

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



## Звіт

**про виконання лабораторних та практичних робіт блоку № 5**

На тему: «Програмування: алгоритм, програма, код. Системи числення.  
Двійкова система числення. Розробка та середовище розробки програми.»

**з дисципліни:** «Основи програмування»

до:

Практичних Робіт до блоку № 5

**Виконав:**

Студент групи ШІ-11

Левченко Денис

Львів 2024

**Тема:** Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек

**Мета:** Ознайомитися з основними поняттями роботи з файлами, включаючи бінарні та текстові файли, символи й рядкові змінні, а також засвоїти методи роботи з файлами за допомогою стандартної бібліотеки, створення та використання власних бібліотек.

## 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
  - Потоки 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:**

Вступ до Роботи з Файлами  
Символи і Рядкові Змінні  
Текстові Файли  
Бінарні Файли  
Стандартна бібліотека та робота з файлами  
Створення й використання бібліотек

### **Джерела:**

- Chat gpt
- Список відтворення на YouTube (  
<https://youtube.com/playlist?list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g&si=sXvmPdnGkwvJLXUi> )

## **Виконання роботи:**

### **VNS Lab 6 - Task 1-16:**

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.

Визначити які слова зустрічаються в рядку по одному разі.

### **VNS Lab 8 - Task 1-15:**

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 "Власник автомобіля":

- прізвище, ім'я, по батькові;
- номер автомобіля;
- телефон;
- номер техпаспорта.

Знищити елемент із заданим номером, додати 2 елементи перед елементом із

заданим прізвищем.

### **VNS Lab 9 - Task 1-16:**

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

інформацію

1) Скопіювати з файлу F1 у файл F2 всі рядки, що починаються на букву «А» і

закінчуються на букву «З», розташовані між рядками з номерами N1 й N2.

2) Визначити кількість слів у першому рядку файлу F2.

## Algotester Lab 4 v3:

Вам дано масив, який складається з NN 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.

## 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.

**Пам'ятайте, ви маєте написати 2 варіанти розв'язку, один з використанням засобів STL (`std::set_intersection`, `std::set_symmetric_difference`, `std::set_difference`, `std::set_union`), інший зі своєю реалізацією. Своє сортування можна не писати.**

## Algotester Lab 4 v2:

Вам дано масив а з N цілих чисел.

Спочатку видаліть масиву аа усі елементи що повторюються, наприклад масив [1, 3, 3, 4] має перетворитися у [1, 3, 4].

Після цього оберніть посортовану версію масиву аа на К, 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 aa

## Вихідні дані

У першому рядку ціле число N - розмір множини aa

У наступному рядку N цілих чисел - множина a

**Пам'ятайте, ви маєте написати 2 варіанти розв'язку, один з використанням засобів STL (std::set\_intersection, std::set\_symmetric\_difference, std::set\_difference, std::set\_union), інший зі своєю реалізацією. Своє сортування можна не писати.**

## Algotester Lab 6:

У вас є шахова дошка розміром  $8 \times 8 \times 8$  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  $\{x,y\}\{x,y\}$ . На кожен запит ви маєте вивести стрічку sisi - 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.

## Class Practice Work:

Реалізувати функцію створення файла і запису в нього даних:

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

```
FileOpResult write_to_file(char *name, char *content);
```

Реалізувати функцію створення файла і запису в нього даних:

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

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

## Self Practice Task (6.1):

Вам дано N слів та число K.

Ваше завдання перелічити букви в словах, які зустрічаються в тексті більше-рівне ніж K разів (саме слово, не буква!).

