

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

**Виконала:**

Студентка групи ІІІ-11

Андрусишин Соломія Володимирівна

**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:**

### **1. Вступ до Роботи з Файлами:**

- 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

### **2. Символи і Рядкові Змінні:**

- Робота з `char` та `string`: основні операції і методи
- Стрічкові літерали та екранування символів

- Конкатенація, порівняння та пошук у рядках

### **3. Текстові Файли:**

- 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

### **4. Бінарні Файли:**

- 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

### **5. Стандартна бібліотека та робота з файлами:**

- Огляд стандартної бібліотеки для роботи з файлами
- Потoki вводу/виводу: `ifstream`, `ofstream`, `fstream`
- Обробка помилок при роботі з файлами

### **6. Створення й використання бібліотек:**

- 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

## *Індивідуальний план опрацювання теорії:*

### **1. Вступ до Роботи з Файлами:**

- Джерела:

<https://acode.com.ua/urok-220-bazovyj-fajlovyj-vvid-i-vyvid/#toc-4>

- Висновок:

Є три основні класи файлового вводу/виводу в мові C++:

`ifstream` (є дочірнім класу `istream`);

`ofstream` (є дочірнім класу `ostream`);

`fstream` (є дочірнім класу `iostream`).

## 2. Символи і Рядкові Змінні:

- Джерела:  
<https://acode.com.ua/urok-82-ryadky-c-style/>
- Висновок:  
**Сучасний C++ підтримує два різних типи рядків:**  
1) `std::string` (як частина Стандартної бібліотеки мови C++);  
2) рядки C-style (успадковані від мови Cі).

## 4. Бінарні Файли:

- Джерела:  
<https://acode.com.ua/urok-221-randomnyj-fajlovyj-vvid-i-vyvid/>
- Висновок:  
**Бінарні файли** — це файли, які зберігають дані в такому вигляді, як вони представлені в пам'яті (байтами), без перетворення їх у текстовий формат.  
**Відмінності від текстових файлів**
  1. **Текстові файли** зберігають дані у вигляді символів (наприклад, 123 як три символи '1', '2', '3').
  2. **Бінарні файли** зберігають дані у вигляді байтів (наприклад, число 123 зберігається як 01111011, тобто 1 байт).**Переваги бінарних файлів:**
  - Зберігають місце, оскільки не потребують конвертації.
  - Швидші при записі та читанні великих даних.
  - Використовуються для збереження структур, об'єктів, медіа

## 5. Стандартна бібліотека та робота з файлами:

- Джерела:  
<https://acode.com.ua/urok-221-randomnyj-fajlovyj-vvid-i-vyvid/>
- Висновок: **Стандартна бібліотека шаблонів** (скор. “*STL*” від “*Standard Template Library*”) — це частина Стандартної бібліотеки C++, яка містить набір шаблонів контейнерних класів (наприклад, [`std::vector`](#) і [`std::array`](#)), алгоритмів і ітераторів.

- **ofstream** (output file stream)

Призначення:

Використовується для запису даних у файл.

Особливості:

Відкриває файл тільки для запису.

За замовчуванням, якщо файл уже існує, його вміст буде видалено (перезапис).

Якщо файл не існує, він буде створений.

- **ifstream** (input file stream)

Призначення:

Використовується для читання даних із файлу.

Особливості:

Відкриває файл тільки для читання.

Якщо файл не існує або не може бути відкритий, об'єкт буде в стані помилки

- **fstream** (file stream)

Призначення:

Поєднує функціональність ofstream та ifstream. Використовується як для читання, так і для запису в один і той самий файл.

Особливості:

- Може одночасно виконувати операції читання та запису.
- Ви повинні явно вказати режим відкриття файлу (наприклад, читання, запис чи обидва).

## *Виконання роботи:*

### Завдання №1

(VNS Lab 6 Task1) - Задано рядок, що складається із символів.

