

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 ШІ-11
Цибух Андріана

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:

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

Символи і Рядкові Змінні:

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

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

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
- 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

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

Теоретичні відомості:

1. Теоретичні відомості з переліком важливих тем:
 - Тема №*.1: Файли.
 - Тема №*.2: Бібліотеки.

2. Індивідуальний план опрацювання теорії:

- Тема №*.1: Файли.
 - Джерела Інформації
 - Лекції О. Пшеничного.
 - Практика М. Фаріон.
 - Відео **HELLO_IT_8 : Робота з файлами.**
- Статус: Ознайомлена

Тема №*.2: Бібліотеки.

- Джерела Інформації:
 - Лекції О. Пшеничного.
 - Практика М. Фаріон.
 - Сайт **tutorialspoint : C Library - <stdlib.h>.**
- Статус: Ознайомлена

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

Завдання №1 Робота з текстовими файлами

- Деталі завдання : 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.

Завдання №2 Algotester Lab 4

- Варіант завдання : 2
- Деталі завдання : дано масив а з N цілих чисел. Спочатку треба видалити масиву а усі елементи що повторюються, наприклад масив [1, 3, 3, 4] має перетворитися у [1, 3, 4]. Після цього потрібно обернути посортовану версію масиву а на K, тобто при K=3 масив [1, 2, 3, 4, 5, 6, 7] перетвориться на [4, 5, 6, 7, 1, 2, 3].
- Примітки: написати 2 варіанти розв'язку, один з використанням засобів STL (std::unique, std::sort, std::rotate), інший зі своєю реалізацією.

Завдання №3 Algotester Lab 6

- Варіант завдання : 2
- Деталі завдання : У вас є шахова дошка розміром 8×8 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 - посортвані 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.

Завдання №4 VNS Lab 6

- Варіант завдання : 13
- 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. Перетворити рядок таким чином, щоб цифри кожного слова в ньому були відсортовані за спаданням.

Завдання №5 VNS Lab 8

- Варіант завдання : 13
- 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.
Структура "Спортивна команда":
 - назва;
 - місто;
 - кількість гравців;
 - кількість набраних очків.Знищити всі елементи з кількістю очків менше заданого, додати 2 елементи на початок файлу.

Завдання №6 VNS Lab 9

- Варіант завдання : 13
- Деталі завдання : 1) Скопіювати з файлу $F1$ у файл $F2$ всі рядки, що починаються на букву «А» і розташовані між рядками з номерами $N1$ й $N2$.
2) Визначити номер того рядка, у якому найбільше приголосних букв, файлу $F2$.

Завдання №7 Self practice Algotester Lab4 V1

- Деталі завдання : дано 2 цілих чисел масиви, розміром NN та MM . Завдання вивести:
 1. Різницю $N-M$
 2. Різницю $M-N$
 3. Їх перетин
 4. Їх об'єднання
 5. Їх симетричну різницю