Великі та маленькі букви вважаються однаковими, виводити необхідно малі,

посортовані від останньої до першої у алфавіті. Букву потрібно виводити лише

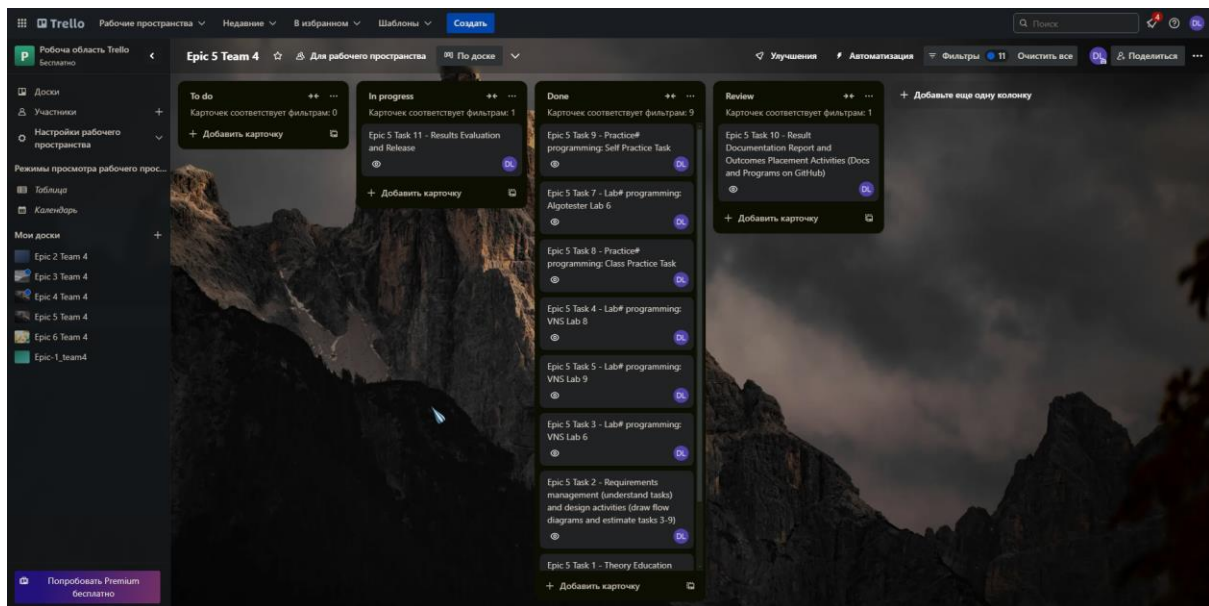
один раз.

У випадку якщо таких букв немає - вивести "Empty!".