Символи поєднуються в слова. Слова одне від одного відокремлюються одним або декількома пробілами. Наприкінці тексту ставиться крапка. Текст містить не більше 255 символів.

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

-Перетворити рядок так, щоб всі букви в ньому були відсортовані за зростанням.

### Завдання №2

(VNS Lab8 Task1) - 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 "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 Lab9 Task1) –

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

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

### Завдання №4

(Algotester Lab 4.2) –

## Lab 4v2

Обмеження: 1 сек., 256 МБ

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  $a$

### Обмеження

$1 \leq N, K \leq 1000$

$0 \leq a_i \leq 100$

## Завдання №5 (Algotester Lab 6)

### Lab 6v2

Обмеження: 2 сек., 256 МБ

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  $s_i$  - 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

$|row_i| = N$

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

$1 \leq Q \leq 64$

## 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.

## 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 операції, 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 стрічка, 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.

## 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.

## 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`, `stringstreambuf` 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)

### Зуби

Обмеження: 2 сек., 256 МБ

Мале Бісеня любить гострити зуби. А Зла Тігонька любить до нього підходити і питатися: «Що, зуби гостриш?». Бісеняті так не дуже подобається, тому воно придумало робити так.

У Малого Бісеняті є  $n$  зубів. Кожен зуб має коефіцієнт загостреності  $a_i$ . Також існує межа загостреності  $k$ . Якщо коефіцієнт загостреності певного зуба є більшим чи рівним межі загостреності, то такий зуб вважається загостреним.

Мале Бісеня хоче наступного разу, коли Зла Тігонька його щось запитає, показати їй якнайбільше загострених зубів, що розташовані поспіль.

Допоможіть Малому Бісеняті дізнатися, скільки найбільше зубів воно зможе показати.

#### Вхідні дані

У першому рядку задані два цілих числа  $n$  та  $k$  — кількість зубів та межа загостреності відповідно.

В другому рядку задано  $n$  цілих чисел  $a_i$  — коефіцієнти загостреності зубів.

#### Вихідні дані

Єдине ціле число — відповідь на задачу.

#### Обмеження

$$1 \leq n \leq 10^5,$$

$$1 \leq k \leq 10^9,$$

$$1 \leq a_i \leq 10^9.$$

## Завдання №1 (VNS Lab 6 Task1)

```
1  #include <iostream>
2  #include <algorithm>
3  #include <string>
4
5  using namespace std;
6
7  int main() {
8      string input;
9      cout << "Введіть рядок: ";
10     getline(cin, input);
11
12     sort(input.begin(), input.end());
13
14     cout << "Відсортований рядок: " << input << endl;
15
16     return 0;
17 }
```

```
Введіть рядок: Hello world!
Відсортований рядок: !Hdellloorw
```