2. Дизайн та планована оцінка часу виконання завдань: Програма №1 Робота з текстовими файлами

```
1 #include <iostream>
2 #include <fstream>
3 #include <cstring>
4
5 using namespace std;
6
7 enum FileOpResult { Success, Failure };
8
9 FileOpResult write_to_file(char* name, char* content) {
10     if (name == nullptr || content == nullptr) {
11         return Failure;
12     }
13
14     ofstream file(name);
15     if (!file.is_open()) {
16         return Failure;
17     }
18
19     file << content;
20     if (!file) {
21         file.close();
22         return Failure;
23     }
24
25     file.close();
26     return file.fail() ? Failure : Success;
27 }
28
29 int main() {
30     char file_name[256];
31     char content[1024];
32
33     cout << "Введіть ім'я файла: ";
34     cin >> file_name;
35     cin.ignore(); // очищує буфер після зчитування ім'я файла
36
37     cout << "Введіть вміст для запису у файл: ";
38     cin.getline(content, 1024);
39
40     FileOpResult result = write_to_file(file_name, content);
41
42     if (result == Success) {
43         cout << "Дані успішно записані у файл." << endl;
44     } else {
45         cout << "Помилка запису у файл." << endl;
46     }
47
48     return 0;
49 }
50
```

```
1 #include <iostream>
2 #include <fstream>
3
4 using namespace std;
5
6 enum FileOpResult { Success, Failure };
7
8 FileOpResult copy_file(char* file_from, char* file_to) {
9     if (file_from == nullptr || file_to == nullptr) {
10         return Failure;
11     }
12
13     ifstream source(file_from, ios::binary);
14     if (!source.is_open()) {
15         return Failure;
16     }
17
18     ofstream destination(file_to, ios::binary);
19     if (!destination.is_open()) {
20         source.close();
21         return Failure;
22     }
23
24     destination << source.rdbuf(); // копіюємо
25
26     if (!destination) {
27         source.close();
28         destination.close();
29         return Failure;
30     }
31
32     source.close();
33     destination.close();
34     return (source.fail() || destination.fail()) ? Failure : Success;
35 }
36
37 int main() {
38     char source_file[256];
39     char destination_file[256];
40
41     cout << "Введіть ім'я вихідного файла: ";
42     cin >> source_file;
43
44     cout << "Введіть ім'я цільового файла: ";
45     cin >> destination_file;
46
47     FileOpResult result = copy_file(source_file, destination_file);
48
49     if (result == Success) {
50         cout << "Файл успішно скопійовано." << endl;
51     } else {
52         cout << "Помилка копіювання файла." << endl;
53     }
54
55     return 0;
56 }
```

- Час затрачений на виконання завдання : 50 хв

```
Введіть ім'я файла: first
Введіть вміст для запису у файл: i love lil peep
Дані успішно записані у файл.
```

```
aj_11 > andriana_tsybukh > epic_5 > first
1 i love lil peep
```

```
Введіть ім'я вихідного файла: first
Введіть ім'я цільового файла: second
Файл успішно скопійовано.
```

```
aj_11 > andriana_tsybukh > epic_5 > second
1 i love lil peep
```

Програма №2 Algotester Lab 4

- Час затрачений на виконання завдання : 25 хв

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm> // sort i rotate
4 #include <iterator> // unique
5
6 using namespace std;
7
8 int main() {
9     int N, K;
10    cin >> N >> K;
11
12    vector<int> a(N);
13    for(int i = 0; i < N; ++i) {
14        cin >> a[i];
15    }
16
17    // забираємо дублікати
18    sort(a.begin(), a.end());
19    auto it = unique(a.begin(), a.end());
20    a.resize(distance(a.begin(), it));
21
22    // розмір array
23    cout << a.size() << endl;
24
25    sort(a.begin(), a.end());
26
27    // повертаємо на K позицій
28    K = K % a.size(); // нормалізуем
29    rotate(a.begin(), a.begin() + K, a.end());
30
31    for(const int &num : a) {
32        cout << num << " ";
33    }
34    cout << endl;
35
36    return 0;
37 }
38
```

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm> // sort
4
5 using namespace std;
6
7 // видалення дублікати
8 void removeDuplicates(vector<int>& a) {
9     sort(a.begin(), a.end());
10    int uniqueIndex = 0;
11
12    for (int i = 1; i < a.size(); ++i) {
13        if (a[i] != a[uniqueIndex]) {
14            ++uniqueIndex;
15            a[uniqueIndex] = a[i];
16        }
17    }
18
19    a.resize(uniqueIndex + 1); // обрізаєм до розміру унікальних елементів
20 }
21
22 // ротація масиву
23 void rotateArray(vector<int>& a, int K) {
24     int N = a.size();
25     vector<int> temp(N);
26     for (int i = 0; i < N; ++i) {
27         temp[(i + N - K) % N] = a[i];
28     }
29     a = temp;
30 }
31
32 int main() {
33     int N, K;
34     cin >> N >> K;
35
36     vector<int> a(N);
37     for (int i = 0; i < N; ++i) {
38         cin >> a[i];
39     }
40
41     removeDuplicates(a);
42     cout << a.size() << endl;
43
44     // повертаємо на K позицій
45     K = K % a.size(); // нормалізуем K
46     rotateArray(a, K);
47
48     for (const int &num : a) {
49         cout << num << " ";
50     }
51     cout << endl;
52
53     return 0;
54 }
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
7 годин тому	C++ 23	Зараховано	0.003	1.336	Перегляд
7 годин тому	C++ 23	Зараховано	0.003	1.406	Перегляд

Програма №3 Algotester Lab 6

- Час затрачений на виконання завдання : 1,5 год

