;
84                 } else if (val > value) {
85                     r = n;
86                 } else {
87                     is_copy = true;
88                     break;
89                 }
90                 n = (r + 1) / 2;
91             }
92             if (l >= r) {
93                 if (array[l] != value) array.insert(array.begin() + l, value);
94                 else is_copy = true;
95             }
96         }
97     }
98 }
99
100 int main() {
101     int size;
102     cin >> size;
103     vector<int> array;
104     array.reserve(10 * 10 * 10);
105
106     int p1 = 0, p2 = 0;
107     int temp;
108
109     for (int i = 0; i < size; i++) {
110         cin >> temp;
111         insert_sorted(array, temp, p1, p2, temp % 3);
112     }
113
114     show_array(array);
115
116     return 0;
117 }

```

Фактичний час на реалізацію: 2 години

## Завдання №5. Algotester task 6. V- 2

```
1  ✓ #include <iostream>
2  #include <vector>
3  #include <cmath>
4  #include <set>
5
6  using namespace std;
7
8  ✓ struct Figure
9  {
10     char name;
11     int x;
12     int y;
13 };
14
15 const char figures_type[] = {'P', 'R', 'N', 'B', 'K', 'Q', 'X', 'O'};
16
17 ✓ int cell_check(Figure &f, int x, int y)
18 {
19     if (f.x == x && f.y == y)
20         return 6;
21     ✓ switch (f.name)
22     {
23     case 'P':
24         if (f.x == x - 1 && abs(f.y - y) == 1)
25             return 0;
26         break;
27     case 'R':
28         if (f.x == x || f.y == y)
29             return 1;
30         break;
31     case 'N':
32         if ((abs(f.x - x) == 2 && abs(f.y - y) == 1) || (abs(f.x - x) == 1 && abs(f.y - y) == 2))
33             return 2;
34         break;
35     case 'B':
36         if (abs(f.x - x) == abs(f.y - y))
37             return 3;
38         break;
39     case 'K':
40         if (abs(f.x - x) <= 1 && abs(f.y - y) <= 1)
41             return 4;
42         break;
43     case 'Q':
44         if (f.x == x || f.y == y || abs(f.x - x) == abs(f.y - y))
45             return 5;
46         break;
47     default:
48         break;
```



```

48         break;
49     }
50     return -1;
51 }
52
53 int main(){
54     vector<vector<char>> answer;
55     vector<Figure> figures;
56     for (int i = 0; i < 8; i++)
57     {
58         string row;
59         cin >> row;
60         for (int i1 = 0; i1 < row.size(); i1++)
61         {
62             if (row[i1] != figures_type[7])
63             {
64                 Figure figure = {row[i1], i + 1, i1 + 1};
65                 figures.push_back(figure);
66             }
67         }
68     }
69
70     int Q;
71     cin >> Q;
72     for (int i = 0; i < Q; i++)
73     {
74         set<char> result;
75         int x, y;
76         cin >> x >> y;
77         bool case1 = true;
78         bool case2 = false;
79
80         for (auto &figure : figures)
81         {
82             int index = cell_check(figure, x, y);
83             if (index <= 5 && index != -1)
84             {
85                 result.insert(figures_type[index]);
86                 case1 = false;
87             } else if (index == 6){
88                 case1 = false;
89                 case2 = true;
90             }
91         }
92     }
93     if (case2)
94     {
95         answer.push_back(vector<char>(1, figures_type[6]));
96     }
97     else if (case1)
98     {
99         answer.push_back(vector<char>(1, figures_type[7]));
100     }
101     else
102     {
103         vector<char> case3;
104         for (auto &el : result)
105         {
106             case3.push_back(el);
107         }
108         answer.push_back(case3);
109     }
110 }
111
112 for (auto e11: answer)
113 {
114     for(auto e12: e11){
115         cout << e12;
116     }
117     cout << endl;
118 }
119
120
121 return 0;
122 }

```

Фактичний час на реалізацію: 3 години

## Завдання №6.1 Class Practice Work