## Завдання №2 (VNS Lab8 Task1)

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include <vector>
5  using namespace std;
6
7  struct SchoolStudent {
8      string surname;
9      string name;
10     string patronymic;
11     int grade;
12     string phoneNumber;
13     int mathGrade;
14     int physicsGrade;
15     int ukrainianGrade;
16     int literatureGrade;
17 };
18
19 void createFile(const string& filename, const vector<SchoolStudent>& students) {
20     ofstream file(filename, ios::binary);
21     if (!file) {
22         cout << "Помилка відкриття файлу для запису!" << endl;
23         return;
24     }
25     for (const auto& student : students) {
26         file.write((char*)&student, sizeof(SchoolStudent));
27     }
28     file.close();
29 }
30
31 void printFile(const string& filename) {
32     ifstream file(filename, ios::binary);
33     if (!file) {
34         cout << "Помилка відкриття файлу для читання!" << endl;
35         return;
36     }
37     SchoolStudent student;
38     while (file.read((char*)&student, sizeof(SchoolStudent))) {
39         cout << "Прізвище: " << student.surname << ", Ім'я: " << student.name
40             << ", По батькові: " << student.patronymic << ", Клас: " << student.grade
41             << ", Телефон: " << student.phoneNumber
42             << ", Оцінки: [Математика: " << student.mathGrade
43             << ", Фізика: " << student.physicsGrade
44             << ", Українська мова: " << student.ukrainianGrade
45             << ", Література: " << student.literatureGrade << "]" << endl;
46     }
47     file.close();
48 }
49
50 void deleteStudentsWithTwos(const string& filename) {
51     ifstream file(filename, ios::binary);
52     if (!file) {
53         cout << "Помилка відкриття файлу для читання!" << endl;
54         return;
55     }
56     vector<SchoolStudent> students;
57     SchoolStudent student;
58     while (file.read((char*)&student, sizeof(SchoolStudent))) {
59         if (student.mathGrade != 2 && student.physicsGrade != 2 &&
60             student.ukrainianGrade != 2 && student.literatureGrade != 2) {
61             students.push_back(student);
62         }
63     }
64     file.close();
65
66     ofstream outFile(filename, ios::binary);
67     if (!outFile) {
68         cout << "Помилка відкриття файлу для запису!" << endl;
69         return;
70     }
71     for (const auto& s : students) {
72         outFile.write((char*)&s, sizeof(SchoolStudent));
73     }
74     outFile.close();
75 }
76
77 // Функція для додавання учня на початок
78 void addStudentToFile(const string& filename, const SchoolStudent& newStudent) {
79     ifstream file(filename, ios::binary);
80     if (!file) {
81         cout << "Помилка відкриття файлу для читання!" << endl;

```

```

81         cout << "Помилка відкриття файлу для читання!" << endl;
82         return;
83     }
84     vector<SchoolStudent> students;
85     SchoolStudent student;
86     while (file.read((char*)&student, sizeof(SchoolStudent))) {
87         students.push_back(student);
88     }
89     file.close();
90
91     ofstream outFile(filename, ios::binary);
92     if (!outFile) {
93         cout << "Помилка відкриття файлу для запису!" << endl;
94         return;
95     }
96     outFile.write((char*)&newStudent, sizeof(SchoolStudent));
97     for (const auto& s : students) {
98         outFile.write((char*)&s, sizeof(SchoolStudent));
99     }
100    outFile.close();
101 }
102
103 int main() {
104     string filename = "students.bin";
105
106     vector<SchoolStudent> initialStudents = {
107         {"Іваненко", "Іван", "Іванович", 10, "1234567890", 5, 4, 3, 2},
108         {"Петренко", "Петро", "Петрович", 11, "0987654321", 2, 3, 4, 5}
109     };
110
111     createFile(filename, initialStudents);
112     cout << "Початковий вміст файлу:" << endl;
113     printFile(filename);
114
115     deleteStudentsWithTwos(filename);
116     cout << "\nПісля видалення учнів з двійками:" << endl;
117     printFile(filename);
118
119     SchoolStudent newStudent = {"Новий", "Учень", "Олексійович", 9, "1122334455", 4, 4, 5, 5};
120     addStudentToFile(filename, newStudent);
121     cout << "\nПісля додавання нового учня на початок:" << endl;
122     printFile(filename);
123
124     return 0;
125 }
126

```

Початковий вміст файлу:

Прізвище: Іваненко, Ім'я: Іван, По батькові: Іванович, Клас: 10, Телефон: 1234567890, Оцінки: [Математика: 5, Фізика: 4, Українська мова: 3, Література: 2]  
 Прізвище: Петренко, Ім'я: Петро, По батькові: Петрович, Клас: 11, Телефон: 0987654321, Оцінки: [Математика: 2, Фізика: 3, Українська мова: 4, Література: 5]

## Завдання №3 (VNS Lab9 Task1)

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include <cctype>
5  using namespace std;
6
7  void createFileF1(const string& filename) {
8      ofstream file(filename);
9      if (!file) {
10         cout << "Помилка відкриття файлу для запису!" << endl;
11         return;
12     }
13     file << "Це перший рядок.\n";
14     file << "Другий рядок містить більше тексту.\n";
15     file << "Третій рядок також є.\n";
16     file << "Четвертий рядок тестового файлу.\n";
17     file << "П'ятий рядок тут.\n";
18     file << "Шостий рядок для перевірки.\n";
19     file << "Сьомий рядок.\n";
20     file << "Восьмий рядок.\n";
21     file << "Дев'ятий рядок тут.\n";
22     file << "Десятий рядок для файлу F1.\n";
23     file.close();
24 }
25
26 void copyLines(const string& fileF1, const string& fileF2, int N, int K) {
27     ifstream inFile(fileF1);
28     ofstream outFile(fileF2);
29
30     if (!inFile || !outFile) {
31         cout << "Помилка відкриття файлів для копіювання!" << endl;
32         return;
33     }
34
35     string line;
36     int lineNumber = 0;
37
38     while (getline(inFile, line)) {
39         ++lineNumber;
40         if (lineNumber >= N && lineNumber <= K) {
41             outFile << line << endl;
42         }
43     }
44
45     inFile.close();
46     outFile.close();
47 }
48
49 int countConsonants(const string& filename) {
50     ifstream file(filename);
51     if (!file) {
52         cout << "Помилка відкриття файлу для підрахунку!" << endl;
53         return 0;
54     }
55
56     int consonantCount = 0;
57     string vowels = "аеєиїїоуюяАЕЄІІІОУЮЯаеіоуяАЕІОУЮЯаеіоуяАЕІОУЮЯ";
58     char ch;
59
60     while (file.get(ch)) {
61         if (isalpha(ch) && vowels.find(ch) == string::npos) {
62             ++consonantCount;
63         }
64     }
65
66     file.close();
67     return consonantCount;