```

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

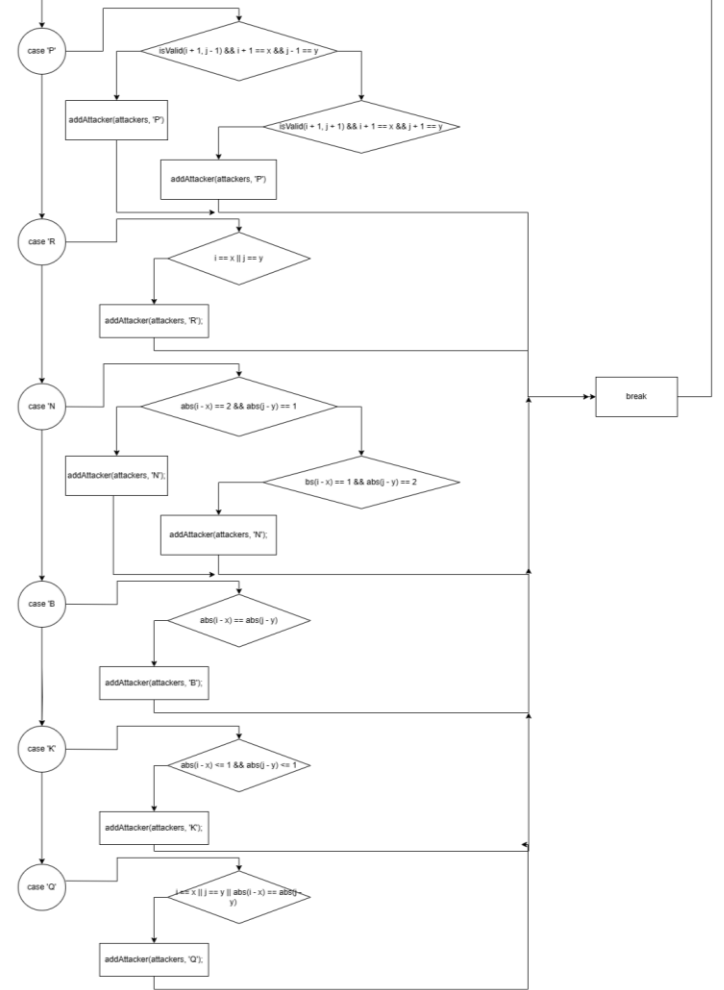
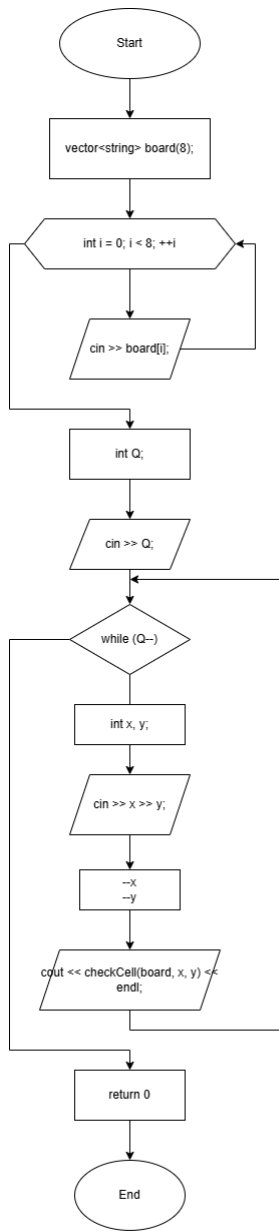
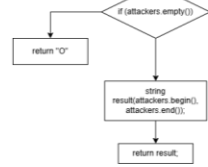
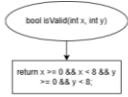
```