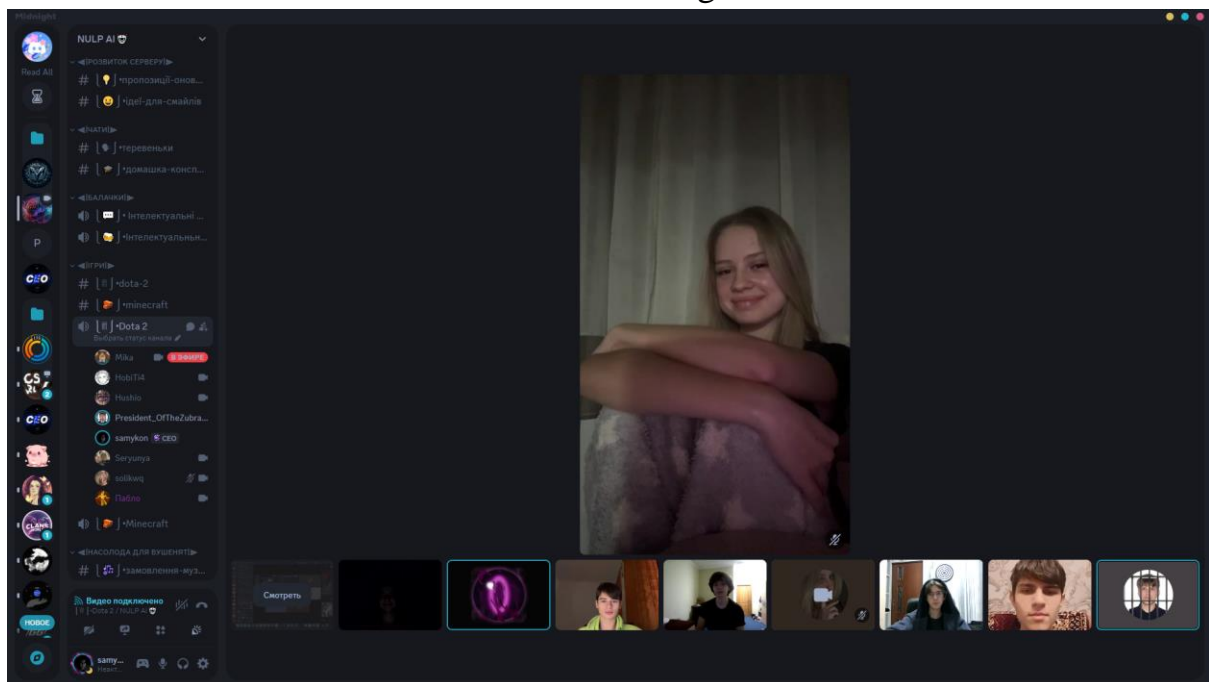


# 1. Requirements management and design activities

## Team Trello dashboard for task control



## Team meeting



## Time planning for tasks & tasks design

VNS Lab 6

$\sim 20x6$

VNS Lab 8

$\sim 3200$

VNS Lab 9

$\sim 30x6$

Algotester Lab 4

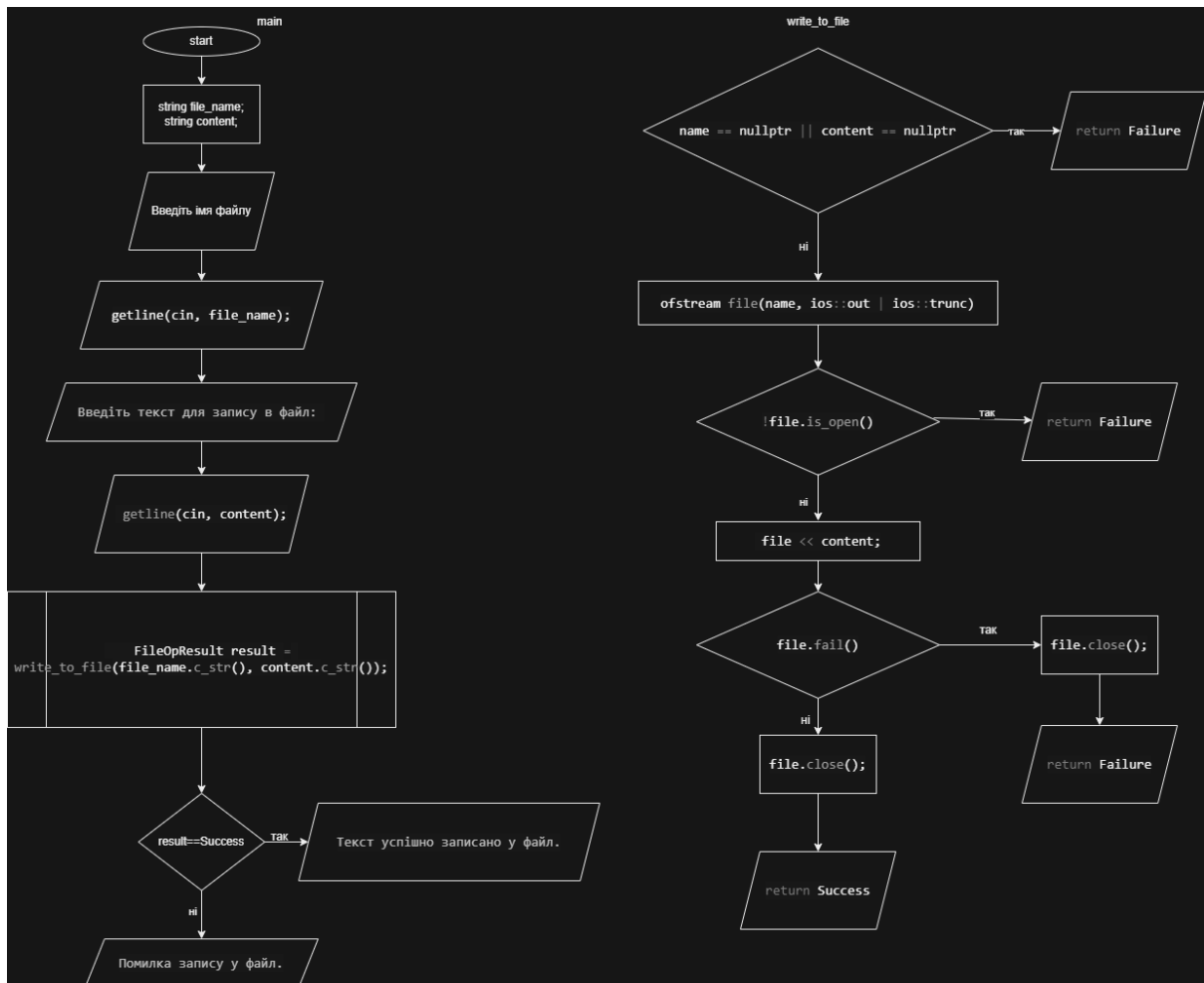
$\sim 10x6$

Algotester Lab 4

$\sim 10x6$

Class Practice Work

$\sim 50x6$



## Self Practice Work

~30хв

Код програми з посиланням на зовнішні ресурси

VNS Lab 6

vns\_lab\_6\_task\_2\_variant\_1\_denys\_levchenko.cpp

```
1  #include <iostream>
2  #include <string>
3  #include <sstream>
4  #include <map>
5  #include <Windows.h>
6
7
8
9  using namespace std;
10 int main() {
11     SetConsoleOutputCP(65001);
12     char s[256];
13     cout << "Введіть рядок (до 255 символів): ";
14     gets(s);
15
16     stringstream ss(s);
17     string word;
18     map<string, int> wordCount;
19
20     while (ss >> word) {
21         wordCount[word]++;
22     }
23
24
25     cout << "Слова, які зустрічаються один раз: ";
26     for (const auto& pair : wordCount) {
27         if (pair.second == 1) {
28             cout << pair.first << " ";
29         }
30     }
31     cout << endl;
32
33     return 0;
34 }
35
```