```

```

66     file.close();
67     return consonantCount;
68 }
69
70 int main() {
71     string fileF1 = "F1.txt";
72     string fileF2 = "F2.txt";
73
74     createFileF1(fileF1);
75     cout << "Файл F1 створено та заповнено.\n";
76
77     int N, K;
78     cout << "Введіть N (початковий номер рядка): ";
79     cin >> N;
80     cout << "Введіть K (кінцевий номер рядка): ";
81     cin >> K;
82
83     copyLines(fileF1, fileF2, N, K);
84     cout << "Рядки з " << N << " до " << K << " скопійовано у файл F2.\n";
85
86     int consonants = countConsonants(fileF2);
87     cout << "Кількість приголосних букв у файлі F2: " << consonants << endl;
88
89     return 0;
90 }
91

```

```

Файл F1 створено та заповнено.
Введіть N (початковий номер рядка): 2
Введіть K (кінцевий номер рядка): 6
Рядки з 2 до 6 скопійовано у файл F2.
Кількість приголосних букв у файлі F2: 0
PS C:\Users\Solomia>

```

## Завдання №4

### (Algot1ester Lab 4.2)

#### 1 Варіант:

```

1  #include <iostream>
2  #include <vector>
3  #include <set>
4  #include <algorithm>
5
6  using namespace std;
7
8  void printVector(vector<int>& vec)
9  {
10     cout << vec.size() << endl;
11     for (int i = 0; i < vec.size(); i++) {
12         cout << vec[i] << " ";
13     }
14     cout << endl;
15 }
16
17 int main() {
18     int N, K;
19     cin >> N;
20     if (N < 1 || N > 1000) return 1;
21
22     cin >> K;
23     if (K < 1 || K > 1000) return 1;
24
25     vector<int> numbers(N);
26     for (int i = 0; i < N; i++) {
27         cin >> numbers[i];
28         if (numbers[i] < 0 || numbers[i] > 100) return 1;
29     }
30
31     set<int> uniqueSet;
32     for (int i = 0; i < N; i++) {
33         uniqueSet.insert(numbers[i]);
34     }
35
36     vector<int> uniqueNumbers;
37     for (int val : uniqueSet) {
38         uniqueNumbers.push_back(val);
39     }
40
41     sort(uniqueNumbers.begin(), uniqueNumbers.end());
42
43     K = K % uniqueNumbers.size();
44     vector<int> rotatedNumbers;
45     for (int i = K; i < uniqueNumbers.size(); i++) {
46         rotatedNumbers.push_back(uniqueNumbers[i]);
47     }
48     for (int i = 0; i < K; i++) {
49         rotatedNumbers.push_back(uniqueNumbers[i]);
50     }
51
52     printVector(rotatedNumbers);
53
54     return 0;
55 }
56