```

53         break;
54     case 'Q': // ферзь
55         if (i == x || j == y || abs(i - x) == abs(j - y))
56             addAttacker(attackers, 'Q');
57         break;
58     }
59 }
60
61
62 if (attackers.empty()) {
63     return "0";
64 }
65
66 string result(attackers.begin(), attackers.end());
67 return result;
68 }
69
70 int main() {
71     vector<string> board(8);
72     for (int i = 0; i < 8; ++i) {
73         cin >> board[i];
74     }
75
76     int Q;
77     cin >> Q;
78
79     while (Q--) {
80         int x, y;
81         cin >> x >> y;
82         --x;
83         --y;
84
85         cout << checkCell(board, x, y) << endl;
86     }
87
88     return 0;
89 }

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
5 годин тому	C++ 23	Зараховано	0.003	1.426	Перегляд
5 годин тому	C++ 23	Неправильна відповідь 5	0.003	0.961	Перегляд
5 годин тому	C++ 23	Неправильна відповідь 8	0.002	1.223	Перегляд
5 годин тому	C++ 23	Неправильна відповідь 9	0.002	1.203	Перегляд
7 годин тому	C++ 23	Неправильна відповідь 5	0.002	0.969	Перегляд



Програма №4 VNS Lab 6

- Час затрачений на виконання завдання : 40 хв

```
1  #include <iostream>
2  #include <string>
3  #include <sstream>
4  #include <cctype>
5  #include <algorithm>
6
7  using namespace std;
8
9  void sortByDescending(string &word) {
10     string numbers;
11     for (char ch : word) {
12         if (isdigit(ch)) {
13             numbers += ch;
14         }
15     }
16
17     sort(numbers.rbegin(), numbers.rend());
18
19     auto it = numbers.begin();
20     for (char &ch : word) {
21         if (isdigit(ch) && it != numbers.end()) {
22             ch = *it++;
23         }
24     }
25 }
26
27 int main() {
28     string input;
29     cout << "Введіть рядок (не більше 255 символів): ";
30     getline(cin, input);
31
32     if (!input.empty() && input.back() == '.') {
33         input.pop_back();
34     }
35
36     stringstream ss(input);
37     string word, result;
38
39     while (ss >> word) {
40         sortByDescending(word);
41         result += word + " ";
42     }
43
44     if (!result.empty()) {
45         result.pop_back();
46         result += ".";
47     }
48
49     cout << "Результат: " << result << endl;
50
51     return 0;
52 }
```