VNS Lab 8:

```

1 #include <iostream>
2 #include <fstream>
3 #include <string>
4
5 using namespace std;
6 struct Owner {
7     char name[50];
8     char car_number[20];
9     char phone[20];
10    char passport_number[20];
11 };
12
13 void addRecord(fstream &file) {
14     Owner owner;
15     cout << "Enter Name: ";
16     cin.ignore();
17     cin.getline(owner.name, 50);
18     cout << "Enter Car Number: ";
19     cin.getline(owner.car_number, 20);
20     cout << "Enter Phone: ";
21     cin.getline(owner.phone, 20);
22     cout << "Enter Passport Number: ";
23     cin.getline(owner.passport_number, 20);
24     file.write(reinterpret_cast<char*>(owner), sizeof(Owner));
25 }
26
27 void printRecords(fstream &file) {
28     Owner owner;
29     while (file.read(reinterpret_cast<char*>(owner), sizeof(Owner))) {
30         cout << "Name: " << owner.name << "\n";
31         cout << "Car Number: " << owner.car_number << "\n";
32         cout << "Phone: " << owner.phone << "\n";
33         cout << "Passport Number: " << owner.passport_number << "\n";
34         cout << "-----\n";
35     }
36 }
37
38 void deleteRecord(const char* filename, const char* passportToDelete) {
39     ifstream file(filename, ios::binary);
40     ofstream temp("temp.dat", ios::binary);
41     Owner owner;
42     bool found = false;
43
44     while (file.read(reinterpret_cast<char*>(owner), sizeof(Owner))) {
45         if (strcmp(owner.passport_number, passportToDelete) != 0) {
46             temp.write(reinterpret_cast<char*>(owner), sizeof(Owner));
47         } else {
48             found = true;
49         }
50     }
51
52     file.close();
53     temp.close();
54     remove(filename);
55     rename("temp.dat", filename);
56
57     if (found) {
58         cout << "Record deleted.\n";
59     } else {
60         cout << "Record not found.\n";
61     }
62 }
63
64 void insertRecordBeforeName(const char* filename, const char* nameToFind) {
65     ifstream file(filename, ios::binary);
66     ofstream temp("temp.dat", ios::binary);
67     Owner owner;
68     bool inserted = false;
69
70     while (file.read(reinterpret_cast<char*>(owner), sizeof(Owner))) {
71         if ((!inserted && strcmp(owner.name, nameToFind) == 0) {
72             cout << "Insert new record:\n";
73             addRecord(temp);
74             inserted = true;
75         }
76         temp.write(reinterpret_cast<char*>(owner), sizeof(Owner));
77     }
78
79     if (!inserted) {
80         cout << "Name not found. Adding record at the end.\n";
81         addRecord(temp);
82     }
83
84     file.close();
85     temp.close();
86     remove(filename);
87     rename("temp.dat", filename);
88 }
89
90 int main() {
91     const char* filename = "owners.dat";
92     int choice;
93     char searchValue[50];
94
95     while (true) {
96         cout << "1. Add Record\n2. Print Records\n3. Delete Record by Passport Number\n4. Insert Record Before Name\n5. Exit\n";
97         cout << "Enter choice: ";
98         cin >> choice;
99
100         if (choice == 1) {
101             ofstream file(filename, ios::binary | ios::app);
102             addRecord(file);
103             file.close();
104         } else if (choice == 2) {
105             ifstream file(filename, ios::binary);
106             printRecords(file);
107             file.close();
108         } else if (choice == 3) {
109             cout << "Enter Passport Number to delete: ";
110             cin >> searchValue;
111             deleteRecord(filename, searchValue);
112         } else if (choice == 4) {
113             cout << "Enter Name to insert record before: ";
114             cin.ignore();
115             cin.getline(searchValue, 50);
116             insertRecordBeforeName(filename, searchValue);
117         } else if (choice == 5) {
118             break;
119         } else {
120             cout << "Invalid choice. Try again.\n";
121         }
122     }
123
124     return 0;
125 }
126
127