```
1  #include <iostream>
2  #include <fstream>
3
4  using namespace std;
5
6  enum FileOpResult { Success, Failure };
7
8  FileOpResult write_to_file(const char *name, const char *content)
9  {
10     if (name == nullptr || content == nullptr)
11     {
12         cout << "Error: Null file name or content.\n";
13         return Failure;
14     }
15
16     ofstream file(name);
17     if (!file.is_open())
18     {
19         cout << "Error: Cannot open file \"" << name << "\" for writing.\n";
20         return Failure;
21     }
22
23     file << content;
24
25     file.close();
26     if (file.fail())
27     {
28         cout << "Error: File closing failed.\n";
29         return Failure;
30     }
31     return Success;
32 }
33
34 int main()
35 {
36     char file_name_write[256];
37     char content[1024];
38
39     cout << "Enter file name: ";
40     cin.getline(file_name_write, 256);
41
42     cout << "Enter content to write: ";
43     cin.getline(content, 1024);
44
45     FileOpResult result = write_to_file(file_name_write, content);
46     if (result == Success)
47     {
48         cout << "File written successfully.\n";
49     }
50     else
51     {
52         cout << "Failed to write to file.\n";
53     }
54
55     return 0;
56 }
```

Фактичний час на реалізацію: 2 години

## Завдання №6.1 Class Practice Work

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  using namespace std;
6
7  enum FileOpResult { Success, Failure};
8
9  FileOpResult copy_file(const char *file_from,const char *file_to){
10     ifstream file1(file_from, ifstream::in);
11     if(!file1)
12         return Failure;
13     fstream file2(file_to, fstream::out);
14     if(!file2)
15         return Failure;
16     string line;
17     while(getline(file1, line)){
18         line += "\n";
19         file2.write(line.c_str(), line.size());
20     }
21     file1.close();
22     file2.close();
23     return Success;
24 }
25
26 int main(){
27     string file1, file2;
28     cout << "Enter name of the first file: ";
29     getline(cin, file1);
30     cout << "Enter name of the second file: ";
31     getline(cin, file2);
32     file1 += ".txt";
33     file2 += ".txt";
34     FileOpResult result = copy_file(file1.c_str(), file2.c_str());
35     if(result == Success)
36         cout << "Content was successfully copied!";
37     else
38         cout << "Failed to copy files(";
39     return 0;
40 }
41

```

Фактичний час на реалізацію: 2 години

## Завдання №7 Self Practice Algotester Task

```

1  #include <iostream>
2  #include <vector>
3  #include <set>
4
5  using namespace std;
6
7  vector<int> find_numbers(const vector<string> &sudoku, const int &x, const int &y)
8  {
9      vector<int> result;
10     if (sudoku[x - 1][y - 1] != '0')
11     {
12         result.push_back(sudoku[x - 1][y - 1] - '0');
13         return result;
14     }
15     set<int> numbers;
16     int N = sudoku.size();
17     for (size_t i = 0; i < N; i++)
18     {
19         numbers.insert(sudoku[x - 1][i] - '0');
20         numbers.insert(sudoku[i][y - 1] - '0');
21     }
22     for (size_t i = 1; i < N + 1; i++)
23     {
24         if (numbers.count(i) == 0)
25         {
26             result.push_back(i);
27         }
28     }
29     return result;
30 }
31
32 int main()
33 {
34     int N;
35     cin >> N;
36     vector<string> sudoku(N);
37     for (auto &el : sudoku)
38     {
39         cin >> el;
40     }
41
42     int Q, x, y;
43     cin >> Q;
44     for (size_t i = 0; i < Q; i++)
45     {
46         cin >> x >> y;
47         vector<int> numbers = find_numbers(sudoku, x, y);
48         cout << numbers.size() << endl;
49         if (numbers.size() != 0)
50         {
51             for (auto &el : numbers)
52             {
53                 cout << el << " ";
54             }
55             cout << endl;
56         }
57     }
58     return 0;

```

Фактичний час на реалізацію: 2 години

#### 4. Результати виконання завдань, тестування:

Завдання №1 VNS. Лабораторна робота №6

```

abc cdb bda hda
abc bcd abd adh

```

## Завдання №2 VNS. Лабораторна робота №5

```
Вміст файлу до змін:
Last name: Ivaniv, First name: Khrystyna, Patronomic: Victorivna, Phone: 123-456Math: 5, Physics: 5, Ukrainian: 5, Literature: 5
Last name: Petrovy, First name: Petro, Patronomic: Petrovych, Phone: 987-654Math: 4, Physics: 5, Ukrainian: 3, Literature: 4
Last name: Ivanovy, First name: Ivan, Patronomic: Ivanovych, Phone: 555-123Math: 2, Physics: 2, Ukrainian: 4, Literature: 4
Вміст файлу після видалення учнів з низькими оцінками:
Last name: Ivaniv, First name: Khrystyna, Patronomic: Victorivna, Phone: 123-456Math: 5, Physics: 5, Ukrainian: 5, Literature: 5
Last name: Petrovy, First name: Petro, Patronomic: Petrovych, Phone: 987-654Math: 4, Physics: 5, Ukrainian: 3, Literature: 4
Вміст файлу після додавання нового елемента на початок:
Last name: Black, First name: John, Patronomic: Stepanovych, Phone: 111-222Math: 5, Physics: 5, Ukrainian: 5, Literature: 5
Last name: Ivaniv, First name: Khrystyna, Patronomic: Victorivna, Phone: 123-456Math: 5, Physics: 5, Ukrainian: 5, Literature: 5
Last name: Petrovy, First name: Petro, Patronomic: Petrovych, Phone: 987-654Math: 4, Physics: 5, Ukrainian: 3, Literature: 4
```