```

## 2 Варіант:

```
1 #include <iostream>
2 using namespace std;
3
4 int removeDuplicates(int arr[], int n) {
5     int temp[1000];
6     int size = 0;
7
8     for (int i = 0; i < n; i++) {
9         bool found = false;
10        for (int j = 0; j < size; j++) {
11            if (arr[i] == temp[j]) {
12                found = true;
13                break;
14            }
15        }
16        if (!found) {
17            temp[size++] = arr[i];
18        }
19    }
20
21    for (int i = 0; i < size; i++) {
22        arr[i] = temp[i];
23    }
24    return size;
25 }
26
27 void sortArray(int arr[], int n) {
28     for (int i = 0; i < n - 1; i++) {
29         for (int j = i + 1; j < n; j++) {
30             if (arr[i] > arr[j]) {
31                 int temp = arr[i];
32                 arr[i] = arr[j];
33                 arr[j] = temp;
34             }
35         }
36     }
37 }
38
39 void rotateArray(int arr[], int n, int k) {
40     int temp[1000];
41     k = k % n;
42
43     for (int i = 0; i < n; i++) {
44         temp[i] = arr[(i + k) % n];
45     }
46
47     for (int i = 0; i < n; i++) {
48         arr[i] = temp[i];
49     }
50 }
51
52 void printArray(int arr[], int n) {
53     cout << n << endl;
54     for (int i = 0; i < n; i++) {
55         cout << arr[i] << " ";
56     }
57     cout << endl;
58 }
59
60 int main() {
61     int N, K;
62     cin >> N;
63     if (N < 1 || N > 1000) return 1;
64
65     cin >> K;
66     if (K < 1 || K > 1000) return 1;
67
68     int arr[1000];
69     for (int i = 0; i < N; i++) {
70         cin >> arr[i];
71         if (arr[i] < 0 || arr[i] > 100) return 1;
72     }
73
74     N = removeDuplicates(arr, N);
75
76     sortArray(arr, N);
77
78     rotateArray(arr, N, K);
79
80     printArray(arr, N);
81
82     return 0;
83 }
84
```

## Завдання №5 (Algotester Lab6)

```

1  #include <iostream>
2  #include <cstring>
3  #include <cmath>
4  #include <set>
5
6  using namespace std;
7
8  struct Figure {
9      char name;
10     int x, y;
11 };
12
13 const char F[] = {'P', 'R', 'N', 'B', 'K', 'Q', 'X', 'O'};
14
15 int cellCheck(Figure &f, int x, int y) {
16     if (f.x == x && f.y == y)
17         return 6;
18     switch (f.name) {
19     case 'P':
20         if (f.x == x - 1 && abs(f.y - y) == 1)
21             return 0;
22         break;
23     case 'R':
24         if (f.x == x || f.y == y)
25             return 1;
26         break;
27     case 'N':
28         if ((abs(f.x - x) == 2 && abs(f.y - y) == 1) || (abs(f.x - x) == 1 && abs(f.y - y) == 2))
29             return 2;
30         break;
31     case 'B':
32         if (abs(f.x - x) == abs(f.y - y))
33             return 3;
34         break;
35     case 'K':
36         if (abs(f.x - x) <= 1 && abs(f.y - y) <= 1)
37             return 4;
38         break;
39     case 'Q':
40         if (f.x == x || f.y == y || abs(f.x - x) == abs(f.y - y))
41             return 5;
42         break;
43     default:
44         break;
45     }
46     return -1;
47 }
48
49 int main() {
50     Figure figures[64];
51     int figCount = 0;
52
53     for (int i = 0; i < 8; i++) {
54         string row;
55         cin >> row;
56         for (int j = 0; j < row.size(); j++) {
57             if (row[j] != F[7]) {
58                 figures[figCount++] = {row[j], i + 1, j + 1};
59             }
60         }
61     }
62 }