```

## VNS Lab 9

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <ctype.h>
5
6
7  int count_words(const char *line);
8  int line_matches_condition(const char *line);
9
10 int main() {
11     FILE *f1, *f2;
12     char line[256];
13     int current_line = 1;
14     int N1 = 1;
15     int N2 = 5;
16
17     if ((f1 = fopen("F1.txt", "r")) == NULL) {
18         printf("Помилка відкриття файлу F1.txt\n");
19         return 1;
20     }
21
22     if ((f2 = fopen("F2.txt", "w")) == NULL) {
23         printf("Помилка відкриття файлу F2.txt\n");
24         fclose(f1);
25         return 1;
26     }
27
28     while (fgets(line, 256, f1) != NULL) {
29         if (current_line >= N1 && current_line <= N2 && line_matches_condition(line)) {
30             fputs(line, f2);
31         }
32         current_line++;
33     }
34
35     fclose(f1);
36     fflush(f2);
37     fclose(f2);
38
39     if ((f2 = fopen("F2.txt", "r")) == NULL) {
40         printf("Помилка відкриття файлу F2.txt\n");
41         return 1;
42     }
43
44     if (fgets(line, 256, f2) != NULL) {
45         int word_count = count_words(line);
46         printf("Кількість слів у першому рядку файлу F2: %d\n", word_count);
47     } else {
48         printf("Файл F2 порожній або помилка читання\n");
49     }
50
51     fclose(f2);
52     return 0;
53 }
54
55
56 int line_matches_condition(const char *line) {
57     while (isspace((unsigned char)*line)) line++;
58     int len = strlen(line);
59     while (len > 0 && isspace((unsigned char)line[len - 1])) len--;
60
61     return (len >= 2 && strncmp(line, "A", 2) == 0 && strncmp(&line[len - 2], "B", 2) == 0);
62 }
63
64
65 int count_words(const char *line) {
66     int count = 0;
67     int in_word = 0;
68
69     while (*line) {
70         if (isspace((unsigned char)*line)) {
71             in_word = 0;
72         } else if (!in_word) {
73             in_word = 1;
74             count++;
75         }
76         line++;
77     }
78
79     return count;
80 }
81
```

### Algotester Lab 4 variant 3:

```
algotester_lab_4_variant_3_v1_denys_levchenko.cpp
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  #include <unordered_set>
5
6  using namespace std;
7  bool customComparator(int a, int b) {
8      if (a % 3 != b % 3)
9          return a % 3 < b % 3;
10     if (a % 3 == 0)
11         return a < b;
12     if (a % 3 == 1)
13         return a > b;
14     return a < b;
15 }
16
17 int main() {
18     int n;
19     cin >> n;
20     vector<int> arr(n);
21
22     for (int i = 0; i < n; ++i) {
23         cin >> arr[i];
24     }
25
26     sort(arr.begin(), arr.end(), customComparator);
27
28     arr.erase(unique(arr.begin(), arr.end()), arr.end());
29
30     cout << arr.size() << endl;
31     for (int num : arr) {
32         cout << num << " ";
33     }
34     cout << endl;
35
36     return 0;
37 }
38
```

### Algotester Lab 4 variant 3 v2:

```

1  #include <iostream>
2  #include <vector>
3  #include <set>
4  #include <algorithm>
5
6  using namespace std;
7
8  int main()
9  {
10     int N;
11     cin >> N;
12     vector<int> arr(N);
13     for (auto &el : arr)
14     {
15         cin >> el;
16     }
17     partition(arr.begin(), arr.end(), [](int x)
18     { return x % 3 == 0; });
19     int firstNot0Pos;
20     for (size_t i = 0; i < arr.size(); i++)
21     {
22         if (arr[i] % 3 != 0)
23         {
24             firstNot0Pos = i;
25             break;
26         }
27     }
28     partition(arr.begin() + firstNot0Pos, arr.end(), [](int x)
29     { return x % 3 == 1; });
30     int firstNot1Pos;
31     for (size_t i = 0; i < arr.size(); i++)
32     {
33         if (arr[i] % 3 == 2)
34         {
35             firstNot1Pos = i;
36             break;
37         }
38     }
39
40     sort(arr.begin(), arr.begin() + firstNot0Pos);
41     sort(arr.begin() + firstNot0Pos, arr.begin() + firstNot1Pos, greater{});
42     sort(arr.begin() + firstNot1Pos, arr.end());
43
44     auto iter = unique(arr.begin(), arr.end());
45     arr.resize(distance(arr.begin(), iter));
46
47     cout << endl
48         << arr.size() << endl;
49     for (auto &el : arr)
50     {
51         cout << el << " ";
52     }
53 }

```

Algotester Lab 4 variant 2:



```

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

```

Algotester Lab 4 variant 2 v2:

algotester\_lab\_4\_variant\_2\_v2\_denys\_levchenko.cpp

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5  void sortArray(vector<int>& arr) {
6      for (size_t i = 0; i < arr.size(); ++i) {
7          for (size_t j = i + 1; j < arr.size(); ++j) {
8              if (arr[i] > arr[j]) {
9                  swap(arr[i], arr[j]);
10             }
11         }
12     }
13 }
14
15 void removeDuplicates(vector<int>& arr) {
16     int index = 0;
17     for (size_t i = 1; i < arr.size(); ++i) {
18         if (arr[i] != arr[index]) {
19             arr[++index] = arr[i];
20         }
21     }
22     arr.resize(index + 1);
23 }
24
25 void rotateArray(vector<int>& arr, int k) {
26     int n = arr.size();
27     k = k % n;
28     vector<int> temp(n);
29
30     for (int i = 0; i < n; ++i) {
31         temp[i] = arr[(i + k) % n];
32     }
33
34     for (int i = 0; i < n; ++i) {
35         arr[i] = temp[i];
36     }
37 }
38
39 int main() {
40     int N, K;
41     cin >> N >> K;
42     vector<int> a(N);
43
44     for (int i = 0; i < N; ++i) {
45         cin >> a[i];
46     }
47
48     sortArray(a);
49     removeDuplicates(a);
50
51     N = a.size();
52     cout << N << endl;
53
54     rotateArray(a, K);
55
56     for (int num : a) {
57         cout << num << " ";
58     }
59     cout << endl;
60
61     return 0;
62 }
```

## Algotester Lab 6 :

```

1 #include <iostream>
2 #include <vector>
3 #include <cast>
4 #include <algorithm>
5
6 using namespace std;
7
8 bool validCoordinates(int x, int y) {
9     return x >= 0 && x < 8 && y >= 0 && y < 8;
10 }
11
12 void addPawnAttacks(vector<set<char>>& attacks, int x, int y) {
13     if (validCoordinates(x + 1, y - 1)) attacks[(x + 1) * 8 + (y - 1)].insert('P');
14     if (validCoordinates(x + 1, y + 1)) attacks[(x + 1) * 8 + (y + 1)].insert('P');
15 }
16
17 void addRookAttacks(vector<set<char>>& attacks, int x, int y, char place) {
18     for (int i = 0; i < 8; i++) {
19         if (i != y) attacks[x * 8 + i].insert(place);
20         if (i != x) attacks[i * 8 + y].insert(place);
21     }
22 }
23
24 void addBishopAttacks(vector<set<char>>& attacks, int x, int y, char place) {
25     for (int i = -7; i <= 7; i++) {
26         if (i != 0) {
27             if (validCoordinates(x + i, y + i)) attacks[(x + i) * 8 + (y + i)].insert(place);
28             if (validCoordinates(x - i, y - i)) attacks[(x - i) * 8 + (y - i)].insert(place);
29         }
30     }
31 }
32
33 void addKnightAttacks(vector<set<char>>& attacks, int x, int y) {
34     int knightMoves[8][2] = {{-2, -1}, {-2, 1}, {2, -1}, {2, 1}, {-1, -2}, {-1, 2}, {1, -2}, {1, 2}};
35     for (auto& move : knightMoves) {
36         int x1 = x + move[0], y1 = y + move[1];
37         if (validCoordinates(x1, y1)) attacks[x1 * 8 + y1].insert('N');
38     }
39 }
40
41 void addKingAttacks(vector<set<char>>& attacks, int x, int y) {
42     for (int dx = -1; dx <= 1; dx++) {
43         for (int dy = -1; dy <= 1; dy++) {
44             if (dx != 0 || dy != 0) {
45                 int x1 = x + dx, y1 = y + dy;
46                 if (validCoordinates(x1, y1)) attacks[x1 * 8 + y1].insert('K');
47             }
48         }
49     }
50 }
51
52 int main() {
53     vector<string> board(8);
54     for (int i = 0; i < 8; i++) {
55         cin >> board[i];
56     }
57
58     vector<set<char>> attacks(64);
59
60     for (int i = 0; i < 8; i++) {
61         for (int j = 0; j < 8; j++) {
62             char place = board[i][j];
63             switch (place) {
64                 case 'P':
65                     addPawnAttacks(attacks, i, j);
66                     break;
67                 case 'R':
68                     addRookAttacks(attacks, i, j, 'R');
69                     break;
70                 case 'N':
71                     addKnightAttacks(attacks, i, j);
72                     break;
73                 case 'B':
74                     addBishopAttacks(attacks, i, j, 'B');
75                     break;
76                 case 'K':
77                     addKingAttacks(attacks, i, j);
78                     break;
79                 case 'Q':
80                     // Queen = Rook + Bishop
81                     addRookAttacks(attacks, i, j, 'Q');
82                     addBishopAttacks(attacks, i, j, 'Q');
83                     break;
84                 default:
85                     break;
86             }
87         }
88     }
89
90     int Q;
91     cin >> Q;
92     for (int i = 0; Q - i > 0; i++) {
93         int x, y;
94         cin >> x >> y;
95         x--; y--;
96
97         char cell = board[x][y];
98         if (cell != 'Q') {
99             cout << "Q\n";
100         } else {
101             if (attacks[x * 8 + y].empty()) {
102                 cout << "Q\n";
103             } else {
104                 vector<char> attackers(attacks[x * 8 + y].begin(), attacks[x * 8 + y].end());
105                 sort(attackers.begin(), attackers.end());
106                 for (char attacker : attackers) {
107                     cout << attacker;
108                 }
109                 cout << "\n";
110             }
111         }
112     }
113 }