## Завдання №3 VNS. Лабораторна робота №9

```
Enter N: 1
Enter K: 1
The amount of consonant letters in file F2: 11
```

## Завдання №4.1 VNS. Algotester task 4. V- 2

```
7
10
2 2 3 2 9 6 6
4
6 9 2 3
```

## Завдання №4.2 VNS. Algotester task 4. V- 3

```
12
1 5 9 7 4 5 3 7 4 2 1 4
7
3 9 7 4 1 2 5
```

## Завдання №5. Algotester task 6. V- 2

```
00000000
0R000000
00N00000
0000P000
00000000
00000000
K0Q00000
0000000R

7
8 1
1 2
5 4
5 1
6 2
8 4
6 7
KR
NR
NP
Q
KQR
QR
O
```

## Завдання №6.1 Class Practice Work

```
Enter name of the first file: 1q
Enter name of the second file: 2q
Content was successfully copied!
```

## Завдання №6.2 Class Practice Work

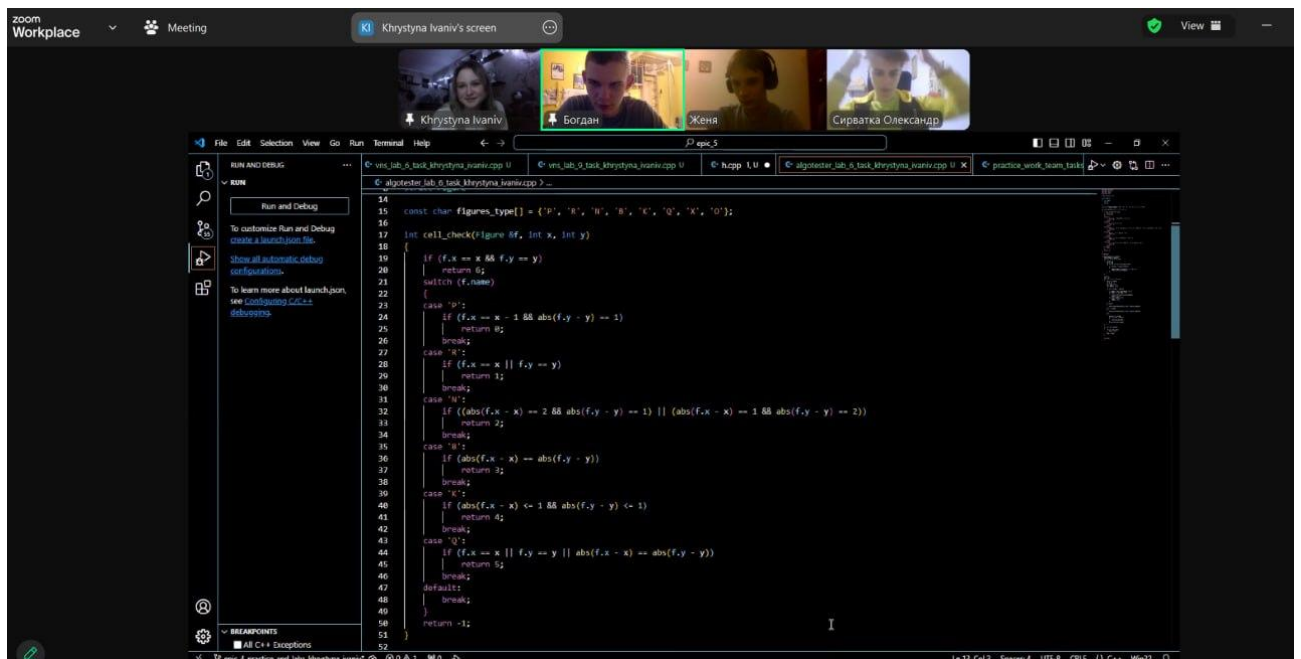
```
Enter file name: q
Enter content to write: 1234nfjbfjhfoejdmlhnfauwe
File written successfully.
```

## Завдання №7 Self Practice Algotester Task

```
9
123456789
000000000
010000000
030000000
040000000
050000000
060000000
070000000
987643215
3
1 1
1
1
2 2
1
9
8 8
6
2 3 4 5 6 9
```

## 5 . Кооперація з командою:

*Провели зустріч у зумі, обговорили деталі виконання завдань*



**Висновок:** Під час виконання епіку я ознайомилась з файлами, бінарними файлами, символами і рядковими змінними, текстовими файлами, стандартними бібліотеками, створенням і використанням бібліотек в мовах C та C++.