```



```

62
63
64     int n;
65     cin >> n;
66     string* results = new string [n];
67
68     for (int i = 0; i < n; i++) {
69         set<char> result;
70         int x, y;
71         cin >> x >> y;
72         bool isEmpty = true;
73         bool onCell = false;
74
75         for (int j = 0; j < figCount; j++) {
76             int index = cellCheck(figures[j], x, y);
77             if (index >= 0 && index <= 5) {
78                 result.insert(F[index]);
79                 isEmpty = false;
80             } else if (index == 6) {
81                 onCell = true;
82                 isEmpty = false;
83             }
84         }
85
86         if (onCell) {
87             results[i] = "X";
88         } else if (isEmpty) {
89             results[i] = "0";
90         } else {
91             for (set<char>::iterator it = result.begin(); it != result.end(); ++it) {
92                 results[i] += *it;
93             }
94         }
95     }
96
97     for (int i = 0; i < n; i++) {
98         cout << results[i] << endl;
99     }
100
101     delete[] results;
102
103     return 0;
104 }
105

```

## Завдання №6 (Class Practice Work Task 1)

```

1  #include <iostream>
2  #include <fstream>
3
4  using namespace std;
5
6  enum OperationStatus { Ok, Error };
7
8  OperationStatus saveToFile(const char *fileName, const char *data) {
9      ofstream outFile(fileName);
10
11      if (!outFile.is_open()) {
12          cerr << "Помилка: неможливо відкрити файл для запису." << endl;
13          return Error;
14      }
15
16      outFile << data;
17
18      if (outFile.fail()) {
19          cerr << "Помилка: запис у файл не вдался." << endl;
20          return Error;
21      }
22
23      outFile.close();
24      if (outFile.fail()) {
25          cerr << "Помилка: закриття файлу завершилося помилкою." << endl;
26          return Error;
27      }
28
29      return Ok;
30 }
31
32 int main() {
33     const char *fileName = "OutputFile.txt";
34     char inputText[256];
35
36     cout << "Введіть текст для запису у файл: ";
37     cin.getline(inputText, 256);
38
39     OperationStatus status = saveToFile(fileName, inputText);
40
41     if (status == Ok) {
42         cout << "Файл успішно записаний." << endl;
43     } else {
44         cout << "Не вдалося записати файл." << endl;
45     }
46
47     return 0;
48 }
49

```

(Class Practice Work Task 2)

```

1  #include <iostream>
2  #include <fstream>
3
4  using namespace std;
5
6  bool copyFile(const char *sourceFile, const char *targetFile) {
7      ifstream source(sourceFile, ios::binary);
8      if (!source) {
9          cout << "Не вдалося відкрити вихідний файл: " << sourceFile << endl;
10         return false;
11     }
12
13     ofstream target(targetFile, ios::binary);
14     if (!target) {
15         cout << "Не вдалося створити або відкрити файл для запису: " << targetFile << endl;
16         return false;
17     }
18
19     target << source.rdbuf();
20
21     if (target.fail()) {
22         cout << "Помилка під час запису в файл: " << targetFile << endl;
23         return false;
24     }
25
26     cout << "Файл успішно скопійовано." << endl;
27     return true;
28 }
29
30 int main() {
31     const char *sourceFile = "source.txt";
32     const char *targetFile = "destination.txt";
33
34     if (!copyFile(sourceFile, targetFile)) {
35         cout << "Копіювання файлу не вдалося." << endl;
36     }
37
38     return 0;
39 }
40