```

## Class Practice Task 1:

practice\_work\_task\_1\_denys\_levchenko.cpp

```
1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  using namespace std;
6  enum FileOpResult { Success, Failure };
7
8  FileOpResult write_to_file(const char *name, const char *content) {
9      if (name == nullptr || content == nullptr) {
10         return Failure;
11     }
12
13     ofstream file(name, ios::out | ios::trunc);
14     if (!file.is_open()) {
15         return Failure;
16     }
17
18     file << content;
19     if (file.fail()) {
20         file.close();
21         return Failure;
22     }
23
24     file.close();
25     return Success;
26 }
27
28 int main() {
29     string file_name;
30     string content;
31
32     cout << "Введіть ім'я файлу: ";
33     getline(cin, file_name);
34
35     cout << "Введіть текст для запису в файл: ";
36     getline(cin, content);
37
38     FileOpResult result = write_to_file(file_name.c_str(), content.c_str());
39
40     if (result == Success) {
41         cout << "Текст успішно записано у файл." << endl;
42     } else {
43         cout << "Помилка запису у файл." << endl;
44     }
45
46     return 0;
47 }
48
```

Class Practice Task 2:

```

practice_work_task_2_denys_levchenko.cpp
1  #include <fstream>
2  #include <iostream>
3  #include <string>
4
5  using namespace std;
6  enum FileOpResult {
7      Success,
8      Failure
9  };
10
11  FileOpResult copy_file(const char* file_from, const char* file_to) {
12      ifstream input(file_from, ios::binary);
13      if (!input.is_open()) {
14          cerr << "Помилка: Не вдалося відкрити вхідний файл " << file_from << "\n";
15          return Failure;
16      }
17
18      ofstream output(file_to, ios::binary);
19      if (!output.is_open()) {
20          cerr << "Помилка: Не вдалося відкрити або створити вихідний файл " << file_to << "\n";
21          input.close();
22          return Failure;
23      }
24
25      output << input.rdbuf();
26
27      if (!output) {
28          cerr << "Помилка: Не вдалося записати дані у файл " << file_to << "\n";
29          input.close();
30          output.close();
31          return Failure;
32      }
33
34      input.close();
35      output.close();
36
37      return Success;
38  }
39
40  int main() {
41      const char* source = "otsuda.txt";
42      const char* destination = "suda.txt";
43
44      FileOpResult result = copy_file(source, destination);
45      if (result == Success) {
46          cout << "Файл успішно скопійовано!\n";
47      } else {
48          cout << "Помилка під час копіювання файлу.\n";
49      }
50
51      return 0;
52  }
53

```

Self Practice Task:

self\_practice\_work\_algotester\_task\_1\_denys\_levchenko.cpp

```
1  #include <iostream>
2  #include <vector>
3  #include <string>
4  #include <unordered_map>
5  #include <set>
6  #include <algorithm>
7  #include <cctype>
8  using namespace std;
9
10 string to_lowercase(const string &s) {
11     string result = s;
12     for (char &c : result) {
13         c = tolower(c);
14     }
15     return result;
16 }
17
18 void RASU(vector<char> &arr) {
19     set<char> unique_chars(arr.begin(), arr.end());
20     arr.assign(unique_chars.rbegin(), unique_chars.rend());
21 }
22
23 int main() {
24     int n, k;
25     cin >> n >> k;
26     vector<string> arr(n);
27     unordered_map<string, int> word_count;
28     vector<char> result;
29
30     for (int i = 0; i < n; ++i) {
31         cin >> arr[i];
32         arr[i] = to_lowercase(arr[i]);
33         word_count[arr[i]]++;
34     }
35
36     for (const auto &entry : word_count) {
37         if (entry.second >= k) {
38             for (char c : entry.first) {
39                 result.push_back(c);
40             }
41         }
42     }
43
44     RASU(result); // sort & erasing zayvi
45     if (result.empty()) {
46         cout << "Empty!" << endl;
47     } else {
48         cout << result.size() << endl;
49         for (char el : result) {
50             cout << el << " ";
51         }
52         cout << endl;
53     }
54
55     return 0;
56 }
```