Програма №5 VNS Lab 8

- Час затрачений на виконання завдання : 2 год

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

```
60 void deleteTeamsWithFewPoints(const string &filename, int minPoints) {
61     ifstream inputFile(filename, ios::binary);
62     if (!inputFile) {
63         cerr << "Не вдалося відкрити файл для читання." << endl;
64         return;
65     }
66
67     Team teams[100];
68     int count = 0;
69
70     Team team;
71     while (readTeam(inputFile, team)) {
72         if (team.points >= minPoints) {
73             teams[count++] = team;
74         }
75     }
76     inputFile.close();
77
78     ofstream outputFile(filename, ios::binary | ios::trunc);
79     if (!outputFile) {
80         cerr << "Не вдалося відкрити файл для запису." << endl;
81         return;
82     }
83
84     for (int i = 0; i < count; ++i) {
85         writeTeam(outputFile, teams[i]);
86     }
87     outputFile.close();
88 }
89
90 void addTeamsToFile(const string &filename, const Team "newTeams", int newCount) {
91     ifstream inputFile(filename, ios::binary);
92     if (!inputFile) {
93         cerr << "Не вдалося відкрити файл для читання." << endl;
94         return;
95     }
96
97     Team teams[100];
98     int count = 0;
99
100    Team team;
101    while (readTeam(inputFile, team)) {
102        teams[count++] = team;
103    }
104    inputFile.close();
105
106    ofstream outputFile(filename, ios::binary | ios::trunc);
107    if (!outputFile) {
108        cerr << "Не вдалося відкрити файл для запису." << endl;
109        return;
110    }
111
112    for (int i = 0; i < newCount; ++i) {
113        writeTeam(outputFile, newTeams[i]);
114    }
115
116    for (int i = 0; i < count; ++i) {
117        writeTeam(outputFile, teams[i]);
118    }
119    outputFile.close();
120 }
```

```

122 int main() {
123     string filename = "teams.bin";
124
125     createBinaryFile(filename);
126     cout << "Початковий вміст файлу:" << endl;
127     printBinaryFile(filename);
128
129     int minPoints = 30;
130     deleteTeamsWithFewPoints(filename, minPoints);
131     cout << "\nВміст файлу після видалення команд з очками менше " << minPoints << ":" << endl;
132     printBinaryFile(filename);
133
134     Team newTeams[] = {
135         {"Команда Е", "Дніпро", 13, 50},
136         {"Команда F", "Тернопіль", 14, 45},
137     };
138     addTeamsToFile(filename, newTeams, 2);
139     cout << "\nВміст файлу після додавання нових команд:" << endl;
140     printBinaryFile(filename);
141
142     return 0;
143 }
144

```

Файл створено успішно.

Початковий вміст файлу:

Назва: Команда А, Місто: Київ, Гравці: 11, Очки: 30

Назва: Команда В, Місто: Львів, Гравці: 12, Очки: 25

Назва: Команда С, Місто: Одеса, Гравці: 10, Очки: 20

Назва: Команда D, Місто: Харків, Гравці: 15, Очки: 15

Вміст файлу після видалення команд з очками менше 30:

Назва: Команда А, Місто: Київ, Гравці: 11, Очки: 30

Вміст файлу після додавання нових команд:

Назва: Команда Е, Місто: Дніпро, Гравці: 13, Очки: 50

Назва: Команда F, Місто: Тернопіль, Гравці: 14, Очки: 45

Назва: Команда А, Місто: Київ, Гравці: 11, Очки: 30

Програма №6 VNS Lab 9

- Час затрачений на виконання завдання : 50 хв

```
1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include <cctype>
5
6  using namespace std;
7
8  // кількість приголосних літер у рядку
9  int countPrigolosni(const string &line) {
10     int count = 0;
11     for (char ch : line) {
12         if (isalpha(ch)) {
13             char lowerCh = tolower(ch);
14             if (lowerCh != 'a' && lowerCh != 'e' && lowerCh != 'i' && lowerCh != 'o' && lowerCh != 'u') {
15                 count++;
16             }
17         }
18     }
19     return count;
20 }
21
22 int main() {
23     ofstream file1("F1.txt");
24     if (!file1) {
25         cerr << "Помилка створення файлу F1!" << endl;
26         return 1;
27     }
28
29     file1 << "Apple is red.\n";
30     file1 << "Banana is yellow.\n";
31     file1 << "Avocado is green.\n";
32     file1 << "Cherry is red.\n";
33     file1 << "Apricot is orange.\n";
34     file1 << "Peach is sweet.\n";
35     file1 << "Almonds are healthy.\n";
36     file1 << "Grapes are purple.\n";
37     file1 << "Ants are tiny.\n";
38     file1 << "Blueberry is blue.\n";
39     file1.close();
40
41     int N1, N2;
42     cout << "Введіть номер першого рядка для копіювання (від 1): ";
43     cin >> N1;
44     cout << "Введіть номер останнього рядка для копіювання (до 10): ";
45     cin >> N2;
46
47     if (N1 < 1 || N2 < 1 || N1 > 10 || N2 > 10 || N1 > N2) {
48         cerr << "Невірно введені номери рядків!" << endl;
49         return 1;
50     }
51
52     ifstream file1_in("F1.txt");
53     ofstream file2("F2.txt");
54     if (!file1_in || !file2) {
55         cerr << "Помилка відкриття файлів!" << endl;
56         return 1;
57     }
58
59     ifstream file1_in("F1.txt");
60     ofstream file2("F2.txt");
61     if (!file1_in || !file2) {
62         cerr << "Помилка відкриття файлів!" << endl;
63         return 1;
64     }
65
66     int lineNum = 0;
67     string line;
68
69     while (getline(file1_in, line)) {
70         lineNum++;
71         if (lineNum >= N1 && lineNum <= N2 && !line.empty() && line[0] == 'A') {
72             file2 << line << endl;
73         }
74     }
75
76     file1_in.close();
77     file2.close();
78
79     // найбільша кількість приголосних у файлі F2
80     ifstream file2_in("F2.txt");
81     if (!file2_in) {
82         cerr << "Помилка відкриття файлу F2!" << endl;
83         return 1;
84     }
85
86     int maxConsonants = 0, maxLineNum = 0, currentLineNum = 0;
87     while (getline(file2_in, line)) {
88         currentLineNum++;
89         int consonants = countPrigolosni(line);
90         if (consonants > maxConsonants) {
91             maxConsonants = consonants;
92             maxLineNum = currentLineNum;
93         }
94     }
95
96     file2_in.close();
97
98     cout << "Номер рядка з найбільшою кількістю приголосних у файлі F2: " << maxLineNum << endl;
99     return 0;
100 }
```

Введіть номер першого рядка для копіювання (від 1): 1
Введіть номер останнього рядка для копіювання (до 10): 4
Номер рядка з найбільшою кількістю приголосних у файлі F2: 2

```
ai_11 > andriana_tsybukh > epic_5 > F1.txt
1  Apple is red.
2  Banana is yellow.
3  Avocado is green.
4  Cherry is red.
5  Apricot is orange.
6  Peach is sweet.
7  Almonds are healthy.
8  Grapes are purple.
9  Ants are tiny.
10 Blueberry is blue.
11
```

```
ai_11 > andriana_tsybukh > epic_5 > F2.txt
1   Apple is red.
2   Avocado is green.
3
```

Програма №7 Self practice Lab4 V1

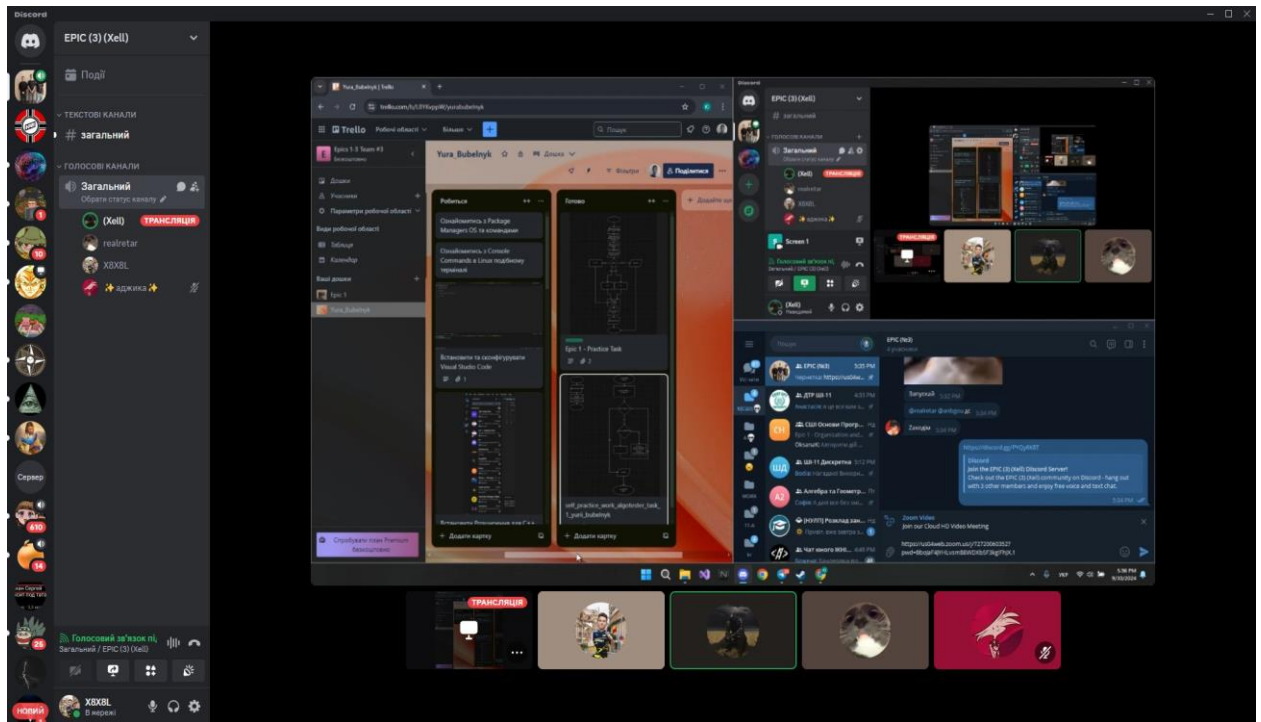
- Час затрачений на виконання завдання : 30 хв

```

1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 #include <set>
5
6 using namespace std;
7
8 void print_set(vector<int>& array){
9     cout << array.size() << endl;
10    for (int num : array){
11        cout << num << " ";
12    }
13    cout << endl << endl;
14 }
15
16 int main(){
17     int N, M;
18     cin >> N;
19     vector<int> arr1(N);
20
21     for(int i=0; i<N; i++){
22         cin >> arr1[i];
23     }
24
25     cin >> M;
26     vector<int> arr2(M);
27
28     for(int i=0; i<M; i++){
29         cin >> arr2[i];
30     }
31
32     sort(arr1.begin(), arr1.end());
33     sort(arr2.begin(), arr2.end());
34
35     // N-M
36     vector<int> diff1;
37     set_difference(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(diff1));
38
39     // M-N
40     vector<int> diff2;
41     set_difference(arr2.begin(), arr2.end(), arr1.begin(), arr1.end(), back_inserter(diff2));
42
43     // перетин
44     vector<int> intersection;
45     set_intersection(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(intersection));
46
47     // об'єднання
48     vector<int> union_set;
49     set_union(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(union_set));
50
51     // симетрична різниця
52     vector<int> symmetric_diff;
53     set_symmetric_difference(arr1.begin(), arr1.end(), arr2.begin(), arr2.end(), back_inserter(symmetric_diff));
54
55     print_set(diff1);
56     print_set(diff2);
57     print_set(intersection);
58     print_set(union_set);
59     print_set(symmetric_diff);
60
61     return 0;
62 }

```

6. Кооперація з командою:



Зустріч в дискорді. Обговорювали задачі та теорію

Висновки:

Із закінченням даного опіку, я ознайомилася із роботою з файлами, символами та рядками й бібліотеками.