```

## Завдання №7 (Self Work)

```

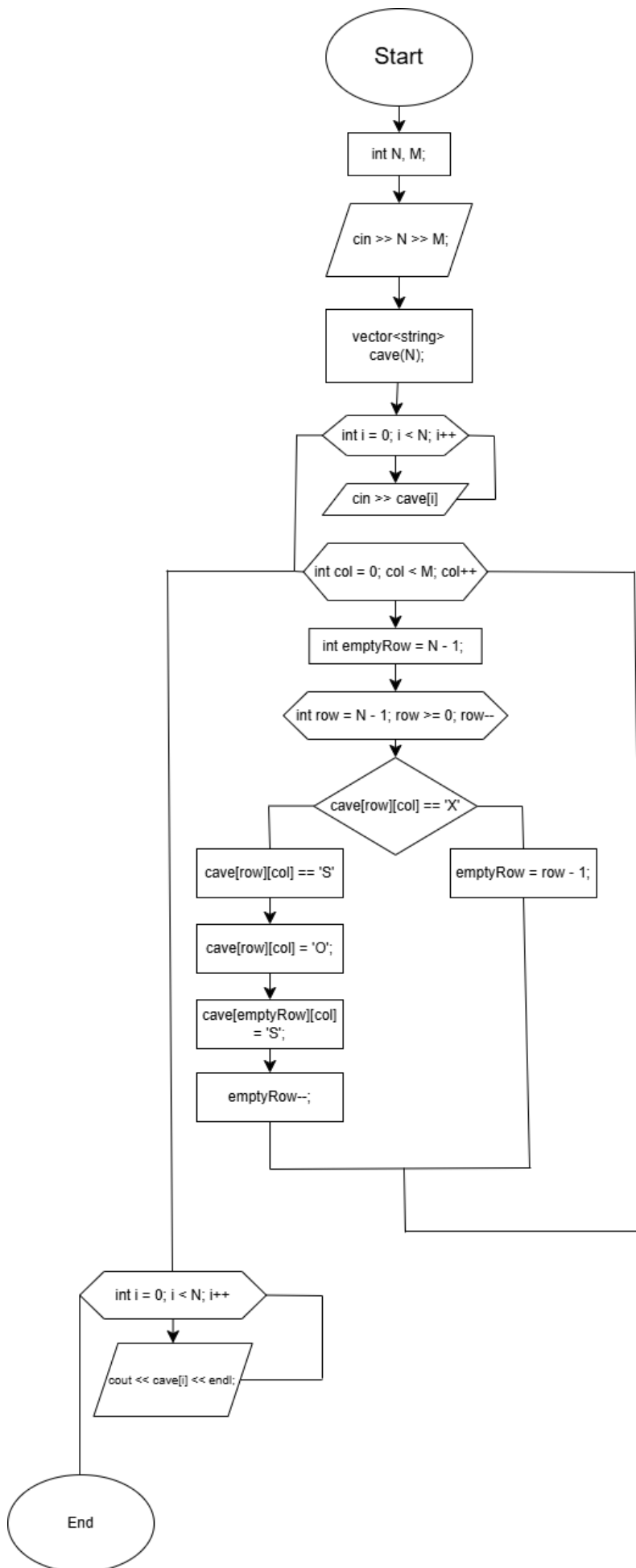
#include <iostream>
#include <vector>
using namespace std;

int main() {
    int n, k;
    cin >> n >> k;
    vector<int> a(n);
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }

    int max_streak = 0, current_streak = 0;
    for (int i = 0; i < n; i++) {
        if (a[i] >= k) {
            current_streak++;
            max_streak = max(max_streak, current_streak);
        } else {
            current_streak = 0;
        }
    }

    cout << max_streak << endl;
    return 0;
}

```



*Висновок:* На лабораторній: роботі №5 (еріс 5) , я ознайомилась з основним принципам роботи з файлами у C++ та розібралася з текстовими і бінарними файлами, включаючи операції відкриття, читання, запису та закриття.