## Результати виконаних завдань, тестування та фактично затрачений час

### VNS Lab 6

```
e-0ut-s4wucut0.aj5' '--stderr=Microsoft-MIEngine-Error-1qcvnfo3.03w' '--pid=Micros
Введіть рядок (до 255 символів): test hello test perevirka test
Слова, які зустрічаються один раз: hello perevirka
PS C:\Users\razzie\Desktop\epics\epic_5_practice_and_labs_denys_levchenko>
```

Затратність ~35хв

### VNS Lab 8

```
e-0ut-s4wucut0.aj5' '--stderr=Microsoft-MIEngine-Error-1qcvnfo3.03w' '--pid=Micros
1. Add Record
2. Print Records
3. Delete Record by Passport Number
4. Insert Record Before Name
5. Exit
Enter choice: 2
Name: Denys
Car Number: Savkov
Phone: +380666008043
Passport Number: JNTSNJTSJKNTSJ
-----
Name: ikamika
Car Number: JNKLKNJKNJK
Phone: 142214
Passport Number: KLGSLG
-----
Name: 1
Car Number: 1
Phone: 1
```

Затратність ~15год

### VNS lab 9

```
e-0ut-s4wucut0.aj5' '--stderr=Microsoft-MIEngine-Error-1qcvnfo3.03w' '--pid=Micros
Кількість слів у першому рядку файлу F2: 1
PS C:\Users\razzie\Desktop\epics\epic_5_practice_and_labs_denys_levchenko>
```

Затратність ~1год





## Algotester Lab 4 variant 2

```
10 3
1 2 2 3 3 3 4 5 6 7
7
4 5 6 7 1 2 3
PS C:\Users\razzie\Desktop\epics\epic_5_practice_and_labs_denys_levchenko>
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (MB)	Дії
2 дні тому	C++ 23	Зарезовано	0.004	1.215	Перегляд
2 дні тому	C++ 23	Помилка компілювання	-	-	Перегляд
2 дні тому	C++ 23	Зарезовано	0.003	1.223	Перегляд
2 дні тому	C++ 23	Помилка компілювання	-	-	Перегляд
2 дні тому	C++ 23	Неправильна відповідь 1	0.002	0.738	Перегляд

Затратність ~3год

## Algotester Lab 4 variant 3

```
10
1 33 4 8 6 5 2 7 5 0
9
0 6 33 7 4 1 2 5 8
PS C:\Users\razzie\Desktop\epics\epic_5_practice_and_labs_denys_levchenko>
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (MB)	Дії
2 дні тому	C++ 23	Зарезовано	0.003	1.195	Перегляд
2 дні тому	C++ 23	Зарезовано	0.003	1.199	Перегляд
3 дні тому	C++ 23	Зарезовано	0.003	1.188	Перегляд

Затратність ~3год

## Class Practice Task

```
C:\Users\razzie\Desktop\epics\epic_5_practice_and_labs_denys_levchenko>
Введіть ім'я файлу: test1.txt
Введіть текст для запису в файл: nLPG;IO'JWERSIOVSNDGVOILNSMLGV;KOSDVM
Текст успішно записано у файл.
PS C:\Users\razzie\Desktop\epics\epic_5_practice_and_labs_denys_levchenko>
```

```
test1.txt
1 nLPG;IO'JWERSIOVSNDGVOILNSMLGV;KOSDVM
```

Затратність ~40хв

## Self Practice Task

```
C:\Users\razzie\Desktop\epics\epic_5_practice_and_labs_denys_levchenko>
5 2
stugna
neptune
grim
oplott
Grim
4
r m i g
PS C:\Users\razzie\Desktop\epics\epic_5_practice_and_labs_denys_levchenko>
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (MB)	Дії
десятьма секунд тому	C++ 23	Зарезовано	0.046	7.086	Перегляд

Затратність ~1год

## **Висновки:**

Я навчився працювати з файлами різних типів, включаючи бінарні та текстові, використовуючи символи й рядкові змінні. Завдяки стандартній бібліотеці та створенню власних бібліотек, я отримав практичні навички ефективного збереження, обробки та взаємодії з